A Unified Model for Solving the OOV Problem of Chinese Word Segmentation

XIAOQING LI and CHENGQING ZONG, Chinese Academy of Sciences
KEH-YIH SU, Academia Sinica

This article proposes a unified, character-based, generative model to incorporate additional resources for solving the out-of-vocabulary (OOV) problem of Chinese word segmentation, within which different types of additional information can be utilized independently in corresponding submodels. This article mainly addresses the following three types of OOV: unseen dictionary words, named entities, and suffix-derived words, none of which are handled well by current approaches. The results show that our approach can effectively improve the performance of the first two types with positive interaction in F-score. Additionally, we also analyze reason that suffix information is not helpful. After integrating the proposed generative model with the corresponding discriminative approach, our evaluation on various corpora—including SIGHAN-2005, CIPS-SIGHAN-2010, and the Chinese Treebank (CTB)—shows that our integrated approach achieves the best performance reported in the literature on all testing sets when additional information and resources are allowed.

Categories and Subject Descriptors: G.4 [**Mathematics of Computing**]: Mathematical Software—*Algorithm design and analysis*; H.4.0 [**Information Systems Applications**]: General; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms: Algorithms, Languages, Experimentation, Performance

Additional Key Words and Phrases: Chinese word segmentation, out-of-vocabulary words, model integration, domain adaptation

ACM Reference Format:
Li, X., Zong, C., and Su, K.-Y. 2015. A unified model for solving the OOV problem of Chinese word segmentation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 14, 3, Article 12 (June 2015), 29 pages.
DOI: http://dx.doi.org/10.1145/2699940

1. INTRODUCTION

Words are the basic units for text analysis, and therefore word segmentation is critical for Chinese (as well as other Asian languages) natural language processing tasks such as parsing, information retrieval, and machine translation. Although great improvements have been achieved for Chinese word segmentation (CWS) in recent years

This research has been supported by the Natural Science Foundation of China under Grant No. 61333018, the International Science & Technology Cooperation Program of China under Grant No. 2014DFA11350, and the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2012AA011102.
Authors' addresses: X. Li (corresponding author) and C. Zong, Institute of Automation, Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Haidian District, Beijing, 100190, China; emails: {xqli, cqzong}@nlpr.ia.ac.cn; K.-Y. Su; email: kysu@iis.sinica.edu.tw.

DOI: http://dx.doi.org/10.1145/2699940

[Emerson 2005; Zhao and Liu 2010], the sentence accuracy rate is still low.[1] Furthermore, OOV is the main problem [Huang and Zhao 2007] and is far from being solved; moreover, OOV is even worse for cross-domain tasks [Zhao and Liu 2010].

According to their occurrence frequencies,[2] OOV words can be sequentially classified as (1) unseen dictionary words (approximately 44.4%), which can be found in a common dictionary but never appear in the training set; (2) named entities (20.7%), which include person names, location names, and organization names; (3) factoids (16.2%), which include time, numbers, foreign strings, and URLs. These factoids are usually composed of Chinese numerals, Arabic numerals, English letters, and some specific marks (e.g., '/', '%'); (4) suffix-derived words (6.5%), which are formed by appending specific suffixes (such as "旅行者" (traveler), which is formed by appending a suffix '者' (-er) to a stem "旅行" (travel)) and will be defined in Section 5; (5) inconsistency (5.2%), which denotes the words that are segmented differently in the training set and the testing set in similar contexts; and (6) others (7.5%), such as abbreviations (e.g., "中科院" is short for "中国科学院" (Chinese Academy of Sciences)), buzzwords (e.g., "安卓" (android)), duplications (e.g., "高高兴兴" (happily)), prefix-derived words (e.g., "子系统" (subsystem)). Among these OOV types, only factoids are handled well by current approaches with character-type information [Zhao et al. 2006a]. For the remaining types, performance is still far from satisfactory [Wang et al. 2012].

The main reason that OOV words are difficult to handle lies in the conflict between the reliability and coverage rates of character n-gram features that are currently adopted. Only large n-grams (e.g., trigrams) are sufficiently reliable for recognizing word boundaries; however, they are very sparse according to Zipf's law [1949]. That is, it is very likely that the trigrams within OOV words will be unseen in the training set. As a result, additional resources such as dictionaries are essential for providing reliable information to segment these OOV words. Because these additional resources are commonly available, it is reasonable to adopt them to further improve the performance of segmenting OOV words.

To improve performance in segmenting different types of OOV words, we propose a unified framework to incorporate additional information and resources for CWS. In this framework, various additional features (corresponding to different OOV types) extracted from additional resources are independently incorporated into a generative model [Wang et al. 2009] in their corresponding submodels. Subsequently, this enhanced generative model is combined with its corresponding discriminative model [Ng and Low 2004] via log-linear combination. Within this framework, we study the following three types of OOV words: unseen dictionary words, named entities, and suffix-derived words. Our study shows that suffix information hardly improved performance, but the additional named-entity recognizer and dictionaries are very helpful. After jointly incorporating identified named entities and associated dictionary entries, our final model achieves the best performance reported in the literature on all

[1]The average sentence length of the SIGHAN-2005 testing corpora is 17 words. The sentence accuracy rate could be raised dramatically from 42% to 50%, even if the word accuracy rate were raised only 1%, from 95% to 96% (corresponding to a 20% error rate reduction). Therefore, it is worthwhile to further improve word performance.

[2]These statistics are based on the SIGHAN-2005 testing corpora. Please see Table III in Section 3.1 for details.

corpora provided by the SIGHAN-2005, CIPS-SIGHAN-2010, and Chinese Treebank for open tests.[3]

The character-based generative model for incorporating dictionary information and the effect of considering suffixes are originally described in conference papers [Li et al. 2012, 2013], respectively. In this article, the enhanced generative model is further generalized to handle various types of additional information and resources (including newly-added named-entity information). Additionally, the effect of incorporating a named-entity recognizer is presented. Furthermore, detailed OOV classification statistics and complete error analysis for each OOV type are provided. Finally, additional dictionaries and named-entity information are jointly incorporated to further improve performance.

The article is organized as follows. Section 2 introduces the framework for incorporating additional information. Experiment settings are described in Section 3. Sections 4, 5, and 6 study the effect of incorporating dictionary entries, named entities, and suffix information, respectively, and Section 7 demonstrates the experiment results with a named-entity recognizer and dictionaries applied jointly. Related work is discussed in Section 8, and Section 9 concludes.

2. A UNIFIED FRAMEWORK FOR INCORPORATING ADDITIONAL INFORMATION

In this section, a character-based integrated model [Wang et al. 2012], which achieved state-of-the-art performance on the closed test, is first introduced. We then propose a general framework for incorporating additional information into this integrated model for the open test. Afterwards, the training and decoding workflows for our proposed framework are presented.

2.1. Character-Based Generative Model

The character-based approach for CWS was first proposed by Xue and Shen [2003], who considered word segmentation to be a sequence-labeling problem by assigning the corresponding position to each character in its associated word. This type of approach can be formulated as follows:

$$\bar{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | c_1^n),$$

where t_i is the position tag of character c_i within the associated word and is a member of {**B**, **M**, **E**, **S**},[4] in which **B**, **M**, and **E** denote the *beginning*, *middle*, and *end* of a *multicharacter* word, respectively, and **S** denotes that it is a *single-character* word.

$P(t_1^n | c_1^n)$ can be decomposed either discriminatively or generatively. The discriminative approach usually adopts the form

$$P(t_1^n | c_1^n) = \prod_{i=1}^{n} P\left(t_i | t_1^{i-1}, c_1^n\right) \approx \prod_{i=1}^{n} P\left(t_i | c_{i-2}^{i+2}\right),$$

[3]All resources and prior knowledge except for the provided training corpus were forbidden for the closed test, but they were allowed for the open test.

[4]There are also other tag sets [Zhao et al. 2006b]. We merely adopt the most common one.

where $P(t_i|c_{i-2}^{i+2})$ is estimated by the maximum entropy[5] (ME) approach with the following widely adopted feature templates [Ng and Low 2004]:

(a) $C_n(n = -2, -1, 0, 1, 2)$;
(b) $C_n C_{n+1}(n = -2, -1, 0, 1)$;
(c) $C_{-1} C_1$.

In contrast, Wang et al. [2009] propose a character-based generative model that decomposes $P(t_1^n|c_1^n)$ as follows:

$$P(t_1^n|c_1^n) = P([c,t]_1^n)/P(c_1^n) \approx \prod_{i=1}^{n} P([c,t]_i|[c,t]_{i-2}^{i-1})/P(c_1^n).$$

Because $P(c_1^n)$ is the same for all tag sequences, the best tag sequence can be obtained by

$$\bar{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P([c,t]_i|[c,t]_{i-2}^{i-1}), \tag{1}$$

where $P([c,t]_i|[c,t]_{i-2}^{i-1})$ is a trigram model with each unit being a pair of characters and its associated position tag.

As noted in Wang et al. [2012], these two approaches possess different error distributions and complement each other. Therefore, they can be combined via the following log-linear interpolation.[6] For the ith character in the input sentence, the score function is

$$Score(t_i) = \alpha \times \log P([c,t]_i\,|[c,t]_{i-2}^{i-1}) + (1-\alpha) \times \log P(t_i|c_{i-2}^{i+2}),$$

where α is the relative weight of the two factors and can be obtained from the development set. The best tag sequence is thus

$$\bar{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \sum_{i=1}^{n} Score(t_i).$$

To address factoids, Wang et al. [2012] further preconverted foreign letters, Arabic numbers, and Chinese numbers into their corresponding metacharacters before applying the preceding formulation. For example, "一, 5m 以下" was converted to "<CN>, <AN><FL>以下," where "CN" denotes a Chinese number, "AN" denotes an Arabic number, and "FL" denotes a foreign letter. The score function was then reformulated as follows:

$$Score(t_i) = \alpha \times \log P([u,t]_i\,|[u,t]_{i-2}^{i-1}) + (1-\alpha) \times \log P(t_i|u_{i-2}^{i+2}), \tag{2}$$

where u (a unit) denotes the corresponding character after conversion, which can be a metacharacter, a punctuation mark, or a common Chinese character. This form is used as our baseline to integrate the additional resources shown next.

2.2. The Proposed Framework

The preceding approach only relies on the training corpus and character type information. However, there is always more that we can take advantage of in practice, such

[5]According to our experiment, ME achieves similar performance as CRF on the CWS task.
[6]Although the character-tag n-gram features also can be directly integrated into the discriminative framework [Jiampojamarn et al. 2010], Wang et al. [2012] show that the log-linear interpolation is better.

as a dictionary consisting of a large word list. Such resources can be very helpful. For example, suppose the character sequence $c_i c_{i+1} c_{i+2}$ happens to be an entry in the dictionary; then the position tags for these three characters would likely be "BME," even if this word entry never appears in the training set.

Formally, hints from additional resources for character c_i can be denoted as A_i. For the preceding example, A_i can simply be the position tag of c_i in the matched dictionary entry (i.e., 'B'). For different types of OOVs (i.e., unseen dictionary words, named entities, and suffix-derived words), A_i is defined differently (see Sections 4–6 for details). Hints from additional resources can be directly encoded as features for the discriminative approach, as previous works [Zhao et al. 2006a, 2010; Zhang et al. 2014] have done.

For the generative approach, A_i can be incorporated in the following way:

$$\begin{aligned} \bar{t}_1^n &= \operatorname*{argmax}_{t_1^n} P\left(t_1^n | u_1^n, A_1^n\right) \\ &= \operatorname*{argmax}_{t_1^n} P\left([\,u,t,A]_1^n\right) / P\left([\,u,A]_1^n\right) \\ &= \operatorname*{argmax}_{t_1^n} P\left([\,u,t,A]_1^n\right). \end{aligned}$$

Then, $P\left([\,u,t,A]_1^n\right)$ can be approximated as

$$\begin{aligned} P\left([\,u,t,A]_1^n\right) &\approx \prod_{i=1}^{n} P\left([\,u,t,A]_i \,\middle|\, [\,u,t,A]_{i-2}^{i-1}\right) \\ &\approx \prod_{i=1}^{n} P\left([\,u,t,A]_i \,\middle|\, [\,u,t]_{i-2}^{i-1}\right) \\ &\approx \prod_{i=1}^{n} P\left([\,u,t]_i, M_i \,\middle|\, [\,u,t]_{i-2}^{i-1}\right) \\ &\approx \prod_{i=1}^{n} P\left(M_i | c_{i-2}^{i}\right) \times P\left([\,u,t]_i \,\middle|\, [\,u,t]_{i-2}^{i-1}\right), \end{aligned} \tag{3}$$

where M_i is the tag-matching status used to check whether the position tag t_i (assigned to u_i) is consistent with the preference implied by A_i, and it is a member of {match[A_i], violate[A_i], neutral}, where "neutral" is reserved for the case that A_i is either invalid or unreliable. For the simple case mentioned previously, M_i would be "match[B]" if t_i were "B", otherwise M_i would be "violate[B]."

There are two factors in the last line of Equation (3). The second factor is the character-tag trigram factor of the original generative model, and the first is a tag-matching factor that reflects the likelihood that the tag assigned to the character will match the given additional information. This factor is mainly introduced to give guidance if the original generative trigram factor cannot give a reliable prediction when the associated character-tag trigram (or even bigram) is unseen in the training corpus. It is reasonable to assert that these two factors should be weighted differently. Therefore, the score function of the enhanced generative model would be

$$Score(t_i) = (1-\alpha) \times \log P\left([\,u,t]_i \middle| [\,u,t]_{i-2}^{i-1}\right) + \alpha \times \log P\left(M_i | u_{i-2}^{i}\right). \tag{4}$$

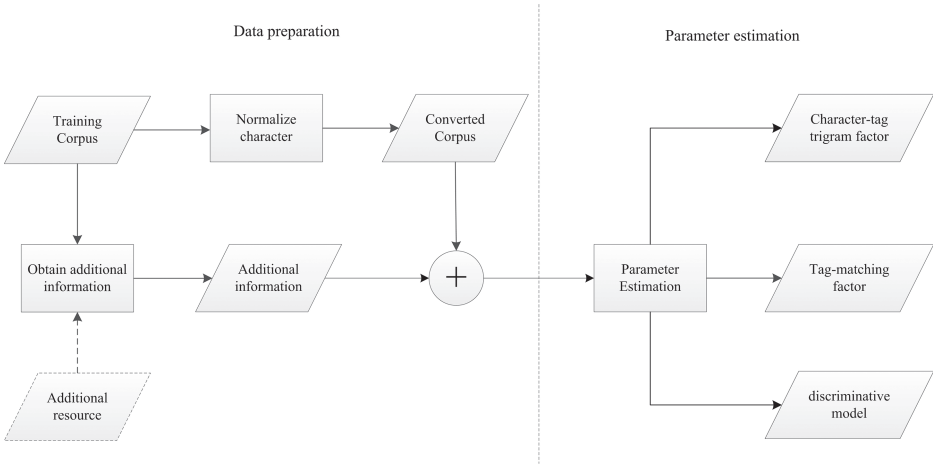

Fig. 1. Training workflow for the proposed framework.

After incorporating the additional information into both the generative and discriminative approaches, we can obtain our enhanced integrated approach via log-linear interpolation. The new score function is thus

$$\begin{aligned} Score(t_i) \;=\; & \beta \times \left((1-\alpha) \times \log P\left([u,t]_i \middle| [u,t]_{i-2}^{i-1} \right) + \alpha \times \log P\left(M_i \middle| u_{i-2}^{i} \right) \right) \\ & + (1-\beta) \times \log P\left(t_i \middle| u_{i-2}^{i+2}, A_i \right). \end{aligned} \tag{5}$$

When there is more than one type of additional information, the score function can be derived similarly. For example, suppose that there are two types of additional information (denoted by A_1, A_2); in that case, the function would be

$$\begin{aligned} Score(t_i) = \beta \times & \left((1-\alpha_1-\alpha_2) \times \log P\left([u,t]_i \middle| [u,t]_{i-2}^{i-1} \right) + \alpha_1 \times \log P\left(M1_i \middle| u_{i-2}^{i} \right) \right. \\ & \left. + \alpha_2 \times \log P\left(M2_i \middle| u_{i-2}^{i} \right) \right) + (1-\beta) \times \log P\left(t_i \middle| u_{i-2}^{i+2}, A1_i, A2_i \right). \end{aligned} \tag{6}$$

2.3. Training and Decoding Workflow

The training workflow for the proposed framework is described in Figure 1. First, numbers and letters are preconverted into their corresponding metacharacters. At the same time, additional information is obtained for each character in the training sentence. Additional resources could be required in this process. For example, a dictionary (or a named entity recognizer) could be adopted to obtain related information. Afterwards, the factors/models could be trained on the converted corpus with the additional information.

For decoding, similar to the data preparation part of the training procedure, the characters in the testing sentence are first normalized and then related information is obtained. The best tag sequence is then found via beam search, in which the score function relies on the models obtained in the training phase.

3. SHARED EXPERIMENT SETTINGS AND BASELINE DESCRIPTION

Because experiment settings and the baseline system are shared by various experiments with different types of additional resources, they are first described as follows.

Table I. Corpus Statistics for SIGHAN-2005

Corpus	Training Set		Testing Set		Testing Set
	tokens	types	tokens	types	OOV Rate
AS	5.45M	141K	122K	19K	0.046
CITYU	1.46M	69K	41K	9K	0.074
MSR	2.37M	88K	107K	13K	0.026
PKU	1.1M	55K	104K	13K	0.058

Table II. Corpus Statistics for CIPS-SIGHAN-2010

Corpus	Domain	Tokens	Types	OOV Rate
Training	**News (PKU)**	1.1M	55K	NA
Testing	**Literature**	36K	6.4K	0.069
	Computer	35K	4.2K	0.152
	Medicine	31K	5.1K	0.110
	Finance	33K	4.9K	0.087

Table III. Distribution of OOV Words over Various Types in the SIGHAN 2005 Testing Corpora

OOV type	PKU	AS	CITYU	MSR	Overall
Unseen Dict. Words[1]	2,369 (65.6%)	1,853 (34.9%)	1,412 (46.6%)	928 (32.8%)	6,562 (44.4%)
NEs	532 (14.7%)	1,098 (20.7%)	464 (15.3%)	959 (33.9%)	3,053 (20.7%)
Factoids	388 (10.7%)	985 (18.6%)	190 (6.3%)	826 (29.2%)	2,389 (16.2%)
Suffix-derived Words	129 (3.6%)	637 (12.0%)	178 (5.9%)	10 (0.4%)	954 (6.5%)
Inconsistency	32 (0.9%)	221 (4.2%)	481 (15.9%)	35 (1.2%)	769 (5.2%)
Others	159 (4.4%)	581 (11.0%)	301 (9.9%)	71 (2.5%)	1,112 (7.5%)
Total	3,611 (100%)	5,308 (100%)	3,028 (100%)	2,829 (100%)	14,776 (100%)

3.1. Datasets Adopted

All of the experiments in Sections 4 through 6 are conducted on the corpora provided by SIGHAN-2005 [Emerson 2005] and CIPS-SIGHAN-2010 [Zhao and Liu 2010], which have been widely adopted in various papers for comparing performance. There are four corpora (each includes both training and testing sets) in SIGHAN-2005: the Academia Sinica Corpus (AS), the City University of Hong Kong Corpus (CITYU), the Microsoft Research Corpus (MSR), and the Peking University Corpus (PKU). These corpora are used to compare performance on the in-domain tests. In contrast, CIPS-SIGHAN-2010 only provides one training set—the same one as the PKU training set from SIGHAN-2005. Additionally, there are four testing sets from different domains: Literature, Computer, Medicine, and Finance. These corpora are used to compare performance on cross-domain tests. The statistics of the SIGHAN-2005 and CIPS-SIGHAN-2010 corpora are shown in Tables I and II, respectively.

To obtain the distribution of various OOV types, we manually classify the OOV words in the SIGHAN-2005 testing corpora into the six types defined in the Introduction. The distributions of various OOV types are shown in Table III.

3.2. Parameter Estimation and Evaluation Metric

For the generative model, the SRI Language Model Toolkit (SRILM)[7] [Stolcke 2002] is used to train $P\left([u,t]_i \,|\, [u,t]_{i-2}^{i-1}\right)$ with modified Kneser-Ney smoothing [Chen and Goodman 1998]. The Factored Language Model [Bilmes and Kirchhoff 2003] in the SRILM is adopted to train $P\left(M_i | u_{i-2}^{i}\right)$, and this factor sequentially backs off to $P(M_i)$.

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

Table IV. Distribution of OOV Error Types of the Baseline System in SIGHAN 2005 Testing Corpora

OOV type	PKU	AS	CITYU	MSR	Overall
Unseen dict. Words	975 (74.3%)	641 (34.5%)	397 (42.8%)	339 (41.1%)	2,352 (47.8%)
NEs	128 (9.7%)	332 (17.9%)	121 (13.0%)	366 (44.4%)	947 (19.2%)
Suffix-derived words	60 (4.6%)	323 (17.4%)	94 (10.1%)	6 (0.7%)	483 (9.8%)
Factoids	21 (1.6%)	210 (11.3%)	41 (4.4%)	60 (7.3%)	332 (6.7%)
Inconsistency	31 (2.4%)	107 (5.8%)	143 (15.4%)	8 (1.0%)	289 (5.9%)
Others	98 (7.5%)	246 (13.2%)	132 (14.2%)	45 (5.5%)	521 (10.6%)
Total	1,313 (100%)	1,859 (100%)	928 (100%)	824 (100%)	4,924 (100%)

Note: The corresponding ratios among the total OOV errors in each column are shown within parentheses.

For the discriminative factor, Zhang's ME Package[8] is adopted to train $P(t_i|u_{i-2}^{i+2}, A_i)$, and the training is conducted with Gaussian prior 1.0 and 300 iterations.

In addition, to obtain the weights of different factors in Equations (2), (4), (5) and (6) for each corpus provided by the SIGHAN bakeoffs, we randomly select 1% of the sentences from the training set as its corresponding development set; the remainder is used as the new training set. Afterwards, MERT [Och 2003] is adopted to determine the best weights of the different factors on the corresponding development sets previously mentioned. After the weights are obtained, the full training set is used again to train the final model for each corpus.

For performance evaluation, the following metrics are adopted: *Precision* (P), *Recall* (R), and *F-score* (F); the F-score is calculated as $F = 2PR/(P + R)$. Finally, we followed Zhang et al. [2004] in conducting the statistical significance tests with 2,000 resampling size and 95% confidence interval. In the following tables, an asterisk is used to indicate that the difference between the proposed approach and the corresponding baseline system (or the systems being compared) is statistically significant.

3.3. Baseline Model

The integrated model proposed in Wang et al. [2012], listed in Equation (2), is adopted as our baseline model because our models are derived based on it. To show the distribution of the various OOV error types in the baseline system, we give the error number of each OOV type in Table IV. Their corresponding ratios among total OOV errors in each corpus are shown within parentheses.

Table IV shows that the OOV errors have different distributions across various corpora. For example, there were 44% NE errors among all OOV errors in the MSR corpus, whereas this ratio is less than 18% in the other corpora. Overall, unseen dictionary words, named entities, and suffix-derived words form the majority of the OOV errors (following the pattern of OOV words observed in Table III after the factoids are excluded, which implies that they were not all wellhandled), and they are addressed separately in the following sections.

4. HANDLING UNSEEN DICTIONARY WORDS

Unseen dictionary words, especially new terms and unseen idioms, are very difficult to recognize because their associated character-tag n-grams within words are hardly seen in the training corpus, especially when the training corpus is from another domain. For example, for the medical term "氨基酸" (amino acid), its character bigrams "氨基" and "基酸" can hardly be found in the news domain corpus. Furthermore, the position tag of a character in a term or an idiom is frequently inconsistent with the distribution

[8]http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html.

learned from the training corpus. For example, the character '基' is always the beginning character in common words such as "基本" (basic) and "基础" (base). However, it is the middle character in the previously mentioned term.

Fortunately, idioms are not productive, and most of them can be found in a general dictionary; additionally, many technical terms can also be found in a corresponding domain dictionary. Therefore, they can be considerably covered by dictionaries, even when these terms are not seen in the training set. In this section, we first define additional information and the corresponding matching status for adopted dictionaries. Afterwards, our unified framework is instantiated with the additional dictionary information.

4.1. Additional Information for In-Dictionary Words

Usually, a given character in a sentence might be covered by more than one dictionary word. However, for simplicity, previous approaches [Zhao et al. 2006a, 2010] only consider the longest one. Such simplification will lose information when there are ambiguities, and we thus propose the following feature to denote the ambiguity status of each character.[9]

Let c_i be the ith character in a given sentence. To establish whether there were ambiguities (and what types of ambiguities) with those dictionary-matching words at c_i, we propose the *dictionary coverage status* feature, which is a member of {No-Dictionary-Word, No-Ambiguity, Crossing-Ambiguity, Including-Ambiguity, Mixed-Ambiguity}and is defined here. This status depends only on the given sentence and the dictionary and is irrelevant to the position tag assigned to the character. Let $\mathcal{D}$ be the given dictionary that only contains multicharacter words and $c_{[i:j]}$ denote the string from c_i to c_j (including c_j), the conditions for "Including-Ambiguity" (which implies that one dictionary word is included in another dictionary word for the given string) and "Crossing-Ambiguity" (which implies that one dictionary word is overlapped by another dictionary word for the given string) are defined next.

(A) *Conditions for Including-Ambiguity (IA)*

(1) Both c_i and c_{i+1} will be assigned "IA"[10] if they meet the following condition (Figure 2(a)):

$$\exists\ j,l>0,k\geq j:\left\{c_{[i-j:i]},c_{[i-k:i+l]}\right\}\subset\mathcal{D}.$$

(2) Both C_{i-1} and c_i will be assigned "IA" if they meet the following condition (Figure 2(b)):

$$\exists\ j,k>0,l\geq j:\left\{c_{[i:i+j]},c_{[i-k:i+l]}\right\}\subset\mathcal{D}.$$

(B) *Conditions for Crossing-Ambiguity (CA)*.

(1) Both c_i and c_{i+1} are assigned "CA" if they meet the following condition (Figure 3(a)):

$$\exists\ j,l>0,0\leq k<j:\{c_{[i-j:i]},c_{[i-k:i+l]}\}\subset\mathcal{D}.$$

(2) Both c_{i+1} and c_i are assigned "CA" if they meet the following condition (Figure 3(b)):

$$\exists\ j,k>0,0\leq l<j:\{c_{[i:i+j]},c_{[i-k:i+l]}\}\subset\mathcal{D}.$$

[9]Please note that ambiguity status is traditionally defined on words, but ours is defined on characters.

[10]We define the ambiguity status for the two characters on the boundary because their position tag will be different when a longer or shorter dictionary entry is matched.

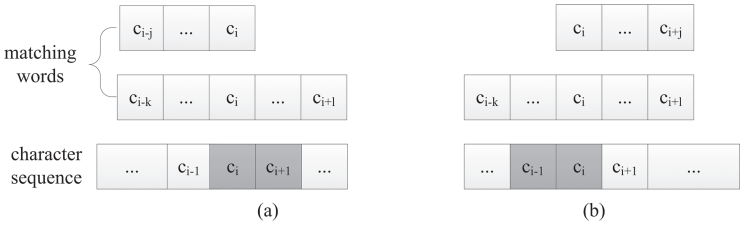

Fig. 2. Cases for including-ambiguous characters (marked in grey).

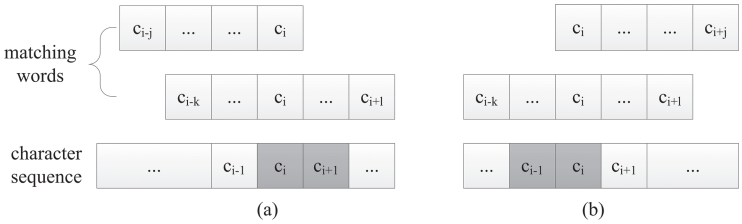

Fig. 3. Cases for crossing-ambiguous characters (marked in grey).

Dictionary Coverage Status at c_i (denoted by DC_i can then be determined as follows:

$$DC_i = \begin{cases} \text{No-Dictionary-Word} & \text{if no matching word is found;} \\ \text{Including-Ambiguity} & \text{if only (A) is satisfied;} \\ \text{Crossing-Ambiguity} & \text{if only (B) is satisfied;} \\ \text{Mixed-Ambiguity} & \text{if both (A)and (B) are satisfied;} \\ \text{No-Ambiguity} & \text{otherwise.} \end{cases}$$

This definition implicitly implies that a character that possesses the same position tag for all associated dictionary matching-words will be assigned "No-Ambiguity." For example, given a character sequence "大学生物" (university biology) and a set of dictionary-matching words {"大学" (university), "大学生" (undergraduate)}, for characters '学' and "生," Condition (A.1) is satisfied but Condition (B) is not; therefore, DC_2 and DC_3 should be set to "Including-Ambiguity." In contrast, if the dictionary-matching-words are {"大学生", "生物" (biology)}, then Condition (B.1) is satisfied but Condition (A) is not; DC_2 and DC_3 should thus be set to "Crossing-Ambiguity." However, if we have all three matching words {"大学","大学生","生物"}, then both Conditions (A.1) and (B.1) are satisfied; therefore, DC_2 and DC_3 should be set to "Mixed-Ambiguity" in this case. Furthermore, if the matching words are {"大学","生物"}, then DC_2 and DC_3 would be "No-Ambiguity."

4.2. Instantiating the Unified Framework

With the preceding definition for dictionary coverage status, the additional information A_i and tag-matching status M_i for dictionary words are represented as $A_i = D_i \triangleq (DC_i, MWL_i)$ and $M_i \triangleq MD_i$ (which is the tag-matching status for dictionary-matching words and will be specified later), where MWL_i denotes the *maximum word length* of those words that cover c_i.

The score function of the generative model in Equation (4) will then be instantiated as

$$Score(t_i) = (1-\alpha) \times \log P\big([u,t]_i \,|\, [u,t]_{i-2}^{i-1}\big) + \alpha \times \log P\big(MD_i | u_{i-2}^{i}\big), \tag{7}$$

where MD_i is the tag-matching status of t_i to D_i. Because more than one word could have covered c_i, MD_i is a member of {MatchLongest(D_i), MatchShorter(D_i), MatchNone(D_i), Neutral}. Denote the set of dictionary-matching words that begin with c_i as $\mathcal{W}_B = \{c_{[i:j]} | c_{[i:j]} \in \mathcal{D}\}$, enclose c_i as $\mathcal{W}_M = \{c_{[j:k]} | c_{[j:k]} \in \mathcal{D}, j < i < k\}$, and end with c_i as $\mathcal{W}_E = \{c_{[m:i]} | c_{[m:i]} \in \mathcal{D}\}$. If c_i is tagged as t_i, then MD_i can be decided as follows.

(1) If $\mathcal{W}_\mathrm{B} \bigcup \mathcal{W}_\mathrm{M} \bigcup \mathcal{W}_\mathrm{E} = \varnothing$, which indicates that this character is not covered by any dictionary word, then $\mathrm{MD_i}$ is set to "Neutral."
(2) If $\mathcal{W}_{t_i} = \varnothing$ and $\mathcal{W}_\mathrm{B} \bigcup \mathcal{W}_\mathrm{M} \bigcup \mathcal{W}_\mathrm{E} = \varnothing$ (where $\mathcal{W}_{t_i}$ is the set of dictionary-matching words corresponding to $\mathrm{t_i}$; for example, $\mathcal{W}_{t_i}$ will be $\mathcal{W}_\mathrm{B}$, if $\mathrm{t_i}$ is 'B'. Please note that $\mathcal{W}_\mathrm{S} = \varnothing$ (because the adopted dictionary only contains multicharacter words), which indicates that the assigned tag does not follow any dictionary-matching word, then $\mathrm{MD_i}$ is set to "MatchNone($\mathrm{D_i}$)."
(3) If $\forall \mathrm{w} \in (\mathcal{W}_\mathrm{B} \bigcup \mathcal{W}_\mathrm{M} \bigcup \mathcal{W}_\mathrm{E}), \exists \mathrm{w}' \in \mathcal{W}_{\mathrm{t_i}} : \mathrm{len(w')} \geq \mathrm{len(w)}$, then $\mathrm{MD_i}$ is set to "MatchLongest($\mathrm{D_i}$)," indicating that the assigned tag matches the corresponding position tag of the longest dictionary matching word at that character.
(4) Otherwise, is set to "MatchShorter($\mathrm{D_i}$)," indicating that the assigned tag does not match the corresponding position tag of the longest dictionary-matching word at that character but matches that of some shorter words.

For example, when we consider the second character '学' in the sequence "大学生" and assume that the dictionary-matching words are {"大学" (university), "大学生" (university student)}, if the position tag assigned to '学' is 'M,' then the corresponding *MD* will be "MatchLongest(Including-Ambiguity, 3);" if it is 'E,' then the *MD* will be "MatchShorter(Including-Ambiguity, 3);" and if it is 'B' or 'S,' the *MD* will be "MatchNone(Including-Ambiguity, 3)." Therefore, this candidate feature is associated with each candidate of the position tag. However, if no dictionary word covers this character, then the *MD* will be set to "Neutral" regardless of which tag is assigned to '学.'

For the discriminative approach, the following two feature templates proposed in Zhao et al. [2006a] are added to the commonly adopted feature templates (a) to (c) described in Section 2.1:

(d) MWL_i, dt_i;
(e) $C_k, dt_i (k = i-1, i, i+1)$.

Let W denote the longest of those dictionary words that cover c_i, in which case MWL_i denotes the length of W and dt_i denotes the corresponding tag of c_i in W.

The score function of the integrated model in Equation (5) would then be instantiated as

$$\begin{aligned} Score(t_i) \;=\; & \beta \times \Big((1-a) \times \log P\big([u,t]_i \,\big|\, [u,t]_{i-2}^{i-1}\big) + a \times \log P\big(MD_i \big| u_{i-2}^{i}\big)\Big) \\ & + (1-\beta) \times \log P\big(t_i \big| u_{i-2}^{i+2}, MWL_i, dt_i\big), \end{aligned} \tag{8}$$

where α and β are two weighting coefficients to be decided from the development set.

To train the factors in Equation (8), we need a dictionary during training. For the tag-matching factor of the generative model, this dictionary could just be the set of all of the IV words that appeared in the training set (because those words not in the dictionary are implicitly ignored by this factor, they do not need to be considered during

training). However, for the discriminative model, some IV words (those occurring fewer than six times in our experiments) have to be excluded from the dictionary for training (because those words not in the dictionary are not ignored by this model, they had to be considered during training); otherwise, the condition of the testing set would seriously mismatch that of the training set. Similarly, the dictionaries used for the tag-matching factor and the discriminative model are also different during testing. For the tag-matching factor, only OOV words should be included in the dictionary,[11] whereas for the discriminative model, the dictionary needs to include both OOV and IV words.[12]

4.3. Experiments

Because no dictionary can cover all OOV words for real applications, we want to know how the generative, discriminative, and integrated models with dictionary information would perform under different OOV coverage rates. Table V gives the results for different OOV coverage rates with $\alpha = \beta = 0.5$ (weights in Equations (7) and (8)) for simplicity; the first row "Baseline" denotes the original character-based approaches introduced in Section 2.1, with preconverting specific types of characters into their corresponding metacharacters.

When only the training words are contained in the dictionary, it can be seen that the performances of the discriminative and integrated approaches with dictionary information are inferior to that of the baselines (0%OOV vs. Baseline), the reason being that the new dictionary-related features in the discriminative model prefer dictionary words and the original features (adopted in the baseline) prefer IV words. Because the dictionary words are simply IV words in this case, those IV words would have been over-emphasized and would thus have hurt the OOV word performance. For example, the new discriminative model incorrectly splits the unseen word "中国社会学会" (Chinese Sociological Association) into "中国/社会学会," because the word "中国" is both an IV word and a dictionary word, it is thus preferred by the new model. On contrary, the original discriminative model recognizes this word correctly. For the integrated model, because the weight between the generative and the discriminative model is not readjusted, it is also influenced by this deterioration effect. Therefore, similar errors can also be found. However, because only OOV words are used during testing for the generative model, its performance is not affected in this case.

When the dictionary begins to cover OOV words, the performances of the three models rises sharply. In any event, all of the proposed models outperform the original models when the dictionary covers only 20% of the OOV words. This condition is easy to satisfy in real scenarios. In addition, the weights in Equations (7) and (8) can also be adjusted to control the degree of preference for dictionary information.

Nonetheless, not every model possesses the robustness with varying dictionary coverage rates. For example, the corresponding result of the word-based generative trigram model,[13] given in the last column of Table V, shows that the trigram model is

[11]The first factor (character-tag trigram) in Equation (8) prefers IV words, but the second factor (for tag matching) prefers dictionary words. If those IV words were also put into the dictionary, then the IV words would be over-emphasized, which would hurt the performance of handling OOV words when the OOV coverage rate is less than 80%, which is higher than what we can usually achieve (please see Tables VII and VIII).

[12]For the discriminative approach, we also tried using a dictionary that contained only OOV words during testing. However, the performance with this strategy was much worse than that of the case adopted in this article because the conditions of the training and testing sets were also seriously mismatched.

[13]This well-known model adopts the form $w_{seq} = \arg\max \prod_{i=1}^{m} P(w_i|w_{i-2}^{i-1})$ [Wang et al. 2012].

Table V. F-scores versus Different OOV Coverage Rates for the Proposed Generative, Discriminative, Integrated Models and the Word-Based Trigram Approach

Dict.	Generative	Discriminative	Integrated	Word-based
Baseline	0.956	0.952	0.961	0.932
0%OOV	0.956	0.951	0.957	0.932
20%OOV	0.960	0.957	0.963	0.942
40%OOV	0.964	0.964	0.968	0.952
60%OOV	0.968	0.970	0.973	0.961
80%OOV	0.971	0.976	0.979	0.970
100%OOV	0.975	0.983	0.984	0.980

Note: (Overall performances on various SIGHAN-2005 testing sets). Baseline: without utilizing dictionary information, corresponding to the original character-based approaches introduced in Section 2.1.

Table VI. Descriptions for the Adopted Dictionaries

Name	Description	#words
PKU word list	A word list extracted from a corpus from Peking University, consisting of words of length up to four characters	10.8K
Wikipedia titles	Titles of Chinese Wikipedia articles	914K
Wiktionary words	A collaborative project to produce a free-content Chinese dictionary	277K
Zdict	A popular online Chinese dictionary	432K
MOE dictionary	An online dictionary provided by Ministry of Education, R.O.C.	151K

quite brittle in comparison with our model. In this model, all words kept in the dictionary are used to construct the word lattice in the decoding process. Those OOV words are treated as unseen events and given a very low score. However, it can be seen that although the results with the full dictionary are satisfactory, the performance drops rapidly when the OOV coverage rate decreases. This indicates that this model is quite sensitive to the coverage rate of OOV words because of its inability to identify OOV words that are not in the dictionary. This model is thus not useful for real applications because it is impossible to know the corresponding dictionary coverage rate in the testing set in advance. Therefore, determining the robustness of dictionary-based models for different dictionary coverage rates is important in selecting an appropriate model.

To give the performance with a true dictionary coverage rate, several publicly-accessible dictionaries are obtained first for open comparison. These dictionaries are downloaded from the Internet,[14] and they are described briefly in Table VI. For simplicity, we adopt the same dictionary (a combination of these dictionaries) for all testing corpora. Because the PKU and MSR corpora are in simplified Chinese, whereas the AS and CITYU corpora are in traditional Chinese, we adopt Open Chinese Convert[15] to convert the dictionary between simplified and traditional Chinese. This external dictionary contains approximately 1.4M words in total and covers 78%, 44%, 54%, and 51% of the OOV words (excluding factoids) in the four testing sets (PKU, AS, CITYU, and MSR) from SIGHAN-2005. Because external dictionaries are expected to be collected by the user in real applications, dictionary words should be consistent with the user's own segmentation criterion. Therefore, to give a true evaluation that reflects the actual situation, words in the dictionary are first transformed into their corresponding variations according to the same criteria adopted in the various given corpora. For

[14]http://ccl.pku.edu.cn/alcourse/nlp/2010/word_freq_list.rar; http://zh.wikipedia.org/; http://zh.wiktionary.org/; http://www.zdic.net/; http://dict.revised.moe.edu.tw/.
[15]https://code.google.com/p/opencc/.

Table VII. Performances in F-Score of the Proposed Generative, Discriminative, and Integrated Approaches with Dictionary Information

Approaches		PKU (78%)	AS (44%)	CITYU (54%)	MSR (51%)	Overall (56%)
Generative	Baseline	0.953	0.950	0.944	0.971	0.956
	+Dict.	**0.967***	**0.958***	**0.961***	**0.976***	**0.966***
Discriminative	Baseline	0.947	0.953	0.943	0.960	0.952
	+Dict.	**0.967***	**0.958***	**0.965***	**0.976***	**0.966***
Integrated	Baseline	0.957	0.957	0.954	0.972	0.961
	+Dict.	**0.977***	**0.962***	**0.969***	**0.981***	**0.972***

Note: Baseline: the corresponding baseline approach described in Section 2.1. Overall: the overall performance of those four corpora. The ratios of OOV covered by the dictionary for each corpus are also shown within parentheses below the corpus name.

Table VIII. Performances in F-Score of the Proposed Integrated Approach for Cross-Domain Tests

Approaches	Literature (73%)	Computer (66%)	Medicine (75%)	Finance (49%)	Overall (67%)
Baseline	0.932	0.941	0.918	0.956	0.937
+Dict.	**0.965***	**0.965***	**0.959***	**0.973***	**0.966***

Note: Dictionary OOV coverage rates for each corpus are also shown within parentheses below the corpus name.

example, "音乐电视" (music television) was converted into "音乐" (music) and "电视" (television) in the MSR corpus according to its adopted gold criterion. Also, best weights (α and β) are obtained on the development set.

To compare the proposed models with the baseline systems described in Section 2.1 more realistically, we first show the in-domain test results of the generative, discriminative, and integrated models with the preceding external dictionary on all of the SIGHAN-2005 testing corpora in Table VII.[16] In the table, the boldface indicates the best results, and it holds for the rest of this article as well. It can be seen that dictionary information results in significant improvement in all models, and larger improvements can be achieved with larger OOV coverage rates, which is consistent with the results in Table V.

In addition to these experiments conducted on in-domain corpora, we test the performance of our proposed integrated approach on cross-domain corpora provided by CIPS-SIGHAN-2010. Compared with in-domain tests, the OOV rates are always higher on the cross-domain tests (cf. Tables I and II), mainly because there are many more domain-specific terms in cross-domain corpora. Based on the experiment results shown in Table VIII for cross-domain tests, we can see that dramatic improvements can be brought about by utilizing dictionary information. Overall, the proposed integrated approach gives 0.029 improvement in F-score for cross-domain tests while only giving 0.011 for the preceding in-domain tests, which again demonstrates the importance of the OOV coverage rate of the adopted dictionary.

4.4. Error Analysis

To further establish the effectiveness of our integrated approach on the unseen dictionary words, Table IX gives the number of unseen dictionary word errors generated by the corresponding baseline system for each SIGHAN-2005 corpus and the number of

[16]Please note that our results for the new discriminative model (0.967) are slightly different from that reported in Low et al. [2005], (0.965), which only adopted the first dictionary mentioned.

Table IX. Error Reduction for Unseen Dictionary Words using the Proposed Integrated Approach

#err for unseen dictionary words	PKU	AS	CITYU	MSR	Overall
Baseline	975	525	397	339	2236
+Dict. System	172 (82%)	122 (77%)	47 (88%)	49 (86%)	390 (83%)

Note: The percentages within parentheses show the corresponding error reduction rates.

remaining unseen dictionary word errors of our integrated approach. The corresponding error reduction rates for the unseen dictionary words (within parentheses) show that the proposed approach is very effective for this type of OOV error.

The remaining unseen dictionary word errors are mainly attributable to the fact that those words also consist of other shorter, high-frequency words. For example, the word "电子商务" (e-Commerce) contains two shorter words, "电子" (electronic) and "商务" (commerce), which are common in the training corpus; thus, it is preferable to split this unseen word with those character n-grams, although this word is contained in the dictionary.

5. HANDLING OOV NAMED ENTITIES

The named entities that we address here include person names, location names and organization names. Unlike terms and idioms, named entities are quite productive, and they cannot be considerably covered by a dictionary. In addition, the statistical correlation between characters within named entity is usually very weak. For example, the character bigram "扬子" in the location name "扬子江" (Yangzi River) is unlikely to be seen in other words. Furthermore, location names and organization names can even be nested; for example, the location name "宋庆龄故居" (Former Residence of Song Ching Ling) contains another person name "宋庆龄" (Song Ching Ling), and the organization name "中国科学院" (Chinese Academy of Sciences) also contains another location name "中国" (China). As a result, recognizing named entities with only character n-grams is very difficult.

Fortunately, because it is one of the key steps in fields such as information extraction, question answering, and machine translation, named-entity recognition has been well studied with features much richer than character n-grams [Nadeau and Sekine 2007; Sun et al. 2002]. Thus, we propose taking advantage of these existing named-entity recognizers that could be obtained off the shelf.

5.1. Incorporating a Named-Entity Recognizer

We choose the named-entity recognizer provided by Zhao Hai[17] (BaseNER), which received three second-place rankings and one third-place ranking in the four named-entity recognition tasks in the Fourth SIGHAN Bakeoff [Jin and Chen 2008]. Aside from those character n-gram features, unsupervised segmentation outputs [Feng et al. 2004], assistant NE recognizers,[18] and additional NE lists are utilized in this approach [Zhao and Kit 2008] to improve performance. After the named entities in the sentence are identified, additional named-entity information for each character is instantiated as a tuple $A_i = NE_i \triangleq (NEtag_i, NEtype_i, NElen_i)$, where $NEtag_i$ denotes the position tag of c_i in the associated NE, which is a member of {B, M, E, S}as adopted in CWS; $NEtype_i$ denotes the type of the NE, which is a member of {PER, LOC, ORG}; and $NElen_i$ denotes the length of the NE, which is a positive integer. In contrast,

[17] http://bcmi.sjtu.edu.cn/~zhaohai/downloads/baseNER-pub-20080601.rar.

[18] Assistant NE recognizers are NE recognizers trained on extra NE annotated training data besides those provided by the bakeoff.

Table X. Performance of the Adopted Named-Entity Recognizer (BaseNER)

	PKU	AS	CITYU	MSR
Precision	0.612	0.919	0.939	0.887
Recall	0.323	0.578	0.674	0.685

if the character is not in any NE, its associated NE_i is simply represented as "Not Applicable (NA)."

With this additional named-entity information, the score function of the generative model in Equation (4) is instantiated as

$$Score(t_i) = (1-\alpha) \times \log P([u,t]_i \,|\, [u,t]_{i-2}^{i-1}) + \alpha \times \log P(MNE_i | u_{i-2}^{i}), \tag{9}$$

where MNE_i is the tag-matching status of t_i to NE_i, and it is a member of $\{\text{Match}[NE_i], \text{Violate}[NE_i], \text{Neutral}\}$. If $NE_i = NA, MNE_i$ is assigned as "Neutral;" if $t_i = NEtag_i$, MNE_i is assigned as "Match[NE_i];" otherwise MNE_i is assigned as "Violate[NE_i]." For example, assume $NE_i = (\text{B}, \text{PER}, 3)$ (which implies that u_i is the beginning of a 3-character person name) and the tag assigned to u_i is "B," MNE_i would be "Match[B, PER, 3]," and if u_i is assigned with other tags, MNE_i would be "Violate[B, PER, 3]."

For the discriminative approach, considering that named entities only account for a small portion of the whole training data, we propose training two discriminative models[19] and then combine them for the purpose of smoothing: one is the model trained with feature templates (a) to (c) described in Section 2.1, and the other is trained on those instances where c_i is in a recognized named entity, with the following feature template in addition to templates (a) to (c).

(d) $NEtag_i, NEtype_i, NElen_i$.

The generative model and discriminative model are then combined via log-linear interpolation.

5.2. Experiments

Table X shows the performance of BaseNER (the adopted named-entity recognizer) on the OOV named entities. Please note that the performances are calculated according to various word segmentation benchmarks adopted in the various corpora previously mentioned. If a named entity recognized by BaseNER is consistent with the gold segmentation, it is considered correct; otherwise, it is considered incorrect. In addition, for simplicity, the recall rate here is calculated based only on those OOV named entities. It can be seen from Table X that the performance on the PKU corpus is much lower than that on the others, the main reason being the criterion mismatch between BaseNER and PKU benchmarks. For example, the surname and the given name in a Chinese person name are considered two words in the PKU corpus, such as "宋/庆龄," whereas BaseNER takes "宋庆龄" to be just one word.

To demonstrate the effectiveness of our approach, forced-decoding (i.e., forcing the decoder to choose the recognized named entity) is adopted for performance comparison. The experiment results of the generative, discriminative, and integrated approaches

[19] In fact, similar performance can be obtained by training only one discriminative model with the additional NE feature, but this approach would be less effective in reducing NE errors. Thus, we adopted the proposed strategy.

Table XI. Performance of the Proposed Generative, Discriminative, and Integrated Approaches with NE Information

Approaches		PKU	AS	CITYU	MSR	Overall
Generative	Baseline	0.953	0.950	0.944	0.971	0.956
	Forced-decoding	0.926	0.952	0.951	0.973	0.951
	+ NE	**0.954***	**0.952***	**0.951***	**0.975***	**0.959***
Discriminative	Baseline	0.947	0.953	0.943	0.960	0.952
	Forced-decoding	0.919	0.953	0.943	0.961	0.945
	+ NE	**0.949***	**0.953**	**0.945***	**0.962***	**0.954***
Integrated	Baseline	0.957	0.957	0.954	0.972	0.961
	Forced-decoding	0.928	0.958	0.956	0.973	0.954
	+ NE	**0.958***	**0.958***	**0.957***	**0.974***	**0.962***

Note: Baseline: the corresponding baseline approaches described in Section 2.1; Forced-decoding: forcing the decoder to choose the recognized NEs. An asterisk indicates that the difference between the proposed approach and the corresponding baseline system was statistically significant.

Table XII. Performances of the Proposed Integrated Approach in F-Score for Cross-Domain Tests

Approaches	Literature	Computer	Medicine	Finance	Overall
Baseline	0.932	0.941	0.918	0.956	0.937
+NE	0.932	0.941	0.918	0.956	0.937

are given in Table XI, where "Baseline" denotes the corresponding generative, discriminative, and integrated approaches mentioned in Section 2.1.

By comparing the forced-decoding approach with the baseline approach, we find that it is harmful for performance when the standards of BaseNER and gold segmentation are considerably inconsistent (see the performance of the PKU corpus in Table XI). Furthermore, even with the same standard, the improvement is still not significant because force-decoding inherits the same errors made by the named-entity recognizer. On the contrary, if the named-entity information is encoded as features and considered during training (as in the proposed approach), both of these problems can be considerably immunized by the learned model if similar cases also occur in the training corpus. Furthermore, the discriminative model seemed less effective than the generative approach, because the baseline discriminative model handles OOV better than the baseline generative model, that is, there were fewer OOV NE errors for the baseline discriminative approach.

We also test the performance of our proposed integrated approach on the cross-domain corpora provided by CIPS-SIGHAN-2010. It can be seen from Table XII that the proposed approach does not bring any improvement for cross-domain tests, one of the reasons being that there are many fewer named entities in the four cross-domain corpora than in the news (in-domain) corpora. Another reason (confusing foreign names with Chinese names) will be studied in the next section.

5.3. Error Analysis

To further investigate the impact of incorporating the named-entity information, Table XIII gives the number of OOV NE errors generated by the baseline integrated approach, the number of those NE errors that could be recognized by the adopted NE recognizer, and the number of OOV NE errors that remain in our proposed integrated approach for each SIGHAN-2005 corpus. The corresponding error reduction rates are also given in Table XIII (within parentheses); they were calculated using (#err-baseline - #err-with-NE) / #Recognized-NE (e.g., $31\% = [\,(128-112)/52]\,$), showing the effectiveness of the proposed approach for this type of OOV error.

Table XIII. Error Reduction for OOV Named Entities using the Proposed Integrated Approach

#err for OOV NE	**PKU**	**AS**	**CITYU**	**MSR**	**Overall**
Baseline	128	291	121	366	906
#Recognized-NE	52	140	46	203	441
+NE System	112 (31%)	258 (24%)	85 (78%)	215 (74%)	670 (54%)

Note: The percentages within parentheses show the corresponding error reduction rates.

It can be seen that most of the OOV NE errors in the CITYU and MSR corpora can be rescued once they are correctly recognized by the adopted NE recognizer. However, the error reduction rates for the PKU and AS corpora are much lower. After identifying those named entities that are recognized but not rescued, we find the following conclusions.

For the PKU corpus, the Chinese family name and the given name are separated according to the gold standard, and there are many more Chinese names than foreign names. Because the NE type given by the NE recognizer does not distinguish foreign names from Chinese names, once a foreign name (which should be considered only one word) with fewer than four characters is recognized by the NE recognizer, our proposed approach tends to split it into two words. For example, our model split the foreign name "库福尔" (Kufuor) into "库/福尔." This problem could have been alleviated if the provided NE-Type could have distinguished foreign names from Chinese names. Because the four cross-domain corpora provided by CIPS-SIGHAN-2010 adopts the same criteria as the PKU corpus, this reason also explains why NE information does not help in cross-domain tests.

In contrast, for the AS training corpus, no "dot" is typically adopted to separate the given name and the surname within foreign names (e.g., "迈克尔乔丹" (Michael Jordan)), and the "dot" frequently behaves as a single character word in this corpus. However, the testing corpus does contain the "dot" within foreign names (i.e., "迈克尔. 乔丹"). Therefore, eventhough such a foreign name was recognized by the NE recognizer, it is still split into three words "迈克尔/. /乔丹." This style inconsistency problem is thus beyond the scope of our proposed model.

However, even for the CITYU and MSR corpora, their relative error reduction rates are still far from perfect, because there are many recognition errors for the NE recognizer. For example, the location name "驿马岭" (post-horse ridge) is recognized as "驿 [马岭]/LOC" in the MSR corpus. As a result, some new errors are correspondingly introduced, although they are correctly handled by the baseline system.

According to this discussion, we can conclude that our enhanced integrated model can effectively utilize the information provided by the given NE recognizer. It can also be inferred that our model performs better with a better NE recognizer.

6. HANDLING SUFFIX-DERIVED[20] OOV WORDS

In linguistics[21], a suffix is a morpheme that can be placed after a stem to form a new word. A suffix also cannot stand alone as a word. According to this definition, only a few characters can be considered suffixes, such as '者' (-er), '化' (-ize), and '率' (rate). However, the character '船' (ship) in the words "储油船" (oil storage ship) and

[20]Because prefixes share similar properties with suffixes, they are expected to behave similarly to suffixes; additionally, they are much rarer than suffixes. Therefore, they are not studied here because of the disappointing results obtained.

[21]http://zh.wikipedia.org/wiki/%E8%AF%8D%E7%BC%80.

"太空船" (space ship) can help recognize those OOV words, although it can also appear as an independent word in the phrase "在/船/上" (on the ship). We thus loosened the constraint that a suffix can not stand alone as a word in this article to cover more such characters. That is, if a character tends to locate at the end of various words, it is regarded as if it plays the role of a suffix in those words.

In Chinese, suffixes are very productive at forming new words. For example, the word "旅行者" (traveler) can be formed by concatenating a stem ("旅行", travel) and a suffix ("者", -er). Although current approaches are able to recognize many suffix-related OOVs, they still remain an important type of error (cf. Table IV). Researchers, especially linguists [Dong et al. 2010], thus seek to further improve OOV word performance by characterizing the word formation process [Li 2011]. Furthermore, prefix- and suffix-related features are proposed as useful for CWS in some previous works [Tseng et al. 2005; Zhang et al. 2006]; nonetheless, no direct evidence is provided because prefix/suffix features are only part of the adopted features.

However, it is difficult to recognize suffix-derived OOV words, because whether a character is a suffix greatly depends on its context. For example, the character '化' is a suffix in the word "初始化" (initialize). However, it becomes a prefix in the word "化纤" (chemical fiber). In addition, whether a character is a suffix varies with the different annotation standards adopted by various corpora. For example, the character "厂" (factory) is a suffix in words such as "服装厂" (clothing factory) in the PKU corpus provided by the SIGHAN 2005 Bakeoff [Emerson 2005]. However, it is considered a single-character word on similar occasions in the MSR corpus. Suffixes thus cannot simply be identified with some prespecified characters prepared by the linguist and should be learned from the given benchmark.

6.1. Obtaining Additional Information for Suffixes

Because of the difficulty in recognizing true suffixes, previous works [Tseng et al. 2005; Zhang et al. 2006] extract a suffix-like list beforehand from each corpus in a context-free manner. Specifically, Tseng et al. [2005] treat characters that frequently appear at the end of rare words as potential suffixes. In their approach, words with a number of training-set occurrences below a given threshold are selected first, and their ending characters are then sorted according to their occurrences in those rare words. Subsequently, the suffix-like list is formed with those high-frequency characters. Zhang et al. [2006] construct their list in a similar way but without pre-extracting the rare words.

To reduce the number of suffix errors resulting from these primitive extraction procedures, we propose obtaining and using the suffix list in a more prudent manner as follows.

— Having considered that a suffix is supposed to be combined with different stems to form new words, we propose suffix productivity as the criteria for extracting the suffix list, which is defined as the size of the set $\{w|w \in \text{IV}, [\text{w} + \text{c}] \in \text{IV}\}$, where w is a word in the training set, c is a specific character that requires determining whether it should be extracted as a suffix character, and IV denotes in-vocabulary words. The size of this set counts how many different IV words can be formed by concatenating the given suffix character to an IV word in the training set. Therefore, larger suffix productivity means that the given suffix character can be combined with more different stems to form new words and thus is more likely to be a suffix.
— According to our investigation, most suffix-derived words are composed of a multicharacter IV word and a suffix, such as "旅行者" (i.e., "旅行" + "者"). Therefore,

we set the suffix status for a given character to be true only when that character is in the suffix list and its previous character is the end of a multicharacter IV word. In this way, we are able to avoid many over-generalization errors (thus improving the precision rate for OOV with suffixes), and this method results in only a little decline in the recall rate.

However, there were still two drawbacks to adopting these proposed suffix-like list. (1) The associated context required to decide whether a character should be considered a suffix is either completely not taken into account (in previous approaches) or treated too coarsely (in our proposed approach), and (2) the probability value (a finer information) that a given character will act as a suffix is not utilized; only a hard-decision flag (within or outside the list) is assigned to each character.

To overcome these two drawbacks, we introduce the context-dependent tagging bias level, which reflects the likelihood that the next character would tend to be the beginning of a new word (or be a single-character word) based on the local context. This is motivated by the following observation: if a trailing character is biased towards 'S' or 'B,' then the current character will prefer to be tagged as 'S' or 'E;' on the contrary, if the trailing character is biased towards 'M' or 'E,' then the current character will prefer to be tagged as 'B' or 'M.'

Having considered that the surrounding context might have been unseen for the testing instances, we introduce four different tagging bias probabilities as follows.

— *Context-free tagging bias level* (qf_i). The quantized value of $P(t_{i+1} \in \{E, M\}|c_{i+1})$ that is estimated from the training corpus. In our experiments, we quantize the probability into five different intervals: [0.0–0.2], [0.2–0.4], [0.4–0.6], [0.6–0.8], and [0.8–1.0]; therefore, qf_i is a corresponding member of $\{-2, -1, 0, 1, 2\}$.
— *Left-context-dependent tagging bias level* (ql_i). Compared with qf_i, $P(t_{i+1} \in \{E, M\}|c_i^{i+1})$ is used instead of $P(t_{i+1} \in \{E, M\}|c_{i+1})$. The quantization procedure is the same.
— *Right-context-dependent tagging bias level* (qr_i). Compared with qf_i, $P(t_{i+1} \in \{E, M\}|c_{i+1}^{i+2})$ was used instead of $P(t_{i+1} \in \{E, M\}c_{i+1})$. The quantization procedure is the same.
— *Surrounding-context-dependent tagging bias level* (qs_i). Compared with qf_i, $P(t_{i+1} \in \{E, M\}|c_i^{i+2})$ is used instead of $P(t_{i+1} \in \{E, M\}|c_{i+1})$. The quantization procedure is the same.

6.2. Instantiating the Unified Framework

The additional information A_i can be instantiated as one of the three types of suffix information previously described (i.e., previous suffix-list, proposed suffix-list, and proposed tagging bias level). Furthermore, the corresponding tag-matching status (denoted as MS_i here) will be decided as follows.

— For the previous suffix-list feature, MS_i is a member of {Match, Violate, Neutral}. If c_{i+1} is in the suffix-list, when t_i is assigned with the position tag 'B' or 'M', MS_i is 'Match;' otherwise, MS_i is 'Violate.' If c_{i+1} is not in the suffix list, MS_i is always 'Neutral,' no matter what position tag is assigned to t_i.
— For the proposed suffix-list feature, MS_i is also a member of {Match, Violate, Neutral}. If C_{i+1} is in the suffix list and c_i is the end of a multicharacter IV word, when t_i is assigned the position tag 'M', MS_i is 'Match;' otherwise, MS_i is 'Violate.' If C_{i+1} is not in the suffix list or c_i is not the end of a multicharacter IV word, MS_i is always 'Neutral.'

— For the proposed tagging bias level feature, MS_i is a member of {Match[q_i], Violate[q_i], Neutral}, where q_i was a member of $\{qs_i, ql_i, Qr_i, qf_i\}$ and is selected according to whether the context c_i^{i+2} in the testing sentence is seen in the training corpus. Specifically, if c_i^{i+2} is seen in the training corpus, then q_i is qs_i; else if c_i^{i+1} is seen, then q_i is ql_i; else if c_{i+1}^{i+2} is seen, then q_i is qr_i; otherwise, q_i is qf_i. When $q_i > 0$ (i.e., c_{i+1} tends to be the beginning of a new word), if t_i is assigned 'S' or 'E', then MS_i is Match[q_i]; otherwise, MS_i is Violate[q_i]. On the contrary, when $q_i < 0$ (i.e., c_{i+1} tends not to be the beginning of a new word), if t_i is 'B' or 'M,' then MS_i is Match[q_i], otherwise, MS_i is Violate[q_i]. For example, if $q_i = 2$ and $t_i = B$, then MS_i is 'Match[2].' On the contrary, if $q_i = -2$ and $t_i =$ B, then MS_i is 'Violate[−2].' Additionally, we had four different $P_{q[i]}(MS_i|c_{i-2}^i)$ (associated with $\{qs, ql, qr, qf\}$, respectively), and $q[i]$ indicates which of them should be adopted at c_i. Subsequently, according to the context of each testing instance, a specific $P_{q[i]}(MS_i|c_{i-2}^i)$ is adopted.

It is reasonable to expect that the character-tag trigram factor is more reliable when c_{i-1}^i is seen in the training corpus. Therefore, the scoring function for the suffix-list feature is

$$\begin{aligned} Score(t_i) \;=\; & (1-\alpha_k) \times \log P\big([u,t]_i \,\big|\, [u,t]_{i-2}^{i-1}\big) \\ & + \alpha_k \times \log P\big(MS_i \big| u_{i-2}^i\big); \quad 1 \le k \le 2, \end{aligned} \tag{10}$$

where α_k is selected according to whether c_{i-1}^i is seen.

For the tagging bias feature, the scoring function is

$$\begin{aligned} Score(t_i) \;=\; & (1-\alpha_{q,k}) \times \log P\big([u,t]_i \,\big|\, [u,t]_{i-2}^{i-1}\big) \\ & + \alpha_{q,k} \times \log P\big(MS_i \big| u_{i-2}^i\big); \quad 1 \le q \le 4, 1 \le k \le 2, \end{aligned} \tag{11}$$

where $\alpha_{q,k}$ is selected according to which tagging bias probability factor is used and whether c_{i-1}^i is seen. Therefore, we will have eight different $\alpha_{q,k}$ in this case.

For the discriminative approach, Tseng et al. [2005] propose an additional suffix-like list feature to utilize the suffix information.

— s_0. A binary feature indicating whether the current character of concern is on the list. In addition to this feature, Zhang et al. [2006] also utilizes some combinational features: $c_0s_1, c_0s_1, c_{-1}s_0, c_{-2}s_0$, where c denotes a character and s denotes the preceding suffix-like list feature.

In addition, we also tested the case of context-free tagging bias level (proposed in Section 6.1), under the discriminative framework, by adding the following template.

— qf. The context-free tagging bias level.

Please note that qs (also ql and qr) are not adopted because it is always qs in the training set (thus over-fitted). Therefore, only qf is adopted to make the training and testing conditions consistent.

6.3. Experiments

The segmentation results of using different generative models proposed in Section 6.2 for the SIGHAN-2005 corpora are shown in Table XIV. 'Baseline' in the table denotes the basic generative model corresponding to Equation (1), with preconverting numbers and foreign letters into their corresponding metacharacters; '+Suffix-Like List'

Table XIV. Segmentation Results for Proposed Generative Approaches in F-Scores

Approaches		PKU	AS	CITYU	MSR	Overall
Baseline		0.953	0.950	0.944	0.971	0.956
+Suffix-Like List	Tseng	0.952	0.950	0.945	0.971	0.956
	Zhang	0.952	0.950	0.945	0.971	0.956
	Proposed	0.953	0.951	0.946	0.971	0.956
+Tagging Bias Level		**0.954**	**0.952***	**0.952***	**0.972***	**0.958***

Table XV. Segmentation Results for Various Discriminative Approaches, in F-Scores

Approaches	PKU	AS	CITYU	MSR	Overall
Baseline	**0.947**	**0.953**	0.943	**0.960**	**0.952**
Tseng	**0.947**	0.952	0.942	0.959	0.951
Zhang	0.946	0.951	0.941	0.959	0.950
With *qf*	**0.947**	0.952	**0.945***	0.959	**0.952**

denotes the model that adopts the related suffix-like list features, corresponding to Equation (8); and each subrow to the right indicates the method used to extract the list. '+Tagging Bias Level' denotes the model that adopts tagging bias level related features, corresponding to Equation (9).

Table XIV shows that the improvement brought by the tagging bias level is statistically significant over the original model for three out of four corpora; however, the difference is small, except for the CITYU corpus. Moreover, for the suffix-like list approaches, the performance is only slightly improved when the suffix-list is extracted and used as we propose. To examine whether the quality of the suffix-list would affect performance, we manually remove those characters that should not have been considered suffixes in each list (e.g., characters such as "斯" and "尔" that always appear at the end of transliteration). However, the performances are nearly the same, even with the cleaned lists (not shown in the table). The reasons are identified and explained in Section 6.4.

Table XV shows the segmentation results for the various discriminative approaches. 'Baseline' in the table denotes the baseline discriminative model that adopts the features (a)–(c) described in Section 2.1; 'Tseng' denotes the model with additional feature (d) in Section 6.2, and 'Zhang' denotes the model with additional features (d) and (e). Finally, 'with *qf*' denotes the model with additional feature (f) instead of features (d) and (e). Please note that *qs* (also *ql* and *qr*) is not adopted (explained in Section 6.2).

The results in Table XV show that neither the suffix-like list related feature nor the context-free tagging bias level feature could provide any help for the discriminative approach. Similar to the generative approach, no significant benefit is brought about, even when the list is further cleaned by a human. This seems contradictory to the claims given in Tseng et al. [2005] and Zhang et al. [2006], and the reason is studied in the next section.

According to the experiment results, it seems that only the generative approach with the tagging bias level feature could slightly improve performance. To determine whether it truly reduced suffix-derived OOV errors, we collect the number of suffix-derived OOV errors in the baseline approach and the proposed generative approach with the tagging bias level feature, as shown in Table XVI. It can be observed that only a small number (less than 12%) of the suffix-derived OOV errors could actually be rescued. Thus, we are able to conclude that none of the approaches studied in this article effectively reduce suffix-derived OOV errors. We thus do not explore whether

Table XVI. Error Reduction Rates (within Parentheses) for OOV Suffix-Derived Words using the Proposed Generative Approach with Tagging Bias Level Feature

#err for OOV suffix-derived words	**PKU**	**AS**	**CITYU**	**MSR**	**Overall**
Baseline	77	420	123	9	629
+ tagging bias level	76 (1.3%)	398 (5.2%)	115 (6.5%)	8 (11%)	597 (5.1%)

Table XVII. The Matching Rates of Various Tagging Bias Factors in the Training Set

Corpus	*qs*	*ql*	*Qr*	*qf*
PKU	0.996	0.977	0.923	0.686
AS	0.993	0.970	0.899	0.662
CITYU	0.997	0.976	0.919	0.653
MSR	0.992	0.970	0.898	0.662

Table XVIII. Unseen Ratios for *qs*, *ql*, *qr*, and *qf* in the Testing Set

Corpus	*qs*	*Ql*	*qr*	*qf*
PKU	0.457	0.135	0.135	0.002
AS	0.374	0.083	0.082	0.004
CITYU	0.515	0.148	0.149	0.008
MSR	0.299	0.060	0.060	0.0003

suffix information could help in the integrated approach. Furthermore, cross-domain tests are also not conducted.

6.4. Problem Investigation and Discussion

As mentioned previously, whether a character can act as a suffix is highly context-dependent. Although context is considered in our proposed suffix-list and tagging bias approaches, the preference implied by the suffix list or the tagging bias level becomes unreliable when the context is unfamiliar. Table XVII shows the percentage that the preference of different tagging bias factors matched the real tag in the training set. It can be seen that the matching rate (or the influence power) is higher with a broader context. When no context is available (the last column; which is the suffix-list approach), the rate drops dramatically. As a result, many overgeneralized words are produced when *qf* is adopted. For example, two single-character words "该/局" (this bureau) are wrongly merged into a pseudo-OOV "该局." As another example, the first three characters in the sequence "冠军/奖碟" (championship award tray) are wrongly merged into a pseudo-OOV "冠军奖" (championship-award). Because the related context "奖碟" is never seen for the character '奖,' it is thus considered a suffix in this case (as it is indeed a suffix in many other cases, such as "医学奖" (medicine prize) and "一等奖" (first prize)).

However, according to the empirical study by Zhao et al. [2010], the OOV rate can be linearly reduced only with an exponential increasing of corpus size, roughly because of Zipf's law; n-gram is expected to also follow this pattern [Baroni 2009]. Therefore, the sparseness problem becomes more serious for the n-gram with a larger "n" (i.e., with a broader context) because its number of possible distinct types becomes much greater. As a consequence, there will be many more unseen bigrams than unseen unigrams in the testing set (of course, there will be even more unseen trigrams). Table XVIII shows the unseen ratios for qs, ql, qr, and qf, in the testing set. It can be observed that the unseen ratio for qs is much larger than that for qf. However, according to the

discussion in the previous section, the preference of tagging bias level is not reliable for qf. Therefore, the main reason behind the disappointing results in this section is the conflict between the reliability and the coverage of those suffix related features. That is, a more reliable suffix feature is less likely to be utilized in the testing set. As a result, no significant improvement could be achieved by using suffix-related features. It can be concluded that this problem cannot be solved with character n-gram features. This conclusion should be valuable for the relevant researchers in preventing them from wasting time on similar attempts.

7. JOINTLY HANDLING UNSEEN DICTIONARY WORDS AND NAMED ENTITIES

Based on the last three sections, the performance on unseen dictionary words and OOV named entities can be effectively improved with additional information, but suffix information can hardly help without side effects. Thus, we propose combining the approaches for unseen dictionary words and named entities and demonstrate how much improvement we can further accomplish with this combination.

7.1. Proposed Model

For the generative approach, the score function is instantiated as

$$\begin{aligned} Score(t_i) \;=\; & (1-\alpha_1-\alpha_2)\times \log P\big([u,t]_i \,\big|\, [u,t]_{i-2}^{i-1}\big) + \alpha_1 \times \log P\big(MD_i \big| u_{i-2}^{i}\big) \\ & + \alpha_2 \times \log P\big(MNE_i \big| u_{i-2}^{i}\big), \end{aligned}$$

where $P\big(MD_i \big| u_{i-2}^{i}\big)$ is the tag-matching factor for dictionary information and $P\big(MNE_i \big| u_{i-2}^{i}\big)$ is the tag-matching factor for named-entity information.

For the discriminative approach, we adopt the following feature templates within one model (i.e., no smoothing as adopted in Section 5.1) for simplicity:

(a) $C_n\ (n=-2,-1,0,1,2)$,
(b) $C_nC_{n+1}\ (n=-2,-1,0,1)$,
(c) $C_{-1}C_1$,
(d) MWL_i, dt_i,
(e) $C_n dt_n\ (n=i-1,i,i+1)$,
(f) $NEtag_i, NEtype_i, NElen_i$.

The generative and discriminative approaches are then combined via log-linear interpolation:

$$\begin{aligned} Score(t_i) \;=\; & \beta \times \big((1-\alpha_1-\alpha_2)\times \log P\big([u,t]_i \,\big|\, [u,t]_{i-2}^{i-1}\big) + \alpha_1 \times \log P(MD_i | u_{i-2}^{i}) \\ & + \alpha_2 \times \log P(MNE_i | u_{i-2}^{i})\big) + (1-\beta) \\ & \times \log P\big(t_i | u_{i-2}^{i+2}, MWL_i, dt_i, NEtag_i, NEtype_i, NElen_i\big). \end{aligned}$$

7.2. Experiments

In addition to the SIGHAN-2005 and CIPS-SIGHAN-2010 corpora, the Chinese Treebank [Xue et al. 2005] is also a popular dataset for evaluating word segmentation methods, so we adopt it for performance comparison. We use three versions of this corpus in our tests, that is, CTB5 (LDC2005T01), CTB6 (LDC2007T36), and CTB7 (LDC2010T07). Additionally, we follow Wang et al. [2011] in splitting them into the training set, the development set, and the testing set. The statistics of these three versions of the corpus are shown in Table XIX.

The results are shown in Tables XX to XXII. The state-of-the-art performances in open tests reported in the literature are also given for comparison. The results in the tables show that our approach achieves the best performances and significantly outperforms the best results reported in the literature on all corpora for both in- and cross-domain tests, which clearly demonstrates the effectiveness of our proposed

Table XIX. Corpora Statistics for the CTB Corpora

Corpus	Training Set		Dev. Set		Testing set		
	Tokens	Types	Tokens	Types	Tokens	Types	OOV Rate
CTB5	494K	37K	6.8K	1.8K	8K	1.8K	0.035
CTB6	641K	42K	60K	9.8K	82K	12K	0.056
CTB7	639K	40K	149K	17K	163K	18K	0.055

Table XX. In-Domain Open Test Results for the SIGHAN 05 Corpora, in F-Scores.

Systems	PKU	AS	CITYU	MSR	Overall
SIGHAN05-Best	0.969	0.956	0.962	0.972	0.965
Zhao10	0.967	0.960	0.966	0.980	0.968
Our baseline	0.957	0.957	0.954	0.972	0.961
+Dict.	0.977	0.962	0.969	0.981	0.972
+NE	0.958	0.958	0.957	0.974	0.962
+Both	**0.979***	**0.963***	**0.973***	**0.984***	**0.974**

Note: SIGHAN-05-Best denotes the best performance under each corpus in the SIGHAN 05 Bakeoff open tests (the best performances didn't come from the same system), and Zhao10 denotes the performance reported in Zhao et al. [2010]. An asterisk indicates that the difference between our combination approach (+Both) and the best previous approach is statistically significant.

Table XXI. Cross-Domain Open Test Results for the SIGHAN-10 Corpora in F-Scores

Systems	Literature	Computer	Medicine	Finance	Overall
SIGHAN-10 Best	0.955	0.950	0.938	0.960	0.951
Our baseline	0.932	0.941	0.918	0.956	0.937
+Dict.	0.965	0.965	0.959	0.973	0.966
+NE	0.932	0.941	0.918	0.956	0.937
+Both	**0.967***	**0.965***	**0.960***	**0.974***	**0.966**

Note: SIGHAN-Best denotes the best performance under each corpus in the SIGHAN 10 Bakeoff open tests (the best performances didn't came from the same system).

approach. In addition, as shown in the tables, better performance can always be achieved by combining dictionary and NE information, which shows that different resources can be independently utilized in their corresponding submodels under our proposed framework; we even achieved better-than-additive improvement in F-scores when the two resources are combined. Taking the performances on CTB5 as an example, using dictionary information improved F-score by 0.007 (from 0.977 to 0.984), using NE information improved the score by 0.002 (from 0.977 to 0.979), and using both, by 0.01 (from 0.977 to 0.987).

8. RELATED WORK

Approaches for CWS can be divided into two groups according to whether any additional information or resources (other than what could be extracted from the training set) is adopted. For the group that does not use any additional information (i.e., the close test), many approaches have been proposed, and they can be further divided into the following categories: only using sub-word-related features [Xue 2003; Ng and Low 2004; Peng et al. 2004; Tseng et al. 2005; Jiang et al. 2008; Xiong et al. 2009]; only using word-related features [Gao et al. 2003; Zhang et al. 2003]; and using both sub-word- and word-related features [Zhang and Clark 2007; Sun 2010]. All of these works that use sub-word-related features adopt the discriminative approach. However, Wang et al. [2009] propose a generative approach and show comparable performance; furthermore, they integrate the generative model with the discriminative model and

Table XXII. Results for the CTB Corpora in F-Scores

Systems	CTB 5	CTB 6	CTB 7	Overall
Sun11a	NA	0.957	NA	NA
Sun11b	0.982	NA	NA	NA
Qian12	0.980	NA	NA	NA
Jiang09	0.982	NA	NA	NA
Wang11	0.981	0.958	0.957	0.958
Hatori12	0.983	0.962	0.961	0.962
Our baseline	0.977	0.955	0.958	0.957
+Dict.	0.984	0.966	0.966	0.966
+NE	0.979	0.956	0.959	0.958
+Both	**0.987***	**0.969***	**0.969***	**0.969***

Note: Sun11a denotes Sun and Xu [2011]; Sun11b denotes Sun [2011]; Qian12 denotes Qian and Liu [2012]; Jiang09 denotes Jiang et al. [2009]; Wang11 denotes Wang et al. [2011]; and Hatori12 denotes Hatori et al. [2012].

show that it outperforms all of the close-set performances reported in the literature [Wang et al. 2012].

On the contrary, for the open test in the SIGHAN bakeoffs, in which any knowledge or resource can be used to improve performance (which matches the scenario of real applications), there are many fewer papers in this category, which are discussed in detail as follows. Ng and Low [2004] adopt character-type-related features to work with factoids. Wang et al. [2012] propose an alternative way to use character-type information by preconverting specific types of characters into their corresponding metacharacters. Zhao et al. [2006a] introduce dictionary-related features to use external dictionaries. They also propose enlarging the training set with additional training corpora annotated with different segmentation standards. Jiang et al. [2009] make use of additional training corpora with annotation adaptation. Zhao et al. [2010] utilize the external dictionary, various assistant segmenters, and the output of an assistant NE recognizer in a discriminative model. Li and Sun [2009] use punctuation marks as implicit annotation. Sun and Xu [2011] enhance word segmentation with unlabeled data. More recently, Jiang et al. [2013] propose utilizing the natural annotations in Web texts to enhance word segmentation.

In addition to these approaches, which aim to improve word segmentation with additional resources, there are also approaches that jointly manage word segmentation, POS tagging, and even parsing, such as Jiang et al. [2008], Zhang and Clark [2008, 2011], Kruengkrai et al. [2009], Wang et al. [2011], Li and Zhou [2012], Qian and Liu [2012], Hatori et al. [2012], Zhang et al. [2013], and Wang et al. [2013]. As reported in these papers, word segmentation can always benefit from additional information about POS and syntax. However, the task will require a greater computation load if the information from later phases is not required in the given application, such as information retrieval.

Our work differs from previous works in several aspects. First, we propose a unified framework to incorporate various additional information and resources for the generative model, whereas previous works mainly focus on adding more features to the discriminative models or only adopt an additional pre/post-process. Second, we study the major OOV types systematically, whereas previous works solely focus on one specific type or ignore the differences between different types. Third, we show that different resources can be independently utilized in their corresponding submodels under our proposed framework and that positive interaction can be achieved in F-scores when dictionaries and an NE recognizer are combined.

9. CONCLUSION

OOV words are the main error sources for Chinese word segmentation, and different types of OOV words behave differently in various corpora. To systematically address the OOV problem, this article first classifies various OOV words into different types, which reveals that unseen dictionary words, named entities, and suffix-derived words account for the majority of OOV words. A unified framework is then proposed within which different types of additional information can be utilized independently in their corresponding submodels. Under this framework, unseen dictionary words, named entities, and suffix-derived words are studied separately. Experiment results and further analysis show that unseen dictionary words and OOV named entities can be effectively improved with additional dictionaries and an off-the-shelf named-entity recognizer, but suffix-related errors can be only marginally improved. Finally, we jointly utilize dictionary words and recognized named entities to outperform all of the results reported in the literature on all testing corpora.

Our main contributions are threefold. First, we propose a unified framework to incorporate various types of additional resources. Under this framework, different types of information can be easily incorporated and independently utilized, and they can be combined in our experiments with no negative interference in F-score. Second, contrary to claims made in some papers, we show that suffix-derived words are difficult to manage with the currently adopted character n-gram features, the reason mainly being in the conflict between the reliability and the coverage rates of those character n-grams. Third, we give the distribution over different OOV types on real corpora and note which ones are more important to solve.

REFERENCES

Baroni, M. 2009. Distributions in text. In *Corpus Linguistics: An International Handbook*, A. Lüdeling and M. Kytö (Eds.). Mouton de Gruyter, Berlin.

Bilmes, J. A. and Kirchhoff, K. 2003. Factored language models and generalized parallel backoff. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL'03)*. 4–6.

Chen, S. F. and Goodman, J. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98. Harvard University Center for Research in Computing Technology.

Dong, Z., Dong, Q., and Hao, C. 2010. Word segmentation needs change—From a linguists view. In *Proceedings of the 1st CIPS-SIGHAN Joint Conference on Chinese Language Processing*. 1–7.

Emerson, T. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*. 123–133.

Feng, H., Chen, K., Deng, X., and Zheng, W. 2004. Accessor variety criteria for Chinese word extraction. *Comput. Linguistics 30*, 1, 75–93.

Gao, J., Li, M., Wu, A., and Huang, C.-N. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Comput. Linguistics 31*, 531–574.

Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. 2012. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. 1045–1053.

Huang, C. and Zhao, H. 2007. Chinese word segmentation: A decade review. *J. Chinese Inf. Process. 21*, 3, 8–20.

Jiampojamarn, S., Cherry, C., and Kondrak, G. 2010. Integrating joint n-gram features into a discriminative training framework. In *Proceedings of the NAACL*. 697–700.

Jiang, W., Huang, L., and Liu, Q. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging—A case study. 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*. 522–530.

Jiang, W., Huang, L., Liu, Q., and Lu, Y. 2008. A cascaded linear model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the ACL*. 897–904.

Jiang, W., Sun, M., Lv, Y., Yang, Y., and Liu, Q. 2013. Discriminative learning with natural annotations: Word segmentation as a case study. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. (Vol. 1, Long Papers). 761–769.

Jin, G. and Chen, X. 2008. The fourth international Chinese language processing bakeoff: Chinese word segmentation, named entity recognition and Chinese POS tagging. In *Proceedings of the 6th SIGHAN Workshop on Chinese Language Processing*. 69.

Kruengkrai, C., Uchimoto, K., Kazama, J., Wang, Y., Torisawa, K., and Isahara H. 2009. An error-driven word-character hybrid model for joint Chinese word segmentation and POS tagging. 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*. 513–521.

Li, X., Wang, K., Zong, C., and Su, K.-Y. 2012. Integrating surface and abstract features for robust cross-domain Chinese word segmentation. In *Proceedings of COLING*. 1653–1670.

Li, X., Zong, C., and Su, K.-Y. 2013. A study of the effectiveness of suffixes for Chinese word segmentation. In *Proceedings of the 27th Pacific Asia Conference on Language, Information and Computation*.

Li, Z. 2011. Parsing the internal structure of words: A new paradigm for Chinese word segmentation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. 1405–1414.

Li, Z. and Sun, M. 2009. Punctuation as implicit annotations for Chinese word segmentation. *Comput. Linguistics 35*, 4, 505–512.

Li, Z. and Zhou, G. 2012. Unified dependency parsing of Chinese morphological and syntactic structures. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1445–1454.

Ng, H. T. and Low, J. K. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based. In *Proceedings of the EMNLP*. 277–284.

Nadeau, D. and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*, 1, 3–26.

Och, F. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. 160–167.

Peng, F., Feng, F., and McCallum, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING*. 562–568.

Qian, X. and Liu, Y. 2012. Joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 501–511.

Stolcke, A. 2002. SRILM—An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. 311–318.

Sun, J., Gao, J., Zhang, L., Zhou, M., and Huang, C. 2002. Chinese named entity identification using class-based language model. In *Proceedings of the 19th International Conference on Computational Linguistics*. pp 1–7.

Sun, W. 2010. Word-based and character-based word segmentation models: Comparison and combination. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. 1211–1219.

Sun, W. 2011. A stacked sub-word model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. 1385–1394.

Sun, W. and Xu, J. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 970–979.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. 2005. A conditional random field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*. 168–171.

Wang, K., Zong, C., and Su, K.-Y. 2009. Which is more suitable for Chinese word segmentation, the generative model or the discriminative one? In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC'23)*. 827–834.

Wang, K., Zong, C., and Su, K.-Y. 2012. Integrating generative and discriminative character-based models for Chinese word segmentation. *ACM Trans. Asian Lang. Inf. Process.*

Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., and Torisawa, K. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. 309–317.

Wang, Z., Zong, C., and Xue, N. 2013. A lattice-based framework for joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (Vol. 2, Short Papers). 623–627.

Xiong, Y., Zhu, J., Huang, H., and Xu, H. 2009. Minimum tag error for discriminative training of conditional random fields. *Inf. Sci. 179*, 1–2, 169–179.

Xue, N., Xia, F., Chiou, F., and Palmer, M. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Nat. Lang. Eng. 11*, 2, 207–238.

Xue, N. and Shen, L. 2003. Chinese word segmentation as LMR tagging. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*. 176–179.

Zhang, H., Yu, H., Xiong, D., and Liu, Q. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*. 184–187.

Zhang, M., Zhang, Y., Che, W., and Liu, T. 2013. Chinese parsing exploiting characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 125–134.

Zhang, M., Zhang, Y., Che, W., and Liu, T. 2014. Type-supervised domain adaptation for joint segmentation and POS tagging. In *Proceedings of the 14th Conference of the European Chapter of the ACL*. 588–597.

Zhang, R., Kikui, G., and Sumita, E. 2006. Subword-based tagging for confidence-dependent Chinese word segmentation. In *Proceedings of the COLING/ACL*. 961–968.

Zhang, Y., Vogel, S., and Waibel, A. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system. In *Proceedings of the 4th International Conference on Language Resource and Evaluation (LREC)*. 2051–2054.

Zhang, Y. and Clark, S. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the ACL*. 840–847.

Zhang, Y. and Clark, S. 2008. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of the ACL/HLT*. 888–896.

Zhang, Y. and Clark, S. 2011. Syntactic processing using the generalized perceptron and beam search. *Comput. Linguistics 37*, 105–151.

Zhao, H., Huang, C., and Li, M. 2006a. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*. 162–165.

Zhao, H., Huang, C.-N., Li, M., and Lu, B.-L. 2006b. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *Proceedings of the PACLIC-20*. 87–94.

Zhao, H., Huang, C.-N., Li, M., and Lu, B.-L. 2010a. A unified character-based tagging framework for Chinese word segmentation. *ACM Trans. Asian Lang. Inf. Process. 9*, 2, 1–32.

Zhao, H. and Kit, C. 2008. 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*. 106–111.

Zhao, H., Song, Y., and Kit, C. 2010b. How large a corpus do we need: Statistical method versus rule-based method. In *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC'10)*.

Zhao, H. and Liu, Q. 2010. The CIPS-SIGHAN CLP 2010 Chinese word segmentation bakeoff. In *Proceedings of the CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP'10)*. 199–209.

Zipf, G. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

Received May 2014; revised July 2014; accepted October 2014