# Integrating Generative and Discriminative Character-Based Models for Chinese Word Segmentation

KUN WANG and CHENGQING ZONG, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
KEH-YIH SU, Behavior Design Corporation

Among statistical approaches to Chinese word segmentation, the *word-based n-gram* (*generative*) model and the *character-based tagging* (*discriminative*) model are two dominant approaches in the literature. The former gives excellent performance for the *in-vocabulary* (IV) words; however, it handles *out-of-vocabulary* (OOV) words poorly. On the other hand, though the latter is more robust for OOV words, it fails to deliver satisfactory performance for IV words. These two approaches behave differently due to the unit they use (word vs. character) and the model form they adopt (generative vs. discriminative). In general, character-based approaches are more robust than word-based ones, as the vocabulary of characters is a closed set; and discriminative models are more robust than generative ones, since they can flexibly include all kinds of available information, such as future context.

This article first proposes a character-based $n$-gram model to enhance the robustness of the generative approach. Then the proposed generative model is further integrated with the character-based discriminative model to take advantage of both approaches. Our experiments show that this integrated approach outperforms all the existing approaches reported in the literature. Afterwards, a complete and detailed error analysis is conducted. Since a significant portion of the critical errors is related to numerical/foreign strings, character-type information is then incorporated into the model to further improve its performance. Last, the proposed integrated approach is tested on cross-domain corpora, and a semi-supervised domain adaptation algorithm is proposed and shown to be effective in our experiments.

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, character-based approach, generative model, discriminative model, model integration, domain adaptation

**ACM Reference Format:**
Wang, K., Zong, C., and Su, K.-Y. 2012. Integrating generative and discriminative character-based models for Chinese word segmentation. ACM Trans. Asian Lang. Inform. Process. 11, 2, Article 7 (June 2012), 41 pages.
DOI = 10.1145/2184436.2184440 http://doi.acm.org/10.1145/2184436.2184440

7

## 1. INTRODUCTION

In English and other western languages, space delimiters are used to mark word boundaries. However, no such spaces are used between adjacent words in Chinese

The research work has been funded by the Natural Science Foundation of China under Grant No. 60975053 and 61003160 and supported by the External Cooperation Program of the Chinese Academy of Sciences.
Authors' addresses: K. Wang and C. Zong, No. 95, Zhongguancun East Road, Handian District, Beijing, 100190, China; email: kunwang@nlpr.ia.ac.cn and cqzong@nlpr.ia.ac.cn; K.-Y. Su, 2F, No. 5, Industry East Road IV, Science-Based Industrial Park, Hsinchu, Taiwan; email: kysu@bdc.com.tw.

DOI 10.1145/2184436.2184440 http://doi.acm.org/10.1145/2184436.2184440

(and some other Asian languages such as Japanese and Korean). *Word segmentation* (WS) is thus required to find the corresponding word sequence for a given Chinese character sequence. As words are the basic units for text analysis, WS plays an important role, and is the first step, in most Chinese *natural language processing* (NLP) applications such as machine translation, information retrieval, and question answering. Since WS is the first phase, its errors will be passed along to subsequent phases. Thus the accuracy of WS is crucial for Chinese NLP.

Although WS is the initial task for Chinese NLP, it would sometimes benefit from subsequent phases that have not been carried out yet. Moreover, some ambiguities can be resolved only by additional contextual information from beyond the sentence. For example, "研讨会很热烈" can be segmented into two different meaningful sequences: "研讨会(discussion forum) 很(very) 热烈(hot)" "or 研讨(discussion) 会(will) 很(very) 热烈(hot)". Even humans cannot disambiguate this segment without the additional context. Such problems are difficult to address and are beyond the scope of this article.

We can, however, address another major problem. In real applications, words are often encountered which have never been encountered before. Among these *out-of-vocabulary* (OOV) words, named entities, numerical expressions, new words, and abbreviations are four typical types. No dictionary and no corpus could possibly contain all of them; so handling them is an unavoidable problem for all WS systems.

In the literature, rule-based approaches [Palmer 1997; Yeh and Lee 1991] and statistics-based approaches [Asahara et al. 2005; Gao et al. 2003, 2005; Xue 2003; Zhang and Clark 2007; Zhang et al. 2006] have been proposed as two major categories of WS algorithms. Due to its robustness in handling OOV words and its capability to automatically acquire knowledge, the statistical approach has been widely adopted and has become the mainstream approach since 1990. We will thus focus here on statistical approaches for further study.

### 1.1. Classification of Statistical Approaches

According to the basic unit adopted for extracting features, statistical approaches can be classified as either word-based [Gao et al. 2003; Zhang and Clark 2007; Zhang et al. 2003] or character-based [Asahara et al. 2005; Jiang et al. 2008; Ng and Low 2004; Peng et al. 2004; Tseng et al. 2005; Xiong et al. 2009; Xue 2003]. The word-based approach of course treats the word as the basic unit, so the desired segmentation result is the best word sequence directly obtained from the search process. By contrast, the character-based approach treats the word segmentation task as a character tagging problem by labeling each character as the beginning, the middle, or the end of a word. The final segmentation result is thus indirectly generated from the tag sequence assigned to the sentence.

Besides classification by basic unit as above, statistical approaches can also be classified as either adopting a *generative model*[1] or a *discriminative model*. The generative model learns the joint probability of the given input and its associated label sequence, while the discriminative model learns the posterior probability directly. Comparison of these two basic models is a perennial and interesting topic [Liang and Jordan 2008; Ng and Jordan 2002; Raina et al. 2004; Toutanova 2006; Xue and Titterington 2008]. In general, for the sequence labeling problem, the generative model closely couples each input and its associated label as a joint event, and thus cannot use the succeeding input to decide the current label in the sequence labeling problem. By contrast, the

[1]According to Wikipedia (http://en.wikipedia.org/wiki/Generative_model), "Generative models contrast with discriminative models, in which a generative model is a full probability model of all variables, whereas a discriminative model provides a model only of the target variable(s) conditional on the observed variables."

discriminative model does not associate each input with its label as a joint event and thus it is capable of using the succeeding input to decide the current label. In recent years, the discriminative model has become the dominant solution for NLP problems due to its flexibility in incorporating features with dependencies between them and directly optimizing classification accuracy [Toutanova 2006]. However, the performance advantage for the discriminative model can be very slight [Johnson 2001]; and the generative model can achieve very similar or even better performance than the corresponding discriminative model if a suitable structure can be adopted that avoids certain unrealistic independence assumptions [Toutanova 2006].

The above two dimensions of classification are orthogonal to each other, and thus can be freely combined. However, in the literature that we have checked, all the character-based tagging approaches adopt the discriminative model, and almost all the word-based approaches adopt the generative model. The exception is Zhang and Clark [2007][2], a word-based approach which adopts the averaged perceptron [Collins 2002] for training. For clarity, the time-honored word-based $n$-gram model will be called the word-based generative approach hereafter, and the model of Zhang and Clark [2007] will be called the word-based discriminative approach. The well-known character-based tagging model will be called the character-based discriminative approach. In addition, the word "model" will be freely exchanged with the word "approach" if there is no confusion.

Since the word-based generative model and the character-based discriminative model are the best-known ones in the generative and discriminative families respectively, and many different approaches have been built by extending them, they will be regarded as our baseline systems for performance comparison and will be briefly introduced now.

### 1.2. Word-Based Generative Model

The word-based $n$-gram generative model can be formulated below.

$$WSeq^* = \arg\max_{WSeq} P(WSeq|c_1^n) \tag{1}$$

where $WSeq \equiv w_1^m = [w_1, w_2, \ldots w_m]$ indicates a specific word sequence with $m$ words, and $c_1^n$ denotes a given sentence with $n$ characters. The classical word-trigram model $P(w_i|w_{i-2}, w_{i-1})$ is then derived as follows.

$$P(w_1^m|c_1^n) = P(c_1^n|w_1^m) \times P(w_1^m)/P(c_1^n) \tag{2}$$

Since $P(c_1^n|w_1^m) = 1$ and $P(c_1^n)$ is the same for various $WSeq$ candidates, only $P(w_1^m)$ should be considered. It can be further simplified with the second order Markov Chain assumption shown below.

$$P(w_1^m) = \prod_{i=1}^{m} P(w_i|w_1^{i-1}) \approx \prod_{i=1}^{m} P(w_i|w_{i-2}^{i-1}) \tag{3}$$

In Equation (3), the dependency between two adjacent characters within a word is implicitly handled by regarding the characters as a joint event (i.e., characters within a word are treated together as a unit). This model works well when there is no OOV word (i.e., only *in-vocabulary* (IV) words appear in the testing-set). However, this condition cannot be met in real applications. For example, named entities and numerical expressions are two kinds of OOV words which are often encountered. Since the

[2]Sun [2010] and Zhang and Clark [2011] also adopted the word-based discriminative model, but they are just slight variations of Zhang and Clark [2007].

Table I. The Tag-Set for Word Segmentation Adopted in this Article

| Tag | Meaning of the tag in a word |
|---|---|
| **B** | Beginning of a word that has more than one character |
| **M** | Middle of a word that has more than two characters |
| **E** | Ending of a word that has more than one character |
| **S** | Single-character word |

associated candidates of multi-character OOV words cannot be generated during the searching process without OOV pre-detection, it is impossible to identify them in this word-based approach. Most OOV words will thus be segmented into their corresponding sequences of uni-character-words. High *recall of IV words* (abbreviated as $R_{IV}$) and low *recall of OOV words* (abbreviated as $R_{OOV}$) are thus obtained (see Table VI). In other words, OOV words are problematic for word-based models. Meanwhile, the overall precision rate may also be low, as OOV words are segmented into more relatively short IV words.

Since the word-based approach has problems handling OOV words, an additional procedure incorporating other knowledge (not covered by the given corpus) is usually required [Gao et al. 2003; Zhang et al. 2003]. Although the performance of word-based models can be greatly enhanced with additional OOV detection and named entity recognition modules, the resulting system is rather complicated in comparison with the character-based discriminative approach described below. Thus the character-based discriminative model has become the dominant approach since it was proposed by Xue and Shen [2003].

### 1.3. Character-Based Discriminative Model

The character-based discriminative model [Xue and Shen 2003] treats segmentation as a tagging problem, which assigns a corresponding tag to each Chinese character. The model is formulated as follows.

$$P(t_1^n \,|c_1^n) = \prod_{k=1}^{n} P(t_k \,|t_1^{k-1}, c_1^n) \approx \prod_{k=1}^{n} P(t_k \,|t_{k-1}, c_{k-2}^{k+2}) \tag{4}$$

where $t_k$ is a member of {**B**egin, **M**iddle, **E**nd, **S**ingle} (which are separately abbreviated as $B$, $M$, $E$, and $S$, and defined in Table I) to indicate the corresponding position of the character $c_k$ in its associated word. For example, the word "北京市 (Beijing City)" will be assigned with the corresponding tags as: "北/B(North) 京/M(Capital) 市/E(City)".

In this work, the feature templates used in the character-based discriminative model are those of Ng and Low [2004], which have been widely adopted and reported in many papers. However, we exclude the features forbidden by the closed test regulations of the second SIGHAN Bakeoff. For example, the feature template $Pu(C_0)$, which indicates whether the current character is a punctuation or not, is not allowed. The adopted feature templates are listed below:

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

For example, when we consider the third character "奥" in the character sequence "北京奥运会", template (a) results in these features: $C_{-2}$=北, $C_{-1}$=京, $C_0$=奥, $C_1$=运,

$C_2$=会. Template (b) generates these features: $C_{-2}C_{-1}$=北京, $C_{-1}C_0$=京奥, $C_0C_1$=奥运, $C_1C_2$=运会. Finally, template (c) generates the feature $C_{-1}C_1$=京运.

In fact, in the literature, various tag-sets have been proposed which include 2, 3, 4, and even 6 tags. It was reported that the 6-tag set is much better than both the 2-tag set and the 3-tag set, but its superiority over the 4-tag set is not obvious [Zhao et al. 2006; Zhao et al. 2010]. According to our experiments on the second SIGHAN Bakeoff, the overall performances of the 4-tag set and the 6-tag set with the same features are 0.9467 (see Table VI) and 0.9472, respectively. Since this performance difference is statistically insignificant, the 4-tag set has been adopted by most previous studies. In order to fairly compare our approach with those in the literature, the 4-tag set will be adopted in this work as well.

Compared with the word-based generative model, this approach can better tolerate OOV words. Since the vocabulary size of the possible character-tag-pairs is limited, there are almost no OOV character-tag-pairs under this approach, and each multi-character OOV word can be converted into its corresponding sequence of character-tag-pairs. It is thus possible to correctly identify those OOV words. Therefore, this approach is robust with respect to OOV words and can yield a high $R_{OOV}$. On the other hand, though the dependencies between adjacent tags (labels) can be addressed in the character-based discriminative model, the dependency between adjacent characters within words cannot be directly modeled under this framework.

The dependency between adjacent characters within a word, which is implicitly handled in Equation (3), makes the word-based approach yield significantly higher $R_{IV.}$ To study how this improvement comes about, we calculated $\log P(c_i|c_{i-1})$ for various character-bigrams collected from all the training corpora provided by the second SIGHAN Bakeoff [Emerson 2005]. Figure 1 gives the distributions of $\log P(c_i|c_{i-1})$ for the class of character-bigrams within words (shown by black bars) and the class of character-bigrams between words (shown by white bars), where the X-axis represents different intervals of $\log P(c_i|c_{i-1})$ and the Y-axis denotes the relative frequencies of events associated with various intervals. As indicated in this figure, $\log P(c_i|c_{i-1})$ for the class of character-bigrams within-words[3] tends to have higher value, which explains why those IV words are more likely to be selected, so that high $R_{IV}$ is obtained in the word-based approach. Hence, even the word-based unigram model gives a much higher $R_{IV}$ than the character-based discriminative model (see Table VI).

### 1.4. Overview

As mentioned above, the traditional word-based generative model gives excellent performance for IV words. However, it is incapable of handling the OOV words in the testing set. Thus, in this paper, we first propose a *character-based generative model* to replace the word-based *n*-gram with *the character-tag-pair-based n*-gram. As the vocabulary of characters is a closed set (as opposed to the open set of words), robustness with respect to OOV words is enhanced in this generative model. Compared with the character-based discriminative approaches in the second SIGHAN Bakeoff, this new generative model achieves competitive results. As a second proposal, since the generative model and the discriminative model complement each other in handling IV words and OOV words, we further suggest a joint model with log-linear interpolation to integrate them. This joint approach achieves a good balance between IV word identification and OOV word recognition. The experiments on closed tests

[3]The pair-of-adjacent-characters features (e.g., $C_nC_{n+1}$ shown above) adopted in the discriminative approach do not distinguish between (1) the case where the two characters belong to the same word, and (2) the case where they belong to two different words. In contrast, $\log P(c_i|c_{i-1})$ of the within-words class directly measures the adhesion between adjacent characters within words.

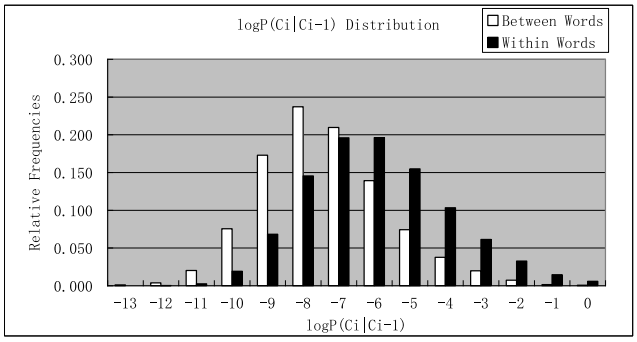


Fig. 1. The distributions of $\log P(c_i|c_{i-1})$.

in the second SIGHAN Bakeoff show that this joint model significantly outperforms the baseline models of both generative and discriminative approaches. Furthermore, statistical significance tests show that the joint model is significantly better than all state-of-the-art systems reported in the literature.

Afterward, a complete and detailed error analysis is conducted. According to the analysis, a significant portion of the critical errors is related to numerical or foreign strings. Information concerning character type, which distinguishes numerical or foreign or punctuation characters from Chinese characters, is thus proposed to further improve performance. Last, the proposed integrated approach is tested on cross-domain corpora, and a semi-supervised learning algorithm is proposed to carry out domain adaptation. Experiments on the CIPS-SIGHAN 2010 set show that this adaptation is effective in improving cross-domain performance, especially when there is a considerable difference between the two domains in the test.

The remainder of this article is organized as follows: Section 2 describes the proposed models in detail. The experiments conducted are reported in Section 3. Statistical significance tests for comparing various approaches are shown in Section 4. Section 5 provides error analysis and related discussion. Section 6 shows the effect of domain adaptation, and related work is described in Section 7. Finally, concluding remarks are made in Section 8.

The character-based generative model and the character-based joint model were originally introduced in Wang et al. [2009, 2010]. In this article, we provide more details concerning model analysis and experiment setting. In addition, complete error analysis is provided here, and new models exploiting character-type information are added and tested. Last, an effective semi-supervised algorithm for domain adaptation is proposed to improve cross-domain performance.

## 2. PROPOSED MODELS

To enhance the robustness of the generative approach in handling OOV words, a character-based model is required. We propose such a model in the following section. Afterwards, in Section 2.2, a character-based joint model is proposed to take full advantage of both the generative and the discriminative models.

### 2.1. Character-Based Generative Model

As explained in Section 1.2, the word-based approach is vulnerable to OOV words. To address the problem of OOV word recognition, we must adopt the character-based approach. However, we also need the generative model's ability to handle the dependency of character-bigrams within-words. Accordingly, we here propose a character-based

generative model to take advantage of both of the above-mentioned approaches by replacing $w_i$ with its corresponding [*character*, *tag*] sequence (abbreviated as $[c, t]$), where tag is the same as in the character-based discriminative model above. With this new representation, $P(w_1^m|c_1^n)$ can be re-derived based on the character-tag pair as follows:

$$P(w_1^m \,|c_1^n) \equiv P([c,t]_1^n \,|c_1^n) = P(c_1^n \,|[c,t]_1^n) \times P([c,t]_1^n) / P(c_1^n) \tag{5}$$

Following the derivation of Equation (2), only $P([c, t]_1^n)$ must be handled. It can be further simplified to:

$$P([c,t]_1^n) \approx \prod_{i=1}^{n} P([c,t]_i \,\Big|\, [c,t]_{i-k}^{i-1}) \tag{6}$$

In this work, the SRI Language Modeling Toolkit[4] (SRILM) [Stolcke 2002] is used to train various character-tag pair $n$-gram models using the modified Kneser-Ney smoothing method [Chen and Goodman 1998]. A beam search decoder with dynamic programming is applied to find the best result.

As shown in the last section, $P(c_i\,|c_{i-1})$ *within* words tends to have a higher value than between words. Therefore, for bi-character-words (and for other multi-character-words as well), when $[c_{i-1}, c_i]$ is an IV word, $P([c,t]_i \,|[c,t]_{i-1})$ for [$t_{i-1}$ = *B or M*; $t_i$ = *E or M*] is frequently higher than for [$t_{i-1}$ = *E or S*; $t_i$ = *B or S*], where the latter expression corresponds to the character-bigram between two adjacent words. In other words, IV words are more likely to be selected in the former case, and high $R_{IV}$ is thus expected. However, the dependency between $[c, t]_i$ and $[c, t]_{i-1}$ (represented by $P([c,t]_i \,|\, [c,t]_{i-1})$ in the generative model) cannot be modeled in the discriminative approach, as $c_i$ and $t_i$ must be jointly considered as an event.

Unlike the existing word-based generative model specified above, this new approach treats the character as a unit. It can thus correctly identify multi-character OOV words, as their corresponding candidates can now be generated during the searching process. In addition, the ability to handle the dependency between adjacent characters within words, which has been shown to be important for obtaining high $R_{IV}$ in the word-based approach, remains in the new model with its adopted generative form. Furthermore, as the basic unit in the new proposed model is the character, the model's vocabulary size is much smaller than that of the word-based approach. Thus the data sparseness problem will be greatly alleviated.

In summary, compared with the character-based discriminative approach, the proposed character-based generative model retains the capability to handle OOV words because it, too, regards the character as a unit. Also, since the generative form is adopted, the dependency between adjacent characters is now directly modeled; the proposed approach thus will prefer the IV word when it is encountered. By contrast, such dependency is not modeled in the character-based discriminative approach. Finally, compared with the word-based model, the character-based generative model can be more easily and naturally integrated with the character-based discriminative model within the searching process (shown in Equation (7)), since both models yield scores based on characters.

Nonetheless, besides those advantages mentioned above, a problem still remains for the new character-based generative model (and for all other generative models as well): The future context cannot be utilized in assigning the tag of the current character, as a character and its tag are jointly observed in the model. However, the future context can help to select the correct tag when the associated trigram has not been observed in the training set, as is the case for OOV words. By contrast, the character-based

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

Table II. The Corresponding Feature Weights for "露宿者" in the Sentence "[該][處][的][露宿者][只][有][數][人]". The Value of Each Entry Is the Corresponding Weight for the Feature when the Specified Tag Is Assigned Under the ME Framework. All Features Are Explained in Section 3.2.1.

| **宿** | Gold and Discriminative Model Tag: M; Generative Trigram Model Tag: E | | | |
|---|---|---|---|---|
| Tag/Probability | B/0.0333 | E/0.2236 | M/0.7401 | S/0.0030 |
| Feature \ Tag | B | E | M | S |
| $C_{-2}$ | −1.4375 | 1.3558 | 1.1071 | −1.0254 |
| $C_{-1}$ | 0.1572 | 0.1910 | −0.5527 | 0.2046 |
| $\boldsymbol{C_0}$ | 0.0800 | 0.7229 | −0.3174 | −0.4856 |
| $\boldsymbol{C_1}$ | 0.2282 | **−1.2696** | **2.9422** | −1.9008 |
| $C_2$ | 0.7709 | **−0.5970** | **0.4636** | −0.6375 |
| $C_{-2}C_{-1}$ | 0.2741 | 0.0049 | −0.1708 | 0.0000 |
| $C_2C_0$ | 0.0000 | 0.0921 | 0.0000 | 0.0000 |
| $C_0C_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $C_1C_2$ | −0.6718 | 0.8049 | −0.9700 | 0.8368 |
| $C_{-1}C_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

| **者** | Gold and Discriminative Model Tag: E; Generative Trigram Model Tag: S | | | |
|---|---|---|---|---|
| Tag/Probability | B/0.0009 | E/0.8137 | M/0.0012 | S/0.1842 |
| Feature \ Tag | B | E | M | S |
| $C_{-2}$ | 0.3586 | 0.3666 | −0.5657 | −0.1595 |
| $C_{-1}$ | 0.4175 | 0.0687 | −0.4330 | −0.0532 |
| $\boldsymbol{C_0}$ | 0.0000 | **4.5381** | 1.8847 | **2.7360** |
| $\boldsymbol{C_1}$ | −0.7207 | **2.8300** | 0.0000 | **1.8223** |
| $C_2$ | 0.4626 | −0.0846 | −0.0918 | −0.2862 |
| $C_{-2}C_{-1}$ | 0.0085 | 0.0000 | 0.0000 | −0.0024 |
| $C_{-1}C_0$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $C_0C_1$ | 0.0000 | −1.0279 | 0.0000 | 1.0494 |
| $C_1C_2$ | 0.0000 | 0.6127 | 0.0000 | 0.7113 |
| $C_{-1}C_1$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

discriminative model can take advantage of the future context in this case, as the character and its tag are not jointly observed in the model. An example will clarify this situation.

In the sentence "該(that) 處(place) 的(of) 露宿者(street sleeper) 只(only) 有(have/exist) 數(some) 人(person) (In that place, there are only some street sleepers)" in the CITYU corpus, "露/B宿/M者/E(street sleeper)" is an OOV word, while "露/B宿/E(sleep on the street)" is an IV word, where the associated tag of each character is given after the slash symbol. The character-based generative model wrongly splits "露宿者" ("sleep street person", i.e., street sleeper) into two words "露/B宿/E" (sleep street) and "者/S (person)", as the associated trigram for "露宿者" is not seen in the training set. However, the character-based discriminative model gives the correct result for "宿/M" (sleep) and this character's dominant features come from its future context "者" (person) and "只" (only), as shown in Table II. Similarly, the future context "只" (only) helps to assign the correct tag "E" to the character "者" (person).

Table II gives the corresponding feature weights (i.e., lambda values under the *Maximum Entropy* (ME) framework [Berger et al. 1996]) for "露 宿 者" in the character-based discriminative model. The table shows that in the Feature row of "$C_1$" below "宿", the lambda value associated with the correct tag "M" is "2.9422", which is the highest value in that row and far greater than that of the wrong tag "E" (i.e., "-1.2696") assigned by the character-based generative model. This indicates that feature "$C_1$" ("者") is the most important feature for correctly tagging "宿". This observation fits a linguistic interpretation perfectly. We can explain this point as follows. Since "者" acts as a Chinese suffix with probability 0.878 (3,202 out of 3,647) in the CITYU corpus examined in this article, once we foresee it as the next character, we can strongly suspect that the current character will be bound to it (i.e., the current character is very unlikely to be tagged with "E"). Similarly, in the Feature row of "$C_1$" below "者", the lambda value associated with the correct tag "E" is "2.8300", which is also much larger than those of the wrong tags "B" (-0.7207) and "M" (0.0000). Also, since the next character "只" acts as either a prefix or a single-character-word with probability 0.976 (2,336 out of 2,394) in CITYU corpus, once we foresee it as the next character, we can strongly suspect that the current character will not be bound with it (i.e., the current character is very unlikely to be tagged with "B" or "M").

### 2.2. Character-Based Joint Model

From the above discussion, it is clear that the proposed character-based generative model and the character-based discriminative model complement each other. Since the performance of IV word identification and the performance of OOV word recognition are both important for real applications, we need the strength of both models.

In general, combining two different models will yield better performance if the following conditions can be met: (1) The two models complement each other with respect to remaining errors. Of course, if both models make the same wrong decision for most errors, then they cannot help each other, and combining them is useless. (2) In cases of error, the model that makes the correct decision gives a strongly preferred or confident answer over its competitor, while the model that makes the wrong decision gives a much weaker preference, so that the strong model can override the weak one.

Table III shows the results for the first test condition comparing the generative model and the discriminative model. It displays the statistics of the remaining errors resulting from these two models (please refer to Section 3.2 for detailed settings). In the table, "D" denotes the discriminative model, and "G" denotes the generative model. Also, "G+" indicates that the generative model gives the correct decision for words in that column, and "G-" indicates that it gives the wrong decision. Similar interpretation also applies for "D+" and "D-" in relation to the discriminative model. Apparently, the errors under "G-D-" cannot be recovered by combining these two models, as both models prefer the incorrect answer. The last row of Table III (labeled "Overall") shows that "G-D-" cases make up only 31.2% of the overall errors (11,456 out of 36,729). After this portion is extracted, the row shows that "G+D-" occupies 71.8% under the IV category (12,027 out of 16,750); however, it occupies only 28.0% under the OOV category (2,384 out of 8,523). By contrast, "G-D+" occupies only 28.2% under the IV category (4,723 out of 16,750), but occupies 72.0% under the OOV category (6,139 out of 8,523). Therefore, we can conclude that these two models do complement each other to a considerable extent.

The situation for the second condition comparing the generative and discriminative models is shown in Table IV. It gives the strength of preference of the character-based generative model and the character-based discriminative model. This table simply follows Table III, replacing its entries with the corresponding average-gap, which is

Table III. Statistics for the Remaining Errors of the Character-Based Generative Model (G) and the Character-Based Discriminative Model (D). The "G+D-" Column Under "IV Errors" Denotes that the Generative Model Segments an IV Word Correctly but the Discriminative One Gives the Wrong Result. Other Abbreviations Are Interpreted as in the "G+D-" Case

| | IV Errors | | | OOV Errors | | |
|---|---|---|---|---|---|---|
| **Corpus** | **G+D-** | **G-D+** | **G-D-** | **G+D-** | **G-D+** | **G-D-** |
| **AS** | 2,240 (46.4%) | 943 (19.6%) | 1,640 (34.0%) | 419 (14.0%) | 1,442 (48.3%) | 1,124 (37.7%) |
| **CITYU** | 1,054 (56.8%) | 350 (18.9%) | 452 (24.3%) | 452 (28.1%) | 776 (48.3%) | 380 (23.6%) |
| **MSR** | 2,843 (64.1%) | 700 (15.8%) | 889 (20.1%) | 217 (14.9%) | 665 (45.7%) | 574 (39.4%) |
| **PKU (ucvt.)** | 3,099 (43.5%) | 1,707 (24.0%) | 2,318 (32.5%) | 978 (22.5%) | 2,256 (51.9%) | 1,115 (25.6%) |
| **PKU (cvt.)** | 2,791 (46.5%) | 1,023 (17.1%) | 2,182 (36.4%) | 318 (15.2%) | 1,000 (47.6%) | 782 (37.2%) |
| **Overall** | 12,027 (49.6%) | 4,723 (19.5%) | 7,481 (30.9%) | 2,384 (19.1%) | 6,139 (49.1%) | 3,975 (31.8%) |

Table IV. Statistics for the Degree of Preference from the Character-Based Generative Model (G) and the Character-Based Discriminative Model (D). This Table Simply Follows Table III, Replacing Its Entries with the Corresponding Average-Gap. (Please Refer to the Above Paragraph for Detailed Explanation)

| | IV Errors | | OOV Errors | |
|---|---|---|---|---|
| **Corpus** | **G+D-** | **G-D+** | **G+D-** | **G-D+** |
| **AS** | 0.73; −0.42 | −0.12; 0.68 | 0.41; −0.39 | −0.58; 0.76 |
| **CITYU** | 0.95; −0.30 | −0.21; 0.75 | 0.17; −0.59 | −0.63; 0.83 |
| **MSR** | 0.84; −0.40 | −0.24; 0.83 | 0.26; −0.41 | −0.64; 0.71 |
| **PKU (ucvt.)** | 0.93; −0.38 | −0.54; 0.95 | 0.20; −0.53 | −0.65; 0.96 |
| **PKU (cvt.)** | 0.79; −0.39 | −0.08; 0.68 | 0.54; −0.30 | −0.62; 0.87 |
| **Overall** | 0.84; −0.39 | −0.25; 0.79 | 0.30; −0.46 | −0.62; 0.84 |

the average score difference between the desired answer and its top competitor. In each cell, the first number denotes the average-gap given by the generative model, and the second number denotes that given by the discriminative model. The last row of this table (labeled "Overall") shows that, for IV Errors, the average-gap of "G" under "G+D-" is 0.84, which is larger than that of "D" (-0.39), and is also larger than that of "G" under "G-D+" (-0.25). In comparison, the average-gap of "D" under "G-D+" is 0.79, which is larger than that of "G" (-0.25), and is also larger than that of "D" under "G+D-" (-0.39). In other words, for the errors in "G+D-" column, the generative model gives relatively strong positive preference, and the discriminative model gives relatively weak negative preference. By comparison, for the errors in "G-D+" column, the generative model gives relatively weak negative preference, and the discriminative model gives relatively strong positive preference.

Similar phenomena can also be observed for the average-gap of "D" of the OOV errors under "G-D+" (marked in the last row of Table IV). However, no such phenomenon are observed for the average-gap of "G" of the OOV errors under "G+D-" (which is only 0.30 versus -0.46 of "D" under "G+D-", and versus -0.62 of "G" under "G-D+"). The above statistics shows that the generative model is relatively weak in handling OOV words. Therefore, combining these two models is unlikely to improve the situation in this portion of the data. Fortunately, it is only a small portion (19.1% of OOV errors, as shown in Table III). Therefore, to a considerable degree, these two models meet the second condition as well as the first. Since both conditions are met in our data set, we do expect that combining the two models would give better performance than employing either model individually.

In the literature, various ways have been proposed to combine the generative and discriminative models. Generative related terms are directly integrated into the discriminative model in Raina et al. [2004] and Fujino et al. [2005], but unfortunately

Table V. Corpus Statistics for the Second SIGHAN Bakeoff

| Corpus | Abbrev. | Encoding | Training Set (Words/Types) | Training Set (Words/Types) | OOV Rate |
|---|---|---|---|---|---|
| **Academia Sinica (Taipei)** | AS | Big5 | 5.45M/141K | 122K/19K | 0.046 |
| **City University of Hong Kong** | CITYU | Big5 | 1.46M/69K | 41K/9K | 0.074 |
| **Microsoft Research (Beijing)** | MSR | GB | 2.37M/88K | 107K/13K | 0.026 |
| **Peking University** | PKU(ucvt.) | GB | 1.1M/55K | 104K/13K | 0.058 |
| | PKU(cvt.) | GB | 1.1M/55K | 104K/13K | 0.035 |

this method cannot be applied to our case. By contrast, Jiampojamarn et al. [2010] integrates the generative joint $n$-gram model into the discriminative model as a set of binary features. However, among the various combining methods, the log-linear interpolation remains a simple but effective one [Bishop 2006]. Thus the following joint model is proposed. For the $k-$th character $c_k$, the score of the tag $t_k$ can be calculated via log-linear interpolation as below:

$$Score(t_k) = \alpha \times \log\left(P\left([c,t]_k \middle| [c,t]_{k-2}^{k-1}\right)\right) + (1-\alpha) \times \log\left(P\left(t_k \middle| t_{k-1}, c_{k-2}^{k+2}\right)\right) \quad (7)$$

In this joint model, $c_k$ and $t_k$ are as previously specified, and $\alpha$ ($0.0 \leq \alpha \leq 1.0$) is the weight for the generative model, obtained from a cross-validation set. $Score(t_k)$ will be directly used when searching for the best sequence. In this proposed joint model, the generative model and the discriminative model are integrated in a natural way[5], since both are character-based.

## 3. EXPERIMENTS

As different NLP tasks may require different segmentation criteria [Zhang and Clark 2007], there is no unified criterion for Chinese WS. Various corpora are thus frequently created with different criteria, and the WS performance for each corpus must be evaluated accordingly. For example, time expressions and organizations are treated as words in the MSR corpus, but are segmented as several smaller units in all other corpora provided in the second SIGHAN Bakeoff [Emerson 2005]. A system trained in one corpus thus might behave worse in another corpus if it is evaluated with different criteria; and almost no system can outperform all others across all different corpora. Therefore, to enable fairer comparison among various approaches, a set of corpora is usually required. For instance, the widely cited second SIGHAN Bakeoff WS contest provides four different standard corpora with their own criteria. These will be described in Section 3.1. Then various experiments that have been conducted on those corpora will be described in Section 3.2.

### 3.1. Data Sets Adopted

Since the corpora provided by the second SIGHAN Bakeoff [Emerson 2005] were widely adopted in various articles for comparing the performance of different approaches, they will be used to conduct various experiments in this article as well. These include the Academia Sinica Corpus (AS), the Hong Kong City University Corpus (CITYU), the Microsoft Research Corpus (MSR), and the Peking University Corpus (PKU). These corpora provide both Unicode coding and Big5/GB coding, and the latter format is adopted in our work. The statistics of these corpora are shown in Table V.

The PKU corpus is a bit different from the others. Arabic digits and English characters are encoded differently across its training set and testing set. In the training

[5]By contrast, it would be difficult to linearly combine the word-based $n$-gram model with the character-based tagging model.

set, Arabic digits and English characters are in full-width format, occupying two bytes. However, in the testing set, these characters are half-width characters occupying only one byte. Most researchers in the SIGHAN Bakeoff competition performed a conversion before segmentation [Xiong et al. 2009]. Since the coding inconsistency issue is unrelated to the WS problem that concerns us, this annoying disturbance ought to be eliminated to reflect the true performance. However, as the performance of both cases has been reported in previous studies, we will follow suit by conducting tests on both the *unconverted case* (denoted as ucvt.), which keeps the original half-width format in the testing set, and the *converted case* (denoted as cvt.), which pre-converts half-width format into full-width format in the testing set before evaluation. After the Arabic digits and English characters are converted, the OOV rate of the converted corpus is significantly lower than that of the unconverted corpus (as shown by the "PKU(cvt)" row in Table V).

### 3.2. Various Approaches Tested

To show the power of our proposed models, the word-based $n$-gram generative model and the character-based discriminative model are first tested as two baseline systems in Section 3.2.1, as they are the best known in the generative and discriminative families. Afterwards, the proposed character-based generative model and the character-based joint model are tested in Sections 3.2.2 and 3.2.3 respectively. Last, in Section 3.2.4, our experiments show that weighting various features initially in the ME approach makes significant improvements.

For performance evaluation, to fairly compare the proposed approaches with previous work, the "closed test" regulation, as stipulated in the second SIGHAN Bakeoff WS contest, will be respected in our experiments. This means that only the information found in the training data can be used: all other data or information is excluded, including knowledge of characters sets, punctuation characters, etc. In all tests to be conducted below, *Precision* (**P**), *Recall* (**R**), *F-score* (**F**), *Recall of OOV* ($\mathbf{R}_{OOV}$) and *Recall of IV* ($\mathbf{R}_{IV}$) are used to evaluate the segmentation results. The balanced $F$-score is calculated as: $F = 2PR / (P + R)$.

*3.2.1. Word-Based Generative Model and Character-Based Discriminative Model.* To evaluate the word-based generative approach, we first extract a word list from the training set as our vocabulary. The word-based generative model is also trained using the SRILM toolkit, with the same settings used in the character-based generative model (mentioned in Section 2.1).

The segmentation results of the word-based generative model are shown in Table VI (where the best $F$-score in each corpus is marked for visibility). As expected, it shows that all word-based $n$-gram models have high $\mathrm{R}_{IV}$ (even its unigram model outperforms the character-based discriminative approach with respect to $\mathrm{R}_{IV}$) and very low $\mathrm{R}_{OOV}$. Having further analyzed the testing-set errors generated by the trigram model, we find that, among the 16,781 error-patterns resulting from all the testing-sets, 11,546 (69%) errors are caused by segmenting an OOV into a sequence of IV words. This result clearly illustrates the model's disadvantage in handling OOV words and accounts for its low $\mathrm{R}_{OOV}$.

For the character-based discriminative model, the ME Package[6] provided by Zhang Le has been used to conduct experiments. (Training was carried out with Gaussian prior 1.0 and 300, 150 iterations for AS and other corpora respectively.) Table VI shows that the character-based discriminative model outperforms the word-trigram model on $F$-score and $\mathrm{R}_{OOV}$ metrics, but the latter obtains higher $\mathrm{R}_{IV}$. The low $\mathrm{R}_{IV}$ for

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

Table VI. Segmentation Results of Various Word-Based $n$-gram Models (Word-Unigram /Bigram/Trigram), the Character-Based Discriminative Model (Discriminative)

| Corpus | Method | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|
| **AS** | Word-unigram | 0.933 | 0.878 | 0.905 | 0.014 | 0.975 |
| | Word-bigram | 0.942 | 0.877 | 0.908 | 0.014 | 0.984 |
| | Word-trigram | 0.941 | 0.877 | 0.908 | 0.014 | 0.983 |
| | Discriminative | 0.955 | 0.946 | **0.951** | 0.707 | 0.967 |
| **CITYU** | Word-unigram | 0.924 | 0.851 | 0.886 | 0.162 | 0.984 |
| | Word-bigram | 0.928 | 0.851 | 0.888 | 0.162 | 0.990 |
| | Word-trigram | 0.929 | 0.852 | 0.889 | 0.162 | 0.990 |
| | Discriminative | 0.941 | 0.944 | **0.942** | 0.708 | 0.959 |
| **MSR** | Word-unigram | 0.965 | 0.925 | 0.945 | 0.025 | 0.990 |
| | Word-bigram | 0.969 | 0.926 | 0.947 | 0.025 | 0.995 |
| | Word-trigram | 0.969 | 0.926 | 0.947 | 0.025 | 0.995 |
| | Discriminative | 0.957 | 0.962 | **0.960** | 0.719 | 0.964 |
| **PKU (ucvt.)** | Word-unigram | 0.919 | 0.853 | 0.885 | 0.069 | 0.971 |
| | Word-bigram | 0.929 | 0.857 | 0.892 | 0.069 | 0.982 |
| | Word-trigram | 0.929 | 0.857 | 0.892 | 0.069 | 0.982 |
| | Discriminative | 0.922 | 0.941 | **0.932** | 0.620 | 0.941 |
| **PKU (cvt.)** | Word-unigram | 0.939 | 0.909 | 0.924 | 0.016 | 0.972 |
| | Word-bigram | 0.949 | 0.913 | 0.931 | 0.016 | 0.982 |
| | Word-trigram | 0.949 | 0.913 | 0.930 | 0.016 | 0.982 |
| | Discriminative | 0.940 | 0.951 | **0.946** | 0.685 | 0.949 |
| **Overall** | Word-unigram | 0.937 | 0.886 | 0.911 | 0.053 | 0.979 |
| | Word-bigram | 0.945 | 0.888 | 0.916 | 0.053 | 0.987 |
| | Word-trigram | 0.945 | 0.888 | 0.915 | 0.053 | 0.987 |
| | Discriminative | 0.944 | 0.950 | **0.947** | 0.680 | 0.956 |

the character-based discriminative model clearly shows the disadvantage of not using dependencies between adjacent characters within multi-character words. Among the 15,336 error-patterns from all of the testing-sets, we see that 9,291 of them (61%) occurred because an IV word-sequence was incorrectly segmented. This illustrates this model's weakness in handling IV words, which accounts for its low $R_{IV}$.

Since the word-based $n$-gram model tends to segment OOV words into uni-character-word sequences, we might think of further raising the performance by replacing the uni-character-word sequence generated via the word-based generative model by the corresponding result from the character-based discriminative model. Unfortunately, this simple merging method yields only results comparable with those of the character-based discriminative model, as not all uni-character-word sequences should be replaced and it is difficult to judge which ones. For instance, "两点之间 (between the two points)" is correctly segmented by the word-trigram model as "[两] (two) [点] (point) [之间] (between)", but the character-based discriminative approach gives the wrong result: "[两点] (an OOV word which would mean two o'clock or two points) [之间] (between)". The above strategy will thus convert the original correct result into an incorrect one, "[两点] [之间]".

*3.2.2. Character-Based Generative Model.* The experimental setting introduced in Section 2.1 is adopted here. Table VII gives the results of the proposed character-based generative model for various character $n$-gram-sizes ranging from $n$ = 2 to $n$ = 5, and

Table VII. Segmentation Results of the Word-Trigram Model (Word-Trigram), the Character-Based Discriminative Model (Discriminative) and Various Proposed Character-Based Generative *n*-gram Models (New (Bigram/Trigram/4-gram/5-gram))

| Corpus | Method | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|
| **AS** | Word-trigram | 0.941 | 0.877 | 0.908 | 0.014 | 0.983 |
| | Discriminative | 0.955 | 0.946 | **0.951** | 0.707 | 0.967 |
| | New (Bigram) | 0.954 | 0.934 | 0.944 | 0.509 | 0.975 |
| | New (Trigram) | 0.958 | 0.938 | 0.948 | 0.518 | 0.978 |
| | New (4-gram) | 0.958 | 0.938 | 0.948 | 0.518 | 0.978 |
| | New (5-gram) | 0.957 | 0.938 | 0.948 | 0.518 | 0.977 |
| **CITYU** | Word-trigram | 0.929 | 0.852 | 0.889 | 0.162 | 0.990 |
| | Discriminative | 0.941 | 0.944 | 0.942 | 0.708 | 0.959 |
| | New (Bigram) | 0.949 | 0.932 | 0.941 | 0.603 | 0.976 |
| | New (Trigram) | 0.951 | 0.937 | **0.944** | 0.609 | 0.978 |
| | New (4-gram) | 0.951 | 0.938 | 0.944 | 0.610 | 0.978 |
| | New (5-gram) | 0.951 | 0.938 | 0.944 | 0.610 | 0.978 |
| **MSR** | Word-trigram | 0.969 | 0.926 | 0.947 | 0.025 | 0.995 |
| | Discriminative | 0.957 | 0.962 | 0.960 | 0.719 | 0.964 |
| | New (Bigram) | 0.965 | 0.955 | 0.960 | 0.522 | 0.977 |
| | New (Trigram) | 0.974 | 0.967 | 0.970 | 0.561 | 0.985 |
| | New (4-gram) | 0.974 | 0.967 | 0.971 | 0.568 | 0.985 |
| | New (5-gram) | 0.974 | 0.967 | **0.971** | 0.568 | 0.985 |
| **PKU (ucvt.)** | Word-trigram | 0.929 | 0.857 | 0.892 | 0.069 | 0.982 |
| | Discriminative | 0.922 | 0.941 | **0.932** | 0.620 | 0.941 |
| | New (Bigram) | 0.924 | 0.923 | 0.924 | 0.400 | 0.956 |
| | New (Trigram) | 0.929 | 0.933 | 0.931 | 0.435 | 0.959 |
| | New (4-gram) | 0.927 | 0.934 | 0.931 | 0.432 | 0.957 |
| | New (5-gram) | 0.928 | 0.935 | 0.931 | 0.438 | 0.957 |
| **PKU (cvt.)** | Word-trigram | 0.949 | 0.913 | 0.930 | 0.016 | 0.982 |
| | Discriminative | 0.940 | 0.951 | 0.946 | 0.685 | 0.949 |
| | New (Bigram) | 0.949 | 0.946 | 0.948 | 0.494 | 0.965 |
| | New (Trigram) | 0.952 | 0.951 | 0.952 | 0.503 | 0.968 |
| | New (4-gram) | 0.952 | 0.952 | **0.952** | 0.511 | 0.967 |
| | New (5-gram) | 0.952 | 0.952 | 0.952 | 0.510 | 0.968 |
| **Overall** | Word-trigram | 0.945 | 0.888 | 0.915 | 0.053 | 0.987 |
| | Discriminative | 0.944 | 0.950 | 0.947 | 0.680 | 0.956 |
| | New (Bigram) | 0.949 | 0.939 | 0.944 | 0.491 | 0.969 |
| | New (Trigram) | 0.953 | 0.946 | **0.950** | 0.511 | 0.973 |
| | New (4-gram) | 0.953 | 0.947 | 0.950 | 0.512 | 0.973 |
| | New (5-gram) | 0.953 | 0.947 | 0.950 | 0.514 | 0.973 |

the best *F*-score for each corpus is highlighted for visibility. Table VII shows that the character-trigram model significantly outperforms the character-bigram model over all four corpora, but also shows that almost no improvement was observed as we increased the *n*-gram length. (Only the 4-gram result is a bit better than that of the MSR corpus, since it has the largest average-word-length.) This outcome strongly suggests that the training data is too sparse to support models with 4-grams and 5-grams.

From these results, it can be seen that the proposed character-trigram generative model significantly outperforms the word-trigram generative model and slightly outperforms the character-based discriminative model. Compared with the word-trigram approach, the proposed character-trigram model has dramatically raised the overall $R_{OOV}$ from 0.053 to 0.511 at the cost of slightly degrading the overall $R_{IV}$ from 0.987 to 0.973. This result clearly shows that the handicap of the word-based generative model in handling OOV words has been alleviated. In addition, compared with the character-based discriminative approach, the proposed character-trigram model is able to increase the overall $R_{IV}$ from 0.956 to 0.973, at the cost of degrading the overall $R_{OOV}$ from 0.680 to 0.511. Also, the overall precision rate of the proposed character-trigram model (0.946) is lower than that of the discriminative model (0.950), while the overall $F$-score of the proposed model (0.950) is higher than that of the discriminative one (0.947). This implies that the proposed model tends to segment OOV words into more words than the discriminative model does, while the higher recall also indicates that the proposed model results in more correct words.

Aside from performance, in regard to execution speed, the generative model is not significantly faster than the discriminative model in our experiments. However, the learning process of the generative approach is found to be much faster than that of the discriminative model, as the discriminative model needs hundreds of iterations for the adopted corpora. Taking the CITYU corpus as an example, the training time of the generative and discriminative models are 8.5s vs. 1333.4s (150 iterations) with the same environment. That is, the first model is approximately 157 times faster. Thus the proposed approach has a large additional advantage when massive training data must be processed.

When the remaining errors are examined, we find that the proposed model fails to handle some OOV words such as "露宿者" due to its inability to utilize future context when required, as illustrated in Section 2.1. However, the future context for the character-based generative model scanning from left to right is just its past context when scanning from right to left. We thus expected that such errors would be fixed if we let the model scan in both directions and then combined their results. However, we actually observed that these two scanning modes share more than 90% of their errors (and thus do not satisfy the first condition stipulated in Section 2.2). For example, in the CITYU corpus, the left-to-right scan generates 1,958 incorrect words and the right-to-left scan results in 1,947, while 1,795 errors are the same. Similar behavior is also observed for other corpora. So, unfortunately, the two scanning modes seem not to complement each other after all.

To analyze the problems, ten errors similar to "露 宿 者" were selected for examination. Only one of them was fixed via the abovementioned right-to-left scanning approach, and "露宿者" still was not segmented correctly. Having analyzed the scores from both scanning directions, we found that the original scores (in the left-to-right scan) when processing "者" and "宿" do improve if the model scans from right-to-left. However, the score when processing "露" deteriorates because the useful feature "者" (person) (a non-adjacent character for "露", seen first when scanning from right to left) still cannot be utilized when the prior context "宿者" as a whole is unseen, when the related probabilities are estimated via the Kneser-Ney smoothing technique. In other words, $P([c,t]_i \mid [c,t]_{i-2})$ is not used for estimating $P([c,t]_i \mid [c,t]_{i-2}^{i-1})$ in the Kneser-Ney smoothing technique. Although either the multiple-backoff [Bilmes and Kirchhoff 2003] or the ME estimator could overcome this drawback, the packages[7] that we have

[7]Factored Language Model: http://www.speech.sri.com/projects/srilm/. Maximum Entropy Modeling Toolkit for Python and C++: http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html.

Table VIII. Corpus Statistics for Development Sets and Testing Sets

| Corpus | Set | Words Number | OOV Number | OOV Rate |
|---|---|---|---|---|
| **AS** | Development set | 17,243 | 445 | 0.026 |
| | Testing Set | 122,610 | 5,308 / 5,311 | 0.043 / 0.043 |
| **MSR** | Development set | 17,324 | 355 | 0.020 |
| | Testing Set | 106,873 | 2,829 / 2,833 | 0.026 / 0.027 |
| **CITYU** | Development set | 12,075 | 537 | 0.044 |
| | Testing Set | 40,936 | 3,028 / 3,034 | 0.074 / 0.074 |
| **PKU** | Development set | 13,576 | 532 | 0.039 |
| | Testing Set (ucvt.) | 104,372 | 6,006 / 6,054 | 0.058 / 0.058 |
| | Testing Set (cvt.) | 104,372 | 3,611 / 3,661 | 0.035 / 0.035 |

adopted yield no obvious improvement or even worse overall performance for all our tri-gram models, as they are not designed to estimate $n$-gram probabilities for our problems. It would be interesting to see whether a multiple-backoff (or an ME) estimator could fix this problem, if either procedure were well designed for estimating $n$-gram probabilities. However, such experimentation is beyond the scope of this article.

*3.2.3. Character-Based Joint Model.* When the remaining errors were inspected, we found that the character-based generative model and the character-based discriminative model complement each other much more than the two scanning modes do. These two approaches share at most 38.7% of their errors in all corpora tested (31.2% overall, as shown in Table III). For example, in the CITYU corpus, the generative approach results in 1,958 incorrect words and the discriminative approach generates 2,338, while only 832 of them (38.7%) are the same. Similar statistics can also be observed in other corpora, per our original expectation.

For the character-based joint model, a development set is required to obtain the weight $\alpha$ for its associated generative model. Therefore, a small portion from each original training corpus is extracted as the development set and the remaining data is treated as the new training-set, which is then used to train two new parameter-sets for both of the associated generative and discriminative models. The number of sentences in each development set is proportional to the size of the corresponding training sets, with a ratio of about 0.5%. For the PKU training corpus, the last 300 sentences are extracted as the corresponding development set. Similarly, the last 400 sentences from the CITYU corpus, the last 600 sentences from the MSR corpus, and the last 2,000 sentences from the AS corpus are extracted as their development sets to obtain the various corresponding $\alpha$ values. The statistics for various new data sets are shown in Table VIII. In the rows of the testing sets, the number before "/" is the OOV number (or OOV rate) with respect to the original training sets, and the number after the slash is the OOV number (or OOV rate) with respect to the new training sets (after excluding the development sets). The variation of the OOV rate is barely noticeable.

The $F$-scores of the character-based joint model versus various $\alpha$ values are evaluated for four different development sets, as shown in Figure 2. The curves are flat near the top, which indicates that the character-based joint model is not sensitive to the $\alpha$ value selected. Judging by these curves, the most appropriate $\alpha$ values for the AS, CITYU, MSR, and PKU corpora are 0.30, 0.60, 0.60, and 0.60, respectively. The $\alpha$ values selected from the development sets are then adopted for conducting experiments on the testing sets.

Note that the AS corpus obtains the lowest $\alpha$ value (0.3, even less than 0.5; however, this curve is quite flat from 0.3 to 0.6). This is because the AS corpus is the only corpus

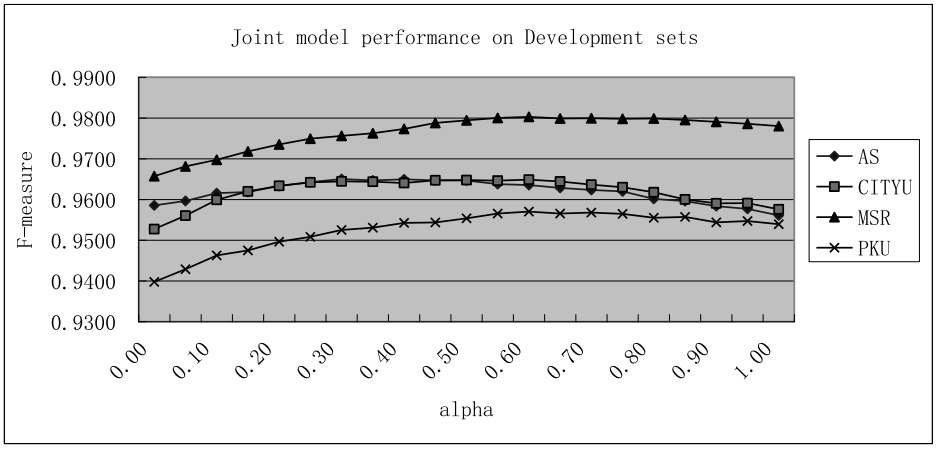


Fig. 2. Joint Model performance on development sets.

for which it's the generative model slightly lags behind the discriminative model in the development set[8] (with *F*-scores of 0.956 vs. 0.958).

Per Table IX, the character-based joint model significantly outperforms both the character-based generative model and the character-based discriminative model in *F*-score for all test corpora. Compared with the character-based generative approach, the joint model increases the overall $R_{OOV}$from 0.510 to 0.633, at the cost of slightly degrading the overall $R_{IV}$ from 0.973 to 0.971. These scores show that the joint model retains the advantage of the character-based generative model on IV words. Compared with the character-based discriminative model, the proposed joint model improves the overall $R_{IV}$ from 0.956 to 0.971, at the cost of degrading the overall $R_{OOV}$ from 0.680 to 0.633.

In addition, a *Recall-Upper-Bound*[9] column is added to Table III to show how much room is left to the proposed approach for further improvement. This score indicates a reasonable upper bound for the recall rate when these two models are integrated. Since combining the scores of the two models cannot recover from "G-D-" errors, and since "G+D-, OOV" errors are almost hopeless (because "D-" yields relatively strong preference for the wrong choice, while "G+" gives only weak preference for the correct one, as shown in Table IV), a reasonable upper-bound of recall rate[10] can be obtained by excluding all the "G-D-" errors and the "G+D-, OOV" errors in Table III. Per the Overall row, the proposed model fails to rescue 29.3%[11] of the errors in the classes "G-D+" and "G+D-, IV" (please see various error patterns at Section 5.1).

Although the proposed joint model has achieved the best performance, it gives the same weight to the character-based generative model for both IV words and OOV

[8]The *F*-scores of the generative and discriminative models in the development sets of various corpora are as follows. (The first number denotes that of the generative model, and the second number denotes that of the discriminative model.) 0.956 vs. 0.958 for AS, 0.958 vs. 0.953 for CITYU, 0.978 vs. 0.966 for MSR, and 0.954 vs. 0.940 for PKU.

[9]Thanks to an anonymous reviewer for suggesting this upper-bound.

[10]It is difficult to know how many words will be actually generated if the errors of "G-D+" and "G+D-, IV" are recovered, as a given error also affects its neighbors. Therefore, only the upper-bound of the recall-rate (not the precision-rate or *F*-score) is calculated.

[11]This number is obtained with the following formula: [(number of errors, in the classes of "G-D+" and "G+D-, IV", that the joint model fails to recover) / (total number of errors in the classes of "G-D+" and "G+D-, IV")]. Given those associated numbers of the last Overall row, we get 29.3% = [6,708 / (12,027 + 4,723 + 6,139)].

Table IX. Testing Set Performance of the Character-Based Generative Trigram Model (Generative), the Character-Based Discriminative Model (Discriminative), and the Character-Based Joint Model (Joint)

| Corpus | Recall-Upper-Bound | Models (Character-Based) | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|---|
| **AS** | 0.974 | Generative | 0.958 | 0.937 | 0.947 | 0.516 | 0.978 |
| | | Discriminative | 0.956 | 0.946 | 0.951 | 0.709 | 0.967 |
| | | Joint | 0.962 | 0.950 | **0.956** | 0.679 | 0.975 |
| **CITYU** | 0.969 | Generative | 0.951 | 0.937 | 0.944 | 0.611 | 0.978 |
| | | Discriminative | 0.940 | 0.944 | 0.942 | 0.709 | 0.959 |
| | | Joint | 0.957 | 0.951 | **0.954** | 0.691 | 0.979 |
| **MSR** | 0.984 | Generative | 0.973 | 0.966 | 0.970 | 0.560 | 0.985 |
| | | Discriminative | 0.957 | 0.963 | 0.960 | 0.720 | 0.964 |
| | | Joint | 0.974 | 0.971 | **0.972** | 0.659 | 0.983 |
| **PKU (ucvt.)** | 0.958 | Generative | 0.929 | 0.932 | 0.931 | 0.435 | 0.959 |
| | | Discriminative | 0.922 | 0.940 | 0.931 | 0.619 | 0.940 |
| | | Joint | 0.935 | 0.946 | **0.941** | 0.561 | 0.958 |
| **PKU (cvt.)** | 0.969 | Generative | 0.952 | 0.951 | 0.951 | 0.502 | 0.968 |
| | | Discriminative | 0.939 | 0.951 | 0.945 | 0.685 | 0.948 |
| | | Joint | 0.954 | 0.958 | **0.956** | 0.616 | 0.966 |
| **Overall** | 0.971 | Generative | 0.953 | 0.946 | 0.949 | 0.510 | 0.973 |
| | | Discriminative | 0.944 | 0.949 | 0.947 | 0.680 | 0.956 |
| | | Joint | 0.957 | 0.955 | **0.956** | 0.633 | 0.971 |

words. However, one might suspect that we should weight the character-based generative model more when IV words are encountered, and less when OOV words are seen. And yet, for character-based approaches, it is difficult to judge if the corresponding word is an IV word or an OOV word before we reach the last character of that word. Therefore, we tried weighting the character-based generative model differently according to whether the given character-bigram has been observed or not. Unfortunately, the results are disappointing and the improvements are slight. The reason for this phenomenon is that an observed character-bigram (even with the tag of $c_{k-1}$) cannot guarantee that its corresponding word is an IV word. For example, "露宿者 (street sleeper)" is an OOV word, but "露宿 (sleep on the street)" is an observed character-bigram in the training set. As another example, "[大] (big) [吊车] (crane)" are two IV words but "大吊" is an unseen character-bigram. It is thus difficult to distinguish IV words from OOV words until we reach the last character of a word. Therefore, adopting more weight parameters seems unnecessary (at least for the corpora that we have adopted).

Furthermore, as mentioned in Section 2.2, there are various ways to combine the generative and discriminative models. Besides the log-linear interpolation method, we have also tested different approaches that incorporate generative features into the discriminative training framework [Andrew 2006; Jiampojamarn et al. 2010]. However, whether we adopt binary joint $n$-gram features such as $C_{-1}t_{-1}C_0$ and $C_{-2}t_{-2}C_{-1}t_{-1}C_0$ (where $t_{-1}$ and $t_{-2}$ are the tags of character $C_{-1}$ and $C_{-2}$), or real-value generative $n$-gram scores, only negligible improvements are achieved for the original discriminative models, and the results are significantly worse than for the proposed joint model. This result is not surprising, as the feature $C_{-1}t_{-1}C_0$ cannot really reflect the dependency that we want to incorporate between $[c,t]_i$ and $[c,t]_{i-1}$. The result thus shows that log-linear interpolation is an effective way to integrate two models, simple though it is.

*3.2.4. Weight Various Features Differently.* Under the ME framework, each feature should be trained only once for a given observation and its associated weight will be

Table X. Performance of the Character-Based Generative Trigram Model, the Discriminative Model, and the Joint Model Weighting Various Features Differently (as Denoted by Generative, Discriminative-Plus and Joint-Plus, Respectively). The Data in Parenthesis (in the Discriminative-Plus and Joint-Plus Rows) Are the Performance of the Corresponding Original Models (i.e., Discriminative and Joint)

| Corpus | Model (Character-Based) | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|
| **AS** | Generative | 0.958 | 0.937 | 0.947 | 0.516 | 0.978 |
| | Discriminative-Plus | 0.960(0.956) | 0.948(0.946) | 0.954(0.951) | 0.680(0.709) | 0.973(0.967) |
| | Joint-Plus | 0.963(0.962) | 0.949(0.950) | **0.956**(0.956) | 0.652(0.679) | 0.977(0.975) |
| **CITYU** | Generative | 0.951 | 0.937 | 0.944 | 0.611 | 0.978 |
| | Discriminative Plus | 0.951(0.940) | 0.952(0.944) | 0.952(0.942) | 0.720(0.709) | 0.970(0.959) |
| | Joint-Plus | 0.959(0.957) | 0.952(0.951) | **0.956**(0.954) | 0.700(0.691) | 0.980(0.979) |
| **MSR** | Generative | 0.973 | 0.966 | 0.970 | 0.560 | 0.985 |
| | Discriminative-Plus | 0.965(0.957) | 0.967(0.963) | 0.966(0.960) | 0.675(0.720) | 0.973(0.964) |
| | Joint-Plus | 0.975(0.974) | 0.970(0.971) | **0.972**(0.972) | 0.632(0.659) | 0.984(0.983) |
| **PKU (ucvt.)** | Generative | 0.929 | 0.932 | 0.931 | 0.435 | 0.959 |
| | Discriminative-Plus | 0.934(0.922) | 0.949(0.940) | 0.941(0.931) | 0.649(0.619) | 0.951(0.940) |
| | Joint-Plus | 0.937(0.935) | 0.947(0.946) | **0.942**(0.941) | 0.556(0.561) | 0.960(0.958) |
| **PKU (cvt.)** | Generative | 0.952 | 0.951 | 0.951 | 0.502 | 0.968 |
| | Discriminative-Plus | 0.949(0.939) | 0.958(0.951) | 0.953(0.945) | 0.674(0.685) | 0.958(0.948) |
| | Joint-Plus | 0.955(0.954) | 0.958(0.958) | **0.957**(0.956) | 0.610(0.616) | 0.967(0.966) |
| **Overall** | Generative | 0.953 | 0.946 | 0.949 | 0.510 | 0.973 |
| | Discriminative-Plus | 0.952(0.944) | 0.955(0.949) | 0.953(0.947) | 0.676(0.680) | 0.965(0.956) |
| | Joint-Plus | 0.958(0.957) | 0.955(0.955) | **0.957**(0.956) | 0.621(0.633) | 0.973(0.971) |

learned from the training corpus automatically. However, when we repeat the work of Jiang et al. [2008], which reports achieving state-of-the-art performance in the data-sets that we have adopted, we find that some features (e.g., $C_0$) are inadvertently trained several times in their original implementation, which is implicitly generated from various feature templates adopted in the paper. Further, we observe that this study's improvements are mainly due to this implicit feature repetition, overlooked by the authors. For example, consider the feature $C_0$. (The meanings of the features are illustrated in Section 3.2.1.) This feature actually appears twice during training, which is implicitly generated from two different templates $C_n$ (with $n = 0$, generates $C_0$) and $[C_0C_n]$ (with $n = 0$, generates $[C_0C_0]$). The repetitive features also include $[C_{-1}C_0]$ and $[C_0C_1]$, which implicitly appear three times.

All the features adopted in Jiang et al. [2008] possess binary values. Thus, if a binary feature is repeated $n$ times, it should behave like a real-valued feature with value "$n$", at least in principle. With the above discovery in mind, we converted all binary-value features into their corresponding real-valued features and set the value of $C_0$ to 2.0; the value of $C_{-1}C_0$ and $C_0C_1$ to 3.0; and the values of all other features to 1.0. Then the original character-based discriminative model was re-trained under the ME framework. Logically, this new implementation is equivalent to simply starting from a different initial point when conducting ME training (i.e., with initial lambda values not necessarily equaling one). The training process was also done using Zhang Le's software with Gaussian prior 1.0 and 300, 150 iterations for AS and other corpora respectively. The new result is shown in Table X (in the *Discriminative-Plus* row). The original data is also shown in parentheses next to the new data for comparison. Table X shows that this new Discriminative-Plus implementation significantly outperforms the original one (the overall $F$-score is raised from 0.947 to 0.953) when both implementations adopt real-valued features for training. Therefore, it seems that, starting from different initial values, training will converge on different points in the parameter space for this case.

Table XI. *F*-Scores of Segmentation Results with Different ME Packages. The Slash Symbol "/" Separates the Results of the Character-Based Discriminative Model and the Character-Based Discriminative-Plus Model. **Zhang:** Zhang Le's ME Package; **Tsujii:** the ME Package from Tsujii Laboratory, University of Tokyo; **Lin:** Dekang Lin's ME Package

| | AS | CITYU | MSR | PKU(ucvt.) | PKU(cvt.) | Overall |
|---|---|---|---|---|---|---|
| **Zhang** | 0.951/0.954 | 0.942/0.952 | 0.960/0.966 | 0.932/0.941 | 0.946/0.953 | 0.947/0.953 |
| **Tsujii** | 0.952/0.953 | 0.942/0.949 | 0.960/0.963 | 0.927/0.936 | 0.946/0.951 | 0.946/0.951 |
| **Lin** | 0.947/0.948 | 0.942/0.949 | 0.960/0.965 | 0.932/0.939 | 0.947/0.951 | 0.946/0.950 |

To test whether this phenomenon is general for various ME training algorithms, we conduct similar experiments with two additional ME packages:

(1) MaxEnt Classifier[12] from Tsujii laboratory, University of Tokyo (denoted as Tsujii). Training was carried out with Gaussian prior 1.0, and the number of iterations is automatically decided.
(2) Dekang Lin's ME package[13] (denoted as Lin). Training was performed with Gaussian prior 1.0 and 500, 300 iterations for the AS and other corpora. Because this software only supports binary features, we directly duplicated $C_0$ twice, and $C_{-1}C_0$ and $C_0C_1$ thrice, for the character-based Discriminative-Plus model.

Similar improvements have been observed again with these two packages (as shown in Table XI). Perhaps giving more initial weight to closely related features would generate better results. ($C_0$ should be the most relevant feature for assigning tag $t_0$.) However, further analysis and detailed explanation of this problem would be beyond the scope of this article[14].

This new implementation is then further integrated with the character-based generative model. The resulting model will be called the *character-based Joint-Plus model*. Table X shows that this Joint-Plus implementation achieves better results than the Discriminative-Plus implementation, demonstrating that our Joint-Plus approach is an effective and robust method of Chinese word segmentation. However, compared with the original Joint model, the new Joint-Plus approach does not show much improvement, as shown in Table X, regardless of the significant improvement made by the Discriminative-Plus model. This is because the additional benefit generated by the Discriminative-Plus model has already been mostly covered by the generative model. (Among the 6,965 error words corrected by the Discriminative-Plus model, 6,292 of them (90%) are covered by the generative trigram model.)

## 4. STATISTICAL SIGNIFICANCE TEST

Although Tables IX and X showed that the proposed character-based joint model outperforms all other approaches mentioned above, we would like to know if the difference is statistically significant. Since the second SIGHAN Bakeoff provides only one testing set for each training corpus and creating a set of additional testing suites is very expensive, the well-known bootstrapping technique [Koehn 2004; Zhang et al. 2004] is adopted to conduct the significance tests. Following this approach, given an original testing-set $T_0$, $M$-1 additional testing-sets $T_1, \ldots T_{M-1}$ will be generated (each with the same size of $T_0$) by repeatedly resampling data from $T_0$. Thus we will obtain a total of $M$ testing-sets for each training corpus (with $M$ = 2,000 in our experiments).

[12]See http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/.
[13]See http://webdocs.cs.ualberta.ca/~lindek/downloads.htm.
[14]Dekang Lin (the author of the Lin package mentioned above) and another anonymous reviewer suspect that this problem is due to the adopted regularization step. However, the same phenomenon still appears even when we remove the regularization setting (i.e., without the Gaussian prior).

Table XII. Statistical Significance Test of $F$-Score Among Various Proposed Models

| Systems (Character-Based) | | AS | CITYU | MSR | PKU (ucvt.) | PKU (cvt.) | Overall |
|---|---|---|---|---|---|---|---|
| System A | System B | | | | | | |
| Generative | Discriminative | < | ∼ | > | ∼ | > | ∼ |
| Discriminative-Plus | Generative | > | > | < | > | > | > |
| Discriminative-Plus | Discriminative | > | > | > | > | > | > |
| Joint | Generative | > | > | > | > | > | > |
| Joint | Discriminative | > | > | > | > | > | > |
| Joint-Plus | Generative | > | > | > | > | > | > |
| Joint-Plus | Discriminative-Plus | > | > | > | ∼ | > | > |
| Joint-Plus | Joint | ∼ | > | ∼ | > | > | > |

### 4.1. Comparison of Proposed Approaches

We then follow Zhang et al. [2004] in measuring the confidence interval for the discrepancy between two models. For each performance measure (e.g., $F$-score) obtained from a specific testing-set $T_i$ ($i = 0, 1 \cdots, M-1$), assuming that its values are assigned by system A and system B as $a_i$ and $b_i$ respectively, then the discrepancy between system A and B for $T_i$ would be $\delta_i = a_i - b_i$. After $M$ discrepancy values are found, the 95% confidence interval for the discrepancy between system A and B is obtained by finding the minimum interval that could cover the middle 95% of the discrepancy values. If this confidence interval does not include the point of origin (value-zero), system A is considered to be significantly different from system B.

Table XII gives the results of significance tests among various models mentioned above. In this table, "$>$" means that system A is significantly better than system B; "$<$" means that system A is significantly worse than B; and "$\sim$" is used to mean that the two are not significantly different. As shown in Table XII, the proposed character-based generative trigram model achieves competitive results with the character-based discriminative model. The proposed Joint model is significantly better than the two baseline models on all corpora. Similarly, the proposed Joint-Plus model also significantly outperforms the character-based generative model and the character-based Discriminative-Plus model on all corpora except for the PKU(ucvt.) corpus. The comparison shows that the proposed Joint model and the Joint-Plus model outperform each of their component models. Further, the generative approach is inferior to the discriminative approach in the AS corpus, as the AS has a relatively high OOV rate. (The Discriminative model performs better for OOV words.) Finally, the Discriminative-Plus model is inferior to the generative one in the MSR corpus, as the MSR has the lowest OOV rate. (The generative model performs better for IV words.)

### 4.2. Comparison with Previous Works

The above comparison shows the superiority of the proposed model among the approaches that we implemented and tested. However, it would be interesting to know if the proposed Joint and Joint-Plus models also outperform state-of-the-art systems that have not been reimplemented in our lab. Since our tests have been performed on the corpora provided by the second SIGHAN Bakeoff[15] [Emerson 2005], the systems that performed best for $F$-score for at least one corpus in that contest have been first

[15]The performance comparison has not been performed on the systems that participated in the CIPS-SIGNAN Bakeoff-2010 (http://www.cipsc.org.cn/clp2010/task1_en.htm), as it is difficult to conduct fair comparisons among the systems. The systems adopt different rule-sets and additional information (e.g., the pinyin representation for characters) since there were no rigid rules for the closed track in this year.

selected for comparison. This category includes Asahara et al. [2005] (denoted as Asahara05) and Tseng et al. [2005][16] (Tseng05). The reported performance of these two systems is listed in Table XIII, and they are briefly summarized as follows. Asahara et al. [2005] achieves the best result on the AS corpus, using the character-based approach to first identify the OOV candidates and then integrate them into the system. Tseng et al. [2005] adds numerous linguistic features such as information on "word-prefixes" and "word-suffixes" (automatically extracted from the training-set) as well as morphological and character reduplication features. This system thus overcomes the drawbacks of character-based approaches and performs best on the remaining three corpora.

In addition, the systems reported to outperform the preceding two systems in the last few years have also been selected for comparison. This category includes Zhang et al. [2006] (denoted as Zhang06), Zhang and Clark [2007] (Z&C07), Jiang et al. [2008] (Jiang08), Sun [2010] (Sun10), and Zhang and Clark [2011] (Z&C11). Their performances are also reported in Table XIII, and their approaches are briefly summarized as follows. Zhang et al. [2006] use a sub-word tagging approach to utilize sub-word information and achieve the best performance on CITYU, MSR, and PKU. Zhang and Clark [2007] use perceptrons to generate word candidates with both word and character features and is the only word-based approach that adopts the discriminative form. Jiang et al. [2008][17] also adopt a perceptron based model for word segmentation based on Ng and Low [2004], with additional lexical-target features associated with the current character. (The feature templates (a) $\sim$ (c) in Section 3.2.1 are called non-lexical-target features in Jiang et al. [2008]. Lexical-target features are generated by adding C0 to each feature templates (a) $\sim$ (c)). Sun [2010] combines the outputs of the word-based discriminative model and the character-based discriminative model with a bagging approach. Last, Zhang and Clark [2011] uses a single discriminative model to adopt both word-based and character-based features.

Since the above-mentioned systems have not been re-implemented, the desired pair-wise samples cannot be obtained from the testing-set as in the last section. To overcome this problem, we associate each system that has been implemented with a 95% confidence interval by finding the minimum interval that could cover the middle 95% of its total $MF$-scores. Afterwards, the un-implemented systems are checked against each of the implemented systems. If (and only if) the $F$-score of system B (un-implemented) does not fall within the 95% confidence interval of system A (implemented), the two systems are considered significantly different statistically. In most cases, this sequential (or non-pair-wise) sampling test will have a wider confidence interval and is thus a stricter test [Zhang et al. 2004]. Table XIV gives a 95% confidence interval for our Joint model and Joint-Plus model for various corpora.

Table XV gives the results of significance tests for the un-implemented systems mentioned in this section. It shows that both our Joint model and Joint-Plus model outperform (or are comparable to) almost all state-of-the-art systems across all corpora, except Zhang and Clark [2007] and Zhang and Clark [2011] on the PKU(ucvt.) corpus. In that special case, Z&C07 outperforms the Joint-Plus model by 0.3% on $F$-score (0.4% for the Joint model). However, the Joint-Plus model exceeds Z&C07 on the

[16]We are not sure if Asahara05 and Tseng05 also perform a conversion before doing segmentation in PKU corpus. However, the comparison with these two systems was performed on PKU(cvt.) in Zhang06. Therefore, we simply follow them, comparing these two systems under the converted case.

[17]The data for Jiang08 in Table XIII are different from the data originally reported. In private communication with the authors, it was found that the script provided by the second SIGHAN Bakeoff for evaluating performance did not work correctly in their platform. After the problem was fixed, the re-evaluated actual performances reported here were less than those of their original version. Please see the announcement in Jiang's homepage (http://nlp.ict.ac.cn/$\sim$jiangwenbin/papers/error_correction.pdf).

Table XIII. Performance Report for Those Unimplemented Systems

| Corpus | Participants | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|
| **AS** | Asahara05 | 0.952 | 0.951 | 0.952 | 0.696 | 0.963 |
| | Tseng05 | 0.950 | 0.943 | 0.947 | 0.718 | 0.960 |
| | Zhang06 | 0.956 | 0.947 | 0.951 | 0.649 | 0.969 |
| | Z&C07 | N/A | N/A | 0.946 | N/A | N/A |
| | Jiang08 | 0.958 | 0.949 | 0.953 | 0.692 | 0.970 |
| | Sun10 | N/A | N/A | 0.952 | N/A | N/A |
| | Z&C11 | N/A | N/A | 0.954 | N/A | N/A |
| | Our Joint | 0.962 | 0.950 | **0.956** | 0.679 | 0.975 |
| | Our Joint-Plus | 0.963 | 0.949 | **0.956** | 0.652 | 0.977 |
| **CITYU** | Asahara05 | 0.937 | 0.946 | 0.941 | 0.736 | 0.953 |
| | Tseng05 | 0.941 | 0.946 | 0.943 | 0.698 | 0.961 |
| | Zhang06 | 0.952 | 0.949 | 0.951 | 0.741 | 0.969 |
| | Z&C07 | N/A | N/A | 0.951 | N/A | N/A |
| | Jiang08 | 0.946 | 0.950 | 0.948 | 0.695 | 0.966 |
| | Sun10 | N/A | N/A | **0.956** | N/A | N/A |
| | Z&C11 | N/A | N/A | 0.951 | N/A | N/A |
| | Our Joint | 0.957 | 0.951 | 0.954 | 0.691 | 0.979 |
| | Our Joint-Plus | 0.959 | 0.952 | **0.956** | 0.700 | 0.980 |
| **MSR** | Asahara05 | 0.952 | 0.964 | 0.958 | 0.718 | 0.958 |
| | Tseng05 | 0.962 | 0.966 | 0.964 | 0.717 | 0.968 |
| | Zhang06 | 0.972 | 0.969 | 0.971 | 0.712 | 0.976 |
| | Z&C07 | N/A | N/A | 0.972 | N/A | N/A |
| | Jiang08 | 0.964 | 0.967 | 0.966 | 0.686 | 0.972 |
| | Sun10 | N/A | N/A | 0.969 | N/A | N/A |
| | Z&C11 | N/A | N/A | **0.973** | N/A | N/A |
| | Our Joint | 0.974 | 0.971 | 0.972 | 0.659 | 0.983 |
| | Our Joint-Plus | 0.975 | 0.970 | 0.972 | 0.632 | 0.984 |
| **PKU (ucvt.)** | Asahara05 | N/A | N/A | N/A | N/A | N/A |
| | Tseng05 | N/A | N/A | N/A | N/A | N/A |
| | Zhang06 | N/A | N/A | N/A | N/A | N/A |
| | Z&C07 | N/A | N/A | **0.945** | N/A | N/A |
| | Jiang08 | 0.929 | 0.946 | 0.937 | 0.633 | 0.947 |
| | Sun10 | N/A | N/A | N/A | N/A | N/A |
| | Z&C11 | N/A | N/A | 0.944 | N/A | N/A |
| | Our Joint | 0.935 | 0.946 | 0.941 | 0.561 | 0.958 |
| | Our Joint-Plus | 0.937 | 0.947 | 0.942 | 0.556 | 0.960 |
| **PKU (cvt.)** | Asahara05 | 0.930 | 0.951 | 0.941 | 0.760 | 0.941 |
| | Tseng05 | 0.953 | 0.946 | 0.950 | 0.636 | 0.972 |
| | Zhang06 | 0.947 | 0.955 | 0.951 | 0.748 | 0.959 |
| | Z&C07 | N/A | N/A | N/A | N/A | N/A |
| | Jiang08 | N/A | N/A | N/A | N/A | N/A |
| | Sun10 | N/A | N/A | 0.952 | N/A | N/A |
| | Z&C11 | N/A | N/A | N/A | N/A | N/A |
| | Our Joint | 0.954 | 0.958 | 0.956 | 0.616 | 0.966 |
| | Our Joint-Plus | 0.955 | 0.958 | **0.957** | 0.610 | 0.967 |

Table XIV. 95% Confidence Interval of *F*-Score for the Proposed Joint Model and the Joint-Plus Model

| Corpus | Model | Mean | Interval |
|---|---|---|---|
| **AS** | Joint | 0.955 | [0.954, 0.957] |
| | Joint-Plus | 0.955 | [0.953[18], 0.957] |
| **CITYU** | Joint | 0.954 | [0.951, 0.957] |
| | Joint-Plus | 0.956 | [0.953, 0.958] |
| **MSR** | Joint | 0.972 | [0.971, 0.974] |
| | Joint-Plus | 0.972 | [0.971, 0.974] |
| **PKU (ucvt.)** | Joint | 0.941 | [0.938, 0.943] |
| | Joint-Plus | 0.942 | [0.939, 0.944] |
| **PKU (cvt.)** | Joint | 0.956 | [0.954, 0.958] |
| | Joint-Plus | 0.957 | [0.954, 0.959] |

Table XV. Statistical Significance Test for *F*-Scores for Unimplemented Systems

| Systems | | AS | CITYU | MSR | PKU (ucvt.) | PKU (cvt.) |
|---|---|---|---|---|---|---|
| **System A** | **System B** | | | | | |
| **Joint** | Asahara05 | > | > | > | N/A | > |
| | Tseng05 | > | > | > | N/A | > |
| | Zhang06 | > | ∼ | ∼ | N/A | > |
| | Z&C07 | > | > | ∼ | < | N/A |
| | Jiang08 | > | > | > | > | N/A |
| | Sun10 | > | ∼ | > | N/A | > |
| | Z&C11 | ∼ | > | ∼ | < | N/A |
| **Joint-Plus** | Asahara05 | > | > | > | N/A | > |
| | Tseng05 | > | > | > | N/A | > |
| | Zhang06 | > | > | ∼ | N/A | > |
| | Z&C07 | > | > | ∼ | < | N/A |
| | Jiang08 | ∼ | > | > | > | N/A |
| | Sun10 | > | ∼ | > | N/A | > |
| | Z&C11 | ∼ | > | ∼ | ∼ | N/A |

AS and CITYU corpora by 1.0% and 0.5%, respectively (1.0% and 0.3% for the Joint model). Similarly, Z&C11 outperforms the Joint model by 0.3% on *F*-score; while the Joint model exceeds Z&C11 by 0.5% on the CITYU corpus. Thus it is fair to say that both our Joint model and Joint-Plus model are superior to all state-of-the-art systems reported in the literature.

## 5. ERROR ANALYSIS AND DISCUSSION

### 5.1. Remaining Errors

We collect and analyze the errors on the CITYU and MSR corpora generated by the character-based joint model, because the OOV rates of these corpora are the highest and lowest among the five corpora. The statistics for remaining errors on CITYU and MSR are shown in Tables XVI and XVII respectively. The tag in parentheses following each character sequence (e.g., "就(IV)" in the first row under the *Example* column)

[18]The 95% confidence intervals for Joint and Joint-Plus in AS are very close. However, the rounding-off mechanism differentiates the two scores.

Table XVI. Statistics for Remaining Errors of the Joint Model on CITYU Corpus

| Word Class | Percentage | Type | Percentage | Example | |
|---|---|---|---|---|---|
| | | | | Gold | Output |
| **OOV** | 71.6% (802) | Critical | 73.8% (592) | [嘉義(IV)] [石弄溪(OOV)] | [嘉義石(NW)] [弄溪(NW)] |
| | | Not Critical | 26.2% (210) | [白玫瑰(OOV)] | [白(IV)] [玫瑰(IV)] |
| **IV** | 28.4% (318) | Critical | 55.3% (176) | [滑(IV)] [倒(IV)] | [滑倒(OOV)] |
| | | Inconsistency | 38.7% (123) | [就(IV)] [是(IV)] | [就是(IV)] |
| | | Not Critical | 6.0% (19) | [小女兒(IV)] | [小(IV)] [女兒(IV)] |

Table XVII. Statistics for Remaining Errors of the Joint Model on MSR Corpus

| Word Class | Percentage | Type | Percentage | Example | |
|---|---|---|---|---|---|
| | | | | Gold | Output |
| **OOV** | 53.0% (944) | Critical | 58.6% (553) | [决不(IV)] [好高骛远(OOV)] | [决(IV)] [不好(IV)] [高骛(NW)] [ 远(IV)] |
| | | Not Critical | 41.4% (391) | [钢针(OOV)] | [钢(IV)] [针(IV)] |
| **IV** | 47.0% (838) | Critical | 74.2% (622) | [自动化(IV)] [工厂(IV)] | [自动化工厂(OOV)] |
| | | Inconsistency | 13.1% (110) | [采访(IV)] [团(IV)] | [采访团(IV)] |
| | | Not Critical | 12.6% (106) | [大规模集成电路(IV)] | [大规模(IV)] [集成电路(IV)] |

indicates whether the sequence is an IV word or an OOV word (or not a word at all, as denoted by NW).

The errors are first classified as IV or OOV. Clearly, most of the remaining errors are related to OOV words. In the CITYU corpus, among 1,120 error-sequences, 802 (71.6%) are related to OOV. In MSR, this ratio is much less but still over 50%. (The gap arises because the OOV ratio of MSR is much less than that of CITYU.) Each category is then further classified into three main sub-categories: Critical, Inconsistency, and Not-Critical (to be defined later). In the headings of the following paragraphs, the first percentage in parentheses indicates the corresponding ratio (combining both IV and OOV) in the CITYU corpus, and the second percentage denotes the ratio in the MSR corpus.

(I) *Critical* (68.5% = 71.6% * 73.8% + 28.4% * 55.3%; 65.9% = 53.0% * 58.6% + 47.0% * 74.2%): Such errors yield information loss or meaning distortion. For example, the two uni-character words [“滑(IV)” (slip)] and [“倒(IV)” (fall down)] in Table XVI are wrongly combined into an OOV word [“滑倒(OOV) (slip)”], which would lose information (since no lexicon information can be found for an OOV word) and thus create additional problems for subsequent tasks such as pos-tagging and parsing. Similarly, the sequence [“自动化(IV)” (automation) and “工厂(IV)” (factory)] is grouped into one word [“自动化工厂(OOV)” (automated factory)] with coarser granularity. For comparison, [“嘉義(IV)” (ChiaYi)], a city name, and [“石弄溪(OOV)” (Shinong Creek)] are two location names, and are segmented into [“嘉義石(NW)” (a nonsense string)] and [“弄溪(NW)” (Nong Creek)], in which the meaning is distorted (and displays bracket-pair crossing). Similarly, [“决不(IV)” (in no way; never)] and [“好高骛远(OOV)” (reach for what is beyond one’s grasp)] are segmented into a sequence [“决(IV)” (definitely) “不好(IV)” (not good) “高骛(NW)” (a nonsense string) “远(IV)” (far)] (with bracket-pair crossing), in which the meaning is distorted as well.

(II) *Not Critical* (20.5% = 71.6% * 26.2% + 28.4% * 6.0%; 27.9% = 53.0% * 41.4% + 47.0% * 12.6%): For such errors, there is no bracket-pair crossing: instead,

different granularity levels have been adopted. Thus the result gives finer granularity without distorting the original meaning. For example, [“小女儿(IV)” (little daughter)] in Table XVI is segmented into [“小(IV)” (little)] and [“女儿(IV)” (daughter)]; also, [“大规模集成电路(IV)” in Table XVII (large scale integrated circuit)] is segmented into [“大规模(IV)” (large scale)] and [“集成电路(IV)” (integrated circuit)]; [“白玫瑰(OOV)” (white rose)] is segmented into [“白(IV)” (white)] and [“玫瑰(IV)” (rose)]; similarly, [“钢针(IV)” (steel needle)] is segmented into [“钢(IV)” (steel)] and [“针(IV)” (needle)]. None of these examples distort the original meaning.

(III) *Inconsistency* (11.0% = 28.4% * 38.7%; 6.2% = 13.1% * 47.0%): This kind of error only applies to IV words. It indicates that the benchmark and the Top-1 candidate are different, and that both of them have been observed in the training set. It may also indicate that the same words (in similar contexts) are segmented differently across the training set and the testing set. For example, both [“就是(IV)” (just like)] and the sequence [“就(IV)” (just), “是(IV)” (be)] are found in the training set in similar contexts in the CITYU corpus; by comparison, [“采访团(IV)” (interview team)] is a single word in the MSR training set but is segmented into two words [“采 访(IV)” (interview)] and [“团(IV)” (team)] in the benchmark.

As the “Inconsistency” category is unrelated to the proposed models, and the “Not Critical” category is usually not critical for the following processing phases, only the “IV-Critical” and the “OOV-Critical” sub-categories will be further analyzed here. For the “IV-Critical” cases, although the ambiguity problem has been reported in the literature as the main issue in segmenting the sequence of IV words [Zong 2008], it is not the major problem among “IV-Critical” errors in our experiments. Most of the remaining “IV-Critical” errors instead result from data sparseness. For example, [“滑(IV) (slip)”, “倒(IV) (fall down)”] are two successive uni-character IV words in the CITYU testing benchmark, and are incorrectly grouped into an OOV word [“滑倒” (slip)]; however, “滑倒” has not been observed as a continuous character sequence in the training set. Although “滑” and “倒” turn up as two uni-character words here, their associated probabilities are very low: the probability of “滑” with tag “S” is only 0.0370 (5 out of 135), while the probability with tag “B” is 0.4074 (55 out of 135); on the other hand, the probability of “倒” with tag “S” is merely 0.1675 (67 out of 400), while the probability with tag “E” is 0.2075 (83 out of 400). Therefore, these two characters tend to be grouped together, if they do not appear consecutively in the training set. Many similar cases can be found in the “IV-Critical” category. This problem is more serious in the MSR corpus than in CITYU, as there are more long words in MSR. This issue would be less significant if a larger corpus were adopted.

Table XVIII and Table XIX give the distributions of “OOV-Critical” errors on the CITYU and MSR corpora, respectively. The errors in this category are further classified into five sub-categories (as shown above) according to the causes; and the subcategories are indexed according to their associated ratios in the CITYU corpus. The errors in each sub-category will be classified again into various sub-classes if necessary, also according to the causes.

The distributions of various sub-categories for these two corpora are quite different. For example, the error type “(A) *Not Adopting Named Entity Preprocessing*” makes up the biggest portion in CITYU (30.5%); however, it is instead “(B) *Not Using Character-Type Information*” which occupies the largest portion in MSR (36.1%). Furthermore, even the distributions of various sub-types under those sub-categories differ considerably between these two corpora. For example, “Foreign name” is the largest sub-class (40%) under sub-category (A) in CITYU; however, it occupies only 6% in MSR. This difference arises because the CITYU corpus is collected from various regions (Beijing,

Table XVIII. Statistics of OOV-Critical Errors for the Joint-Model on the CITYU Corpus. In Each Class in the Examples Column, the First Row Shows the Gold Results and the Second Row Gives the Error Results Given by the Joint Model. (LOC: Location; ORG: Organization)

| Error Type | Percentage | Sub-Class | Examples |
|---|---|---|---|
| *(A) Not Adopting Named Entity Pre-processing* | 30.5% (181) | Foreign Name (40%); LOC (28%); ORG (23%); Chinese Name (9%) | [史丹利柯瓦斯基(OOV)] |
| | | | [史丹利(OOV)] [柯瓦斯基(OOV)] |
| | | | [嘉義(IV)] [石弄溪(OOV)] |
| | | | [嘉義石(NW)] [弄溪(NW)] |
| *(B) Not Using Character-Type Information* | 24.2% (143) | Punctuation (71%) | [光棍(IV)] [:(OOV)] |
| | | | [光棍:(NW)] |
| | | Numerical Expression (21%) | [二三六六一四一六(OOV)] |
| | | | [二三六六(OOV)] [一四一六(OOV)] |
| | | Foreign Alphabet (8%) | [英語(IV)] [valley(OOV)] |
| | | | [英語 valley(NW)] |
| *(C) Not Adopting Prefix/Suffix Information* | 15.7% (93) | Suffix (84%); Prefix (16%) | [*非*種子(OOV)] |
| | | | [*非*(IV)] [種子(IV)] |
| *(D) Idioms* | 14.4% (85) | | [巧奪天工(OOV)] |
| | | | [巧(IV)] [奪(IV)] [天工(NW)] |
| *(E) Others* | 15.2% (90) | | [出出入入(OOV)] |
| | | | [出(IV)] [出入入(NW)] |

Table XIX. Statistics of OOV-Critical Errors for the Joint Model on the MSR Corpus. In Each Class in the Examples Column, the First Row Shows the Gold Results and the Second Shows the Error Results Given by the Joint Model

| Error Type | Percentage | Sub-Class | Examples |
|---|---|---|---|
| *(B) Not Using Character-Type Information* | 36.1% (203) | Numerical Expression (85%) | [1 2 . 4 %(OOV)] |
| | | | [1 2 .(OOV)] [4 %(IV)] |
| | | Foreign Alphabet (13%) | [发明(IV)] [c c e d (OOV)] |
| | | | [发明 c c e d (NW)] |
| | | Punctuation (2%) | [第一(IV)] [′ (OOV)] |
| | | | [第一′(NW)] |
| *(A) Not Adopting Named Entity Pre-processing* | 25.4% (143) | ORG (55%); Chinese Name (28%); LOC (11%); Foreign Name (6%) | [北京和利时公司(OOV)] |
| | | | [北京(IV)] [和(IV)] [利时公司(OOV)] |
| | | | [任尧森(OOV)] |
| | | | [任(IV)] [尧森(OOV)] |
| *(D) Idioms* | 11.7% (66) | | [决不(IV)] [好高骛远(OOV)] |
| | | | [决(IV)] [不好(IV)] [高骛(NW)] [远(IV)] |
| *(C) Not Adopting Prefix/Suffix Information* | 10.3% (58) | Suffix (90%); Prefix (10%) | [沙漠*化*(OOV)] |
| | | | [沙漠(IV)] [*化*(IV)] |
| *(E) Others* | 16.4% (92) | | [双保(OOV)] |
| | | | [双(IV)] [保(IV)] |

Hong Kong, Shanghai, Taiwan, Singapore and Macao), so there are many more foreign names than in the MSR corpus, which mainly covers the China news. As another example, "Punctuation" makes up the largest portion (71%) under sub-category (B) in CITYU; however, it occupies only 2% in MSR. This large difference is mainly due to the corpus's coding inconsistency problem. In this corpus, punctuation symbols are encoded inconsistently in the training set and testing set; thus many punctuation errors occur. However, in the MSR corpus, such inconsistency appears only for the decimal point and not for other punctuation symbols.

The detailed description of each error type in the above tables is given as follows. Following the convention adopted above, the first percentage in parentheses in the

headings of the following paragraphs indicates its corresponding ratio in the CITYU corpus, and the second percentage denotes that in the MSR corpus.

(A) *Not Adopting Named Entity Pre-processing* (30.5%; 25.4%): Named entities (NE) frequently cannot be handled with character *n*-gram information only. Errors of this type can be further classified into four sub-classes as follows. Again, the first percentage in parentheses following the sub-class name indicates the corresponding ratio within this sub-category for the CITYU corpus, and the second number denotes that for the MSR.

(1) Foreign Name (40%; 6%). Most foreign names are rendered via syllable-by-syllable transliteration, which may be performed differently depending on the source. For example, in Table XVIII, the transliterated foreign name ["史丹利柯瓦斯基(OOV)"] is incorrectly segmented into two words as ["史丹利(OOV)"] and ["柯瓦斯基(OOV)"]. Since only a restricted set of characters is usually employed to render syllables in foreign names, the membership of this set might be helpful in providing useful information for correctly segmenting these names.
(2) LOC – location (28%; 11%). For example, in Table XVIII (and also in Table XVI), the semantic meanings of "嘉義(IV)" (ChiaYi, a location name) and "溪 (creek)" suffice to let a human know that "石弄 (Shinong)" is the name of the relevant creek, but this reasoning is beyond the capability of character *n*-grams.
(3) ORG – organization (23%; 55%). As in the case of LOC, more features, rather than simply character *n*-gram information, will be required for segmentation of organization names. For example, in Table XIX, "北京和利时公司(OOV)" (Beijing Hollysys Company), a company name, is incorrectly segmented into a sequence of three words ["北京(IV)" (Beijing), "和(IV)" (and), "利时公司(OOV)" (Lishi Company)]. The error occurs because "和" often serves as a conjunction in Chinese. Again, semantic analysis is required to know that a location and a company name would not normally be connected by "和(IV)" (and).
(4) Chinese Name (9%; 28%). Unlike in English, Chinese family names (mostly composed of one or two characters) come before given names (again, usually made up of one or two characters). Chinese family names compose a closed set: only 504 surnames are listed in the book "Hundred Family Surnames"[18], written about 1,000 years ago. For example, in Table XIX, the Chinese name "任尧森(OOV)" is segmented into two words as "任(IV)" and "尧森(OOV)". However, if a given character is known to be a family-name-character, the chance that it will be tagged as "B" increases.

A named-entity recognizer, which usually incorporates many features beyond character *n*-gram information [Gao et al. 2005; Wu et al. 2005], should be helpful in correcting errors in this category. Such modules should thus be used to provide candidate information for word segmentation models; however, this work is beyond the scope of the present paper.

(B) *Not Using Character-Type Information* (24.2%; 36.1%): Some OOV errors can be corrected if character-type information (indicating if the character is a digit, a punctuation symbol, a foreign-character, or a Chinese-character) is utilized. As an example for numbers, the numerical string ["二三六六一四一六(OOV)" (23661416)] shown in Table XVIII, which is "23661416" written in Chinese, is wrongly segmented into ["二三六六(OOV)" (2366)] and ["一四一六(OOV)" (1416)] as two OOV words. Also, "12.4(OOV)" (shown in Table XIX) is incorrectly segmented into "12.(OOV)" and

[18]See http://en.wikipedia.org/wiki/Hundred_Family_Surnames.

"4(IV)". Both errors can be corrected if we realize that all the associated characters are characters related to numerical expressions. As an example for punctuation symbols, the word ["光棍(IV)" (single man)] and its following word ["：(OOV)"], a colon symbol, are currently grouped together as one OOV non-word ["光棍：(NW)"] (shown in Table XVIII). Of course, this error can also be corrected if we know that "：(OOV)" is a punctuation symbol. Similarly, two words [" 第一(IV)" (the first)] and ["′(OOV)"] are incorrectly grouped into [第一′(NW)] in Table XIX; and this error can also be corrected using character-type information. Lastly, as an example for foreign-characters, the Chinese word ["英語(IV)" (English)] and its following English word ["valley(OOV)"], an English string, are currently grouped together as one OOV non-word ["英語valley(NW)"] (shown in Table XVIII). This error, too, can be corrected if we are aware that ["valley(OOV)"] is an English string. Similarly, ["发明(IV)" (invent)] and ["c c e d (OOV)"] in Table XIX will be correctly separated if the English string ["c c e d (OOV)"] can be recognized as a string of foreign characters.

(C) *Not Adopting Prefix/Suffix Information* (15.7%; 10.3%): Some OOV errors with prefixes or suffixes are likely to be corrected if the information concerning prefixes and suffixes can be utilized. As two examples in Table XVIII and Table XIX, the prefixed word "**非**種子" (unseeded) and the suffixed word "沙漠**化**" (desertization) are wrongly segmented into sequences of two words ["非(IV)" (un-), "種子(IV)" (seed)] and ["沙漠(IV)", (desert) "化(IV)" (-ization)] respectively. However, if we know that "非" (un-) is a prefix-character, then the chance that "非" is tagged as "B" will increase. Similarly, if "化" (-ization) is known to be a suffix-character, then the likelihood of "化" being tagged as "E" will increase as well. Clearly, it would be beneficial to integrate relevant features into the current models.

(D) *Idioms* (14.4%; 11.7%): Chinese idioms are special words which often contain four characters, and they form a nearly closed-set, which grows very slowly. According to the most stringent definition, there are about 5,000 such idioms in the Chinese language, though some dictionaries list over 20,000[19]. As an example in Table XVIII, the Chinese idiom "巧奪天工(OOV)" (wonderful workmanship excelling nature) is incorrectly segmented into a sequence of three words ["巧(IV)", "奪(IV)", "天工(NW)"]. A similar example "好高骛远(OOV)" (reach for what is beyond one's grasp) can be also found in Table XIX. If we recognized this string as an idiom via a pre-constructed table, then the sequence ["决不(IV)", "好高骛远(OOV)"] would not be segmented into ["决(IV)", "不好(IV)", "高骛(NW)", "远(IV)"].

(E) *Others* (15.2%; 16.4%): All remaining errors belong to this category. There are various sub-classes according to the causes, but the portion of each sub-class is small. Thus only some typical sub-classes, comprised of errors with clear linguistic causes, are discussed here. The first percentage in parentheses following the sub-class names indicates the corresponding ratio within a given sub-category for the CITYU corpus, and the second number denotes that for the MSR corpus. (1) *Split Compound* (7%, 4%): Since the distance between elements of a split compound is usually beyond trigram scope in either the generative or discriminative model (i.e., usually indicates a long-distance dependency), neither model can handle this problem. For an example in the MSR corpus, ["当(IV)" (when), "日全食(OOV)" (solar total eclipses; total solar eclipses), "时(IV)"(time)] is segmented into ["当 日(IV)" (this day), "全 食(OOV)" (complete eclipses), "时(IV)"(time)], because "当日" is an IV word. However, humans will know not to treat "当日" as a group if the end of the sentence is "时" (when), because

[19]http://en.wikipedia.org/wiki/Chinese_idiom

"当……时" (at the time/when) is an indicator of adverbial of time. A list of split compounds (or a syntactic parser with an appropriate grammar) would be required to resolve such errors. (2) *Abbreviations* (6%, 13%): Chinese abbreviations are mainly formed in three major ways: reduction, elimination, and generalization [Lee 2005]. The abbreviation ["双保(OOV)" (double insurance)] (shown in Table XIX) is formed via generalization: its full name in the MSR corpus is "保夏粮丰收, 保春播出苗" (Ensure the summer harvest; Ensure the spring seedling emergence). Compared with other sub-categories, abbreviations are much more difficult to correct. Different strategies would need to be adopted for various formation patterns. Related information can be found in Chang and Lai [2004] and Sun and Wang [2006]; (3) *Reduplication* (2%; 2%), which includes "AABB" and "ABAB" two forms (where A and B are different Chinese characters); related features have been adopted in Tseng et al. [2005] and Andrew [2006]. Table XVIII gives an example of "AABB": ["出出入入(OOV)" (go out and come in; literally go out, go out, come in, come in)] is incorrectly split into ["出(IV)" (go out)] and ["出入入(NW)"]. It seems a collection of these patterns will be required to solve these errors. The remaining sub-classes are sparse, and thus will be skipped here.

Among the various categories mentioned above, sub-category (B) is selected for further improvement in this article, as it occupies the biggest portion in MSR (36.1%) and ranks the second in CITYU (24.2%). Furthermore, it can be implemented without additional resources such as prefix/suffix lists, idiom tables, etc. Lastly, the CIPS-SIGNAN Bakeoff 2010 [Zhao and Liu 2010] allowed the use of character-type information. There are two ways to utilize character-type information in our proposed Joint model: (1) Write rules for handling numerical expressions, punctuations, and English strings during construction of the word lattice (by constraining the possible candidates); (2) Regard character-type information as a feature, and then integrate that information into the character-based generative/discriminative model. Since various rule-sets would be required for different corpora (according to their different segmentation criteria), method (2) is preferred, as no rule modification will be needed if criteria are updated: only the parameters need be retrained. The following section shows how much improvement can be achieved if character-type information is integrated into the Joint model as just suggested.

### 5.2. With Number, Punctuation, and Foreign Character Features

As mentioned, the knowledge of character-type can be integrated into the model as an additional feature, which classifies a given character into five different types: (1) a Chinese character; (2) a punctuation-symbol; (3) an Arabic digit; (4) a Chinese numeral; and (5) a foreign character. Since the discriminative approach (under the ME framework) is capable of easily incorporating additional features, this character-type feature can be directly integrated into the character-based discriminative model. The new feature template (d) is thus added to the original list as follows:

$$
\begin{aligned}
&(a)\ C_n(n = -2, -1, 0, 1, 2);\\
&(b)\ C_nC_{n+1}(n = -2, -1, 0, 1);\\
&(c)\ C_{-1}C_1;\\
&(d)\ T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)
\end{aligned}
$$

Where templates (a) $\sim$ (c) are the same as those used in the closed-test mentioned before, and $T(C_i)$ represents the corresponding character-type mentioned above for $C_i$. For example, when considering the punctuation-symbol "," in the character sequence

"八月, 阿 Q" , its feature $T(C_{-2})T(C_{-1})T(C_0)T(C_1)T(C_2)$ will be set to "41215", where each individual digit indicates the corresponding character-type defined above.

By contrast, incorporating additional features is more complicated for the generative approach. The original character-tag pair $[c, t]$ is first expanded into $[c, y, t]$, where letter "$y$" denotes the corresponding character-type (i.e., the class of the given character) and is the same as $T(C_i)$ mentioned above. Afterwards, this new $[c, y, t]$ trigram model is the log-linear integration of two simpler models (named Method 1, and shown below), since we would like to retain the advantages of the original model for handling IV words.

$$\log P([c, y, t]_i|[c, y, t]_{i-2}^{i-1}) \approx \beta \times \log P([c, t]_i|[c, t]_{i-2}^{i-1}) + (1 - \beta) \times \log P([y, t]_i|[y, t]_{i-2}^{i-1}) \quad (8)$$

The above formulation integrates the primary character-based generative model with a new character-type-based generative model. Since many character-bigrams of numerical or foreign strings cannot be covered in the training-set, this new model should play an essential role when such strings are encountered in a testing-set. The parameter $\beta$ ($0.0 \leq \beta \leq 1.0$) is the weight for the $[c, t]$ trigram factor, obtained from the cross-validation set (described in Section 3.2.3).

An alternative method for incorporating character-type information into the generative model, to be designated Method 2, is to pre-convert various foreign alphabets, Arabic digits, and Chinese numerical characters into meta-foreign characters, meta-Arabic-numeral characters, and meta-Chinese-numeral characters, respectively. Note that the punctuations are not pre-converted into meta-punctuation characters, because most punctuations have been seen in training-sets (except the CITYU corpus), and different punctuation symbols behave differently (especially in the computer domain, which includes various path-names); therefore, various punctuation symbols will not be pooled together into a meta-class. Otherwise, we proceed as we have previously done for the generative model for "Closed" tests.

The segmentation results of the above open-test approaches (adopting the additional character-type feature) are shown in Table XX. In this table, the symbol "/" separates the results of closed-test and open-test. The "Generative-1" rows show the segmentation results of the original generative trigram model in "Closed" tests and of the Generative-1 model (with Method 1) in "Open" tests. Likewise, the "Generative-2" rows give the results of the original generative trigram model in "Closed" tests and the Generative-2 model (with Method 2) in "Open" tests. The results show that the character-type feature is useful in improving the performance of generative models for both Methods 1 and 2; and that Method 2 is superior to Method 1 (with 0.952 vs. 0.948 in Overall $F$-score). Also, Method 2 is much better than Method 1 in the PKU(ucvt.) corpus (0.951 vs. 0.938 in F-score), due to the problem of character-encoding-inconsistency mentioned above: Arabic digits and English characters are encoded differently across its training and testing sets. This inconsistency problem exists for Method 1, but not for Method 2. This difference arises because Method 1 still uses the character-tag-pair trigram as a factor, while in Method 2 English characters and Arabic numerical characters are pre-converted into the meta-foreign character and the meta-Arabic-numeral character, respectively.

In addition, we test Method 2 with pre-conversion of various punctuation symbols into the same meta-punctuation character. Compared with the original Method 2, which did not pool the various punctuation-symbols, this new modification significantly improves the $F$-score from 0.946 to 0.950 in the CITYU corpus (not shown in Table XX), because punctuation symbols are encoded inconsistently in this corpus. However, this trick gives no improvement in other corpora, and even causes some minor deterioration. The degradation occurs because punctuation symbols are not always

Table XX. Segmentation Results with the Character-Type Feature. The Symbol "/" in Each Entry Separates the Performance of "Closed" Tests and "Open" Tests (Which Use This Additional Character-Type Feature). The Associated Weights of "Closed" and "Open" Tests Are Shown in the Form "A / B" in the "Weight" Column. (A for "Closed" Tests and Is Just the Alpha-Value, and B for "Open" Tests Which Would Be the Beta-Values for the Generative Model and Would Be Alpha-Values for Other Cases)

| Corpus | Model | Weight | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|---|
| **AS** | Generative-1 | NA/0.80 | 0.958/0.959 | 0.937/0.940 | 0.947/0.949 | 0.516/0.532 | 0.978/0.978 |
| | Generative-2 | NA/NA | 0.958/0.958 | 0.937/0.942 | 0.947/0.950 | 0.516/0.554 | 0.978/0.977 |
| | Discriminative | NA/NA | 0.956/0.957 | 0.946/0.949 | 0.951/0.953 | 0.709/0.729 | 0.967/0.967 |
| | Discriminative-Plus | NA/NA | 0.960/0.961 | 0.948/0.952 | 0.954/0.957 | 0.680/0.703 | 0.973/0.973 |
| | Joint-1 | 0.30/0.50 | 0.962/0.964 | 0.950/0.953 | 0.956/0.958 | 0.679/0.680 | 0.975/0.977 |
| | Joint-2 | 0.30/0.30 | 0.962/0.964 | 0.950/0.954 | 0.956/0.959 | 0.679/0.710 | 0.975/0.975 |
| | Joint-Plus-1 | 0.35/0.40 | 0.963/0.965 | 0.949/0.953 | 0.956/0.959 | 0.652/0.677 | 0.977/0.978 |
| | Joint-Plus-2 | 0.35/0.35 | 0.963/0.964 | 0.949/0.954 | **0.956/0.959** | 0.652/0.684 | 0.977/0.977 |
| **CITYU** | Generative-1 | NA/0.85 | 0.951/0.953 | 0.937/0.938 | 0.944/0.946 | 0.611/0.639 | 0.978/0.978 |
| | Generative-2 | NA/NA | 0.951/0.953 | 0.937/0.939 | 0.944/0.946 | 0.611/0.630 | 0.978/0.979 |
| | Discriminative | NA/NA | 0.940/0.953 | 0.944/0.952 | 0.942/0.953 | 0.709/0.816 | 0.959/0.964 |
| | Discriminative-Plus | NA/NA | 0.951/0.960 | 0.952/0.958 | 0.952/0.959 | 0.720/0.805 | 0.970/0.973 |
| | Joint-1 | 0.60/0.65 | 0.957/0.962 | 0.951/0.954 | 0.954/0.958 | 0.691/0.742 | 0.979/0.980 |
| | Joint-2 | 0.60/0.40 | 0.957/0.964 | 0.951/0.959 | 0.954/0.962 | 0.691/0.791 | 0.979/0.978 |
| | Joint-Plus-1 | 0.50/0.40 | 0.959/0.965 | 0.952/0.958 | 0.956/0.961 | 0.700/0.779 | 0.980/0.979 |
| | Joint-Plus-2 | 0.50/0.35 | 0.959/0.965 | 0.952/0.959 | **0.956/0.962** | 0.700/0.785 | 0.980/0.980 |
| **MSR** | Generative-1 | NA/1.0 | 0.973/0.973 | 0.966/0.966 | 0.970/0.970 | 0.560/0.560 | 0.985/0.985 |
| | Generative-2 | NA/NA | 0.973/0.975 | 0.966/0.969 | 0.970/0.972 | 0.560/0.633 | 0.985/0.985 |
| | Discriminative | NA/NA | 0.957/0.959 | 0.963/0.964 | 0.960/0.962 | 0.720/0.744 | 0.964/0.965 |
| | Discriminative-Plus | NA/NA | 0.965/0.967 | 0.967/0.970 | 0.966/0.968 | 0.675/0.724 | 0.973/0.973 |
| | Joint-1 | 0.60/0.60 | 0.974/0.976 | 0.971/0.973 | 0.972/0.975 | 0.659/0.709 | 0.983/0.983 |
| | Joint-2 | 0.60/0.60 | 0.974/0.976 | 0.971/0.974 | 0.972/0.975 | 0.659/0.721 | 0.983/0.983 |
| | Joint-Plus-1 | 0.65/0.55 | 0.975/0.976 | 0.970/0.974 | 0.972/0.975 | 0.632/0.699 | 0.984/0.984 |
| | Joint-Plus-2 | 0.65/0.65 | 0.975/0.977 | 0.970/0.974 | **0.972/0.975** | 0.632/0.702 | 0.984/0.985 |
| **PKU (ucvt.)** | Generative-1 | NA/0.75 | 0.929/0.933 | 0.932/0.938 | 0.931/0.936 | 0.435/0.493 | 0.959/0.960 |
| | Generative-2 | NA/NA | 0.929/0.952 | 0.932/0.951 | 0.931/0.952 | 0.435/0.703 | 0.959/0.967 |
| | Discriminative | NA/NA | 0.922/0.937 | 0.940/0.945 | 0.931/0.941 | 0.619/0.738 | 0.940/0.949 |
| | Discriminative-Plus | NA/NA | 0.934/0.945 | 0.949/0.951 | 0.941/0.948 | 0.649/0.735 | 0.951/0.958 |
| | Joint-1 | 0.60/0.70 | 0.935/0.946 | 0.946/0.952 | 0.941/0.949 | 0.561/0.692 | 0.958/0.961 |
| | Joint-2 | 0.60/0.65 | 0.935/0.955 | 0.946/0.957 | 0.941/0.956 | 0.561/0.757 | 0.958/0.967 |
| | Joint-Plus-1 | 0.60/0.75 | 0.937/0.948 | 0.947/0.952 | 0.942/0.950 | 0.556/0.686 | 0.960/0.964 |
| | Joint-Plus-2 | 0.60/0.60 | 0.937/0.956 | 0.947/0.958 | **0.942/0.957** | 0.556/0.766 | 0.960/0.967 |
| **PKU (cvt.)** | Generative-1 | NA/0.75 | 0.952/0.952 | 0.951/0.951 | 0.951/0.952 | 0.502/0.521 | 0.968/0.968 |
| | Generative-2 | NA/NA | 0.952/0.952 | 0.951/0.951 | 0.951/0.952 | 0.502/0.510 | 0.968/0.968 |
| | Discriminative | NA/NA | 0.939/0.943 | 0.951/0.952 | 0.945/0.948 | 0.685/0.690 | 0.948/0.952 |
| | Discriminative-Plus | NA/NA | 0.949/0.951 | 0.958/0.958 | 0.953/0.954 | 0.674/0.680 | 0.958/0.961 |
| | Joint-1 | 0.60/0.70 | 0.954/0.954 | 0.958/0.958 | 0.956/0.956 | 0.616/0.651 | 0.966/0.964 |
| | Joint-2 | 0.60/0.65 | 0.954/0.955 | 0.958/0.958 | 0.956/0.957 | 0.616/0.621 | 0.966/0.967 |
| | Joint-Plus-1 | 0.60/0.75 | 0.955/0.955 | 0.958/0.958 | 0.957/0.957 | 0.610/0.642 | 0.967/0.966 |
| | Joint-Plus-2 | 0.60/0.60 | 0.955/0.956 | 0.958/0.958 | **0.957/0.957** | 0.610/0.617 | 0.967/0.968 |
| **Overall** | Generative-1 | NA/NA | 0.953/0.955 | 0.946/0.948 | 0.949/0.951 | 0.510/0.538 | 0.973/0.974 |
| | Generative-2 | NA/NA | 0.953/0.959 | 0.946/0.952 | 0.949/0.955 | 0.510/0.611 | 0.973/0.975 |
| | Discriminative | NA/NA | 0.944/0.950 | 0.949/0.952 | 0.947/0.951 | 0.680/0.740 | 0.956/0.959 |
| | Discriminative-Plus | NA/NA | 0.952/0.957 | 0.955/0.957 | 0.953/0.957 | 0.676/0.726 | 0.965/0.967 |
| | Joint-1 | NA/NA | 0.957/0.960 | 0.955/0.958 | 0.956/0.959 | 0.633/0.685 | 0.971/0.973 |
| | Joint-2 | NA/NA | 0.957/0.963 | 0.955/0.960 | 0.956/0.962 | 0.633/0.722 | 0.971/0.974 |
| | Joint-Plus-1 | NA/NA | 0.958/0.961 | 0.955/0.959 | 0.957/0.960 | 0.621/0.691 | 0.973/0.974 |
| | Joint-Plus-2 | NA/NA | 0.958/0.964 | 0.955/0.961 | **0.957/0.962** | 0.621/0.711 | 0.973/0.975 |

tagged with "S": some, like "/", "-", "_", etc., may be tagged with "M". Thus, to further improve the performance of the generative model, we will need to categorize punctuation symbols according to their different behavior patterns.

Table XX also gives the segmentation results of the modified Discriminative model and the modified Discriminative-Plus model. Again, we see that character-type information is useful in improving the performance on all corpora and is most effective in the CITYU and PKU(ucvt.) corpora. Next, the modified generative and the modified Discriminative models are further integrated as in the previous "Closed" tests: the Generative-1 model and the modified Discriminative model are integrated into the character-based Joint-1 model (abbreviated as Joint-1); the Generative-1 model and the modified Discriminative-Plus model are integrated into the character-based Joint-Plus-1 model (abbreviated as Joint-Plus-1); the Generative-2 model and the modified Discriminative model are integrated into the character-based Joint-2 model (abbreviated as Joint-2); and the Generative-2 model and the modified Discriminative-Plus model are integrated into the character-based Joint-Plus-2 model (abbreviated as Joint-Plus-2).

The segmentation results for the modified Joint models and the modified Joint-Plus models are also given in Table XX. We see that the Joint-Plus-2 model achieves the best *F*-score on each corpus. It is marked for visibility. Also, improvement is especially noticeable in the CITYU and PKU(ucvt.) corpora, because the punctuations are inconsistent in the CITYU corpus, as are the Arabic numbers and English characters in the PKU(ucvt.) corpus. In addition, the Joint-2 model holds an apparent edge over the modified Discriminative model on both the closed-tests and the open-tests. For the overall *F*-score, the Joint-2 model outperforms the Discriminative model by 0.9 percent on the closed-test and 1.1 percent on the open-test. The advantage of using the Joint-Plus-2 model over the Discriminative-Plus model is smaller but still significant (in comparison to that of the Joint model over the Discriminative model). For the overall *F*-score, the Joint-Plus model exceeds the Discriminative-Plus model by 0.4 percent on the closed-test, while the advantage rises up to only 0.5 percent on the open-test.

Although the overall improvement does not seem very impressive, most errors related to numerical expressions, punctuations, and foreign character sequences are indeed fixed. With the character-based Joint-2 model, 129 out of the original 143 related errors in the CITYU corpus and 185 out of 203 in the MSR corpus have been tagged correctly with character-type information. The remaining errors (totaling 32 from both the CITYU corpus and the MSR corpus) can be classified into two sub-classes: (1) *Missing Space Character* (53%), and (2) *Grouping with Quantifier* (47%). The first subclass indicates cases in which the usual space characters are missing. For example, in strings such as ["2 . 1 9 9 2 年" (Item 2, in 1992)] and ["i n t e r n e t s e r v i c e p r o v i d e r" (Internet service provider)], the spaces normally used to separate different words are absent; however, the benchmark segments them into ["2(IV)" (Item 2), ".(IV)" (a period symbol), "1 9 9 2 年(IV)" (In year 1992)] and ["i n t e r n e t (OOV)", "s e r v i c e (OOV)", "p r o v i d e r (OOV)"], respectively. For these two examples, the former is wrongly recognized as a false numerical expression ["2 . 1 9 9 2 年(OOV)"], while the latter is identified as only one English OOV word because there is no space character. The errors in this sub-class are due to incorrect text format conversion, and are beyond the capability of the character-type feature.

The second sub-class (*Grouping with Quantifier*, 47%) denotes cases in which not all the Chinese characters following a number should be separated from the number. For example, in the MSR corpus, some Chinese measurement-units such as "个" (a Chinese quantifier for counting items) and "元" (Yuan, the currency unit for RMB), are often grouped with numbers, such as ["3 4 个" (34), "县" (county)] and ["5

Table XXI. Statistics of Corpora of Simplified Chinese Text

| Corpora | | Characters | Tokens | Word Types | OOV Rate |
|---|---|---|---|---|---|
| **Training** | Labeled PKU-News | 1,820,456 | 1,109,947 | 55,303 | NA |
| | Unlabeled-Literature | 100,352 | NA | NA | NA |
| | Unlabeled-Computer | 103,764 | NA | NA | NA |
| **Testing** | Literature | 50,637 | 35,736 | 6,364 | 0.069 |
| | Computer | 53,382 | 35,319 | 4,150 | 0.152 |
| | Medicine | 50,969 | 31,490 | 5,076 | 0.110 |
| | Finance | 53,253 | 33,028 | 4,918 | 0.087 |

8 0 万元" (5,800,000 RMB)]. However, as the character-type adopted above cannot distinguish "Measurement-Unit" characters from others, our model incorrectly groups two consecutive uni-character words ["5 (IV)" (five)] and ["省(IV)" (province)] into one OOV word ["5 省(OOV)" (five provinces)]. Similarly, the two-word sequence ["9 9 3 万(OOV)" (9,930,000), "工(IV)" (working-day)] is grouped into one OOV word as ["9 9 3 万工(OOV)" (9,930,000 working-day)], because "省" and "工" are often tagged with "E" in the training-set. Apparently Chinese "measurement-unit" characters should be grouped together and treated as an additional character-type. Since the new character-type does not target the errors mentioned above, we can conclude that it is actually quite effective.

## 6. DOMAIN ADAPTATION

As shown in the above section, the proposed Joint-2 model and Joint-Plus-2 model achieve better performance than previously reported models on all corpora tested. However, both the training-sets and the testing-sets are from the same domain. To test the cross-domain performance of the proposed models, we will adopt the Simplified Chinese Text corpora of different domains provided by CIPS-SIGNAN Bakeoff 2010 [Zhao and Liu 2010]. Four testing sets are provided by this Bakeoff, each of them from a different domain: literature, computers, medicine, and finance. However, only two of them (literature and computers) are provided with unlabeled training sets. In addition, it turns out that the labeled training data of Simplified Chinese Text in this Bakeoff is the same as the PKU training data of SIGHAN Bakeoff 2005, which comes from the News domain. The statistics of the SIGHAN 2010 corpora are shown in Table XXI.

Since the character-based Joint-Plus-2 model (with the character-type feature) achieves the best performance so far, we will conduct the domain adaptation test on this model only, for simplicity. Furthermore, the cross-domain performance and the effect of domain adaptation will not be compared with the systems reported in SIGNAN Bakeoff-2010: because strict regulation was not provided, these systems adopted various rule-sets and additional information (e.g., the pinyin spelling of each character), so that it would be difficult to conduct fair comparisons between them. Finally, again for simplicity, we fix the weight of the generative score in our Joint-Plus-2 model during the training procedure (to$\alpha$ = 0.60, acquired from the PKU development set in the last section).

To incorporate unlabeled training data, Mcclosky et al. [2006] adopted a semi-supervised learning method called *self-training*. The issue of convergence during such learning has been studied in Haffari and Sarkar [2007] and Culp and Michailidis [2008]. In the present study, we adopt a similar semi-supervised learning method, which proceeds as follows: The initial segmenter is first trained with a pre-labeled corpus; then this trained segmenter is used to segment the unlabeled training data;

**Given**:
- Labeled training corpus: $L_0$
- Unlabeled training corpus: $U$

Use $L_0$ to train a segmenter $S_0$;
Use $S_0$ to segment the unlabeled corpus $U$ and then get labeled corpus $U_0$;
**for** $i$ = 1 to $K$ **do**
  Add $U_{i-1}$ to $L_0$ and get a new corpus $L_i$;
  Use $L_i$ to train a new segmenter $S_i$;
  Use $S_i$ to segment the unlabeled corpus $U$ and then get labeled corpus $U_i$;
  **if** convergence criterion meets
    **break**
**end for**
**Output**: the last segmenter $S_K$

Fig. 3. Semi-supervised learning algorithm.

Table XXII. Segmentation Results Under Different Conditions. **Joint-Plus-2**: The Character-Based Joint-Plus-2 Model; Joint-Plus-2+S: Joint-Plus-2 Model with Semi-Supervised Learning. The Pre-Labeled Training Set is in News Domain

| Domain | Condition | OOV Rate | R | P | F | $R_{OOV}$ | $R_{IV}$ |
|---|---|---|---|---|---|---|---|
| **News (In domain)** | Joint-Plus-2 | 0.035 | 0.956 | 0.958 | 0.957 | 0.617 | 0.968 |
| **Finance** | Joint-Plus-2 | 0.087 | 0.961 | 0.954 | 0.958 | 0.812 | 0.976 |
| **Medicine** | Joint-Plus-2 | 0.110 | 0.930 | 0.912 | 0.921 | 0.644 | 0.965 |
| **Literature** | Joint-Plus-2 | 0.069 | 0.938 | 0.936 | 0.937 | 0.615 | 0.962 |
| | Joint-Plus-2+S | 0.056 | 0.940 | 0.938 | **0.939** | 0.621 | 0.964 |
| **Computer** | Joint-Plus-2 | 0.152 | 0.944 | 0.927 | 0.935 | 0.759 | 0.977 |
| | Joint-Plus-2+S | 0.079 | 0.951 | 0.929 | **0.940** | 0.790 | 0.980 |

and finally, the resulting data is regarded as labeled data (denoted by $U_i$ for the $i$-th iteration) and added to the original pre-labeled data to form a new training set. The above procedure is repeated until the convergence criterion is met. Currently, the iterations stop when the similarity between the results of two consecutive iterations ($U_{i-1}$ and $U_i$) reaches a high level ($F$-score $> 0.9999$, evaluated by treating $U_{i-1}$ as the benchmark and $U_i$ as the testing set). In our observation, the above procedure converges quickly, with only three or four iterations for both Literature and Computer corpora. The flow of the adopted semi-supervised learning procedure is shown in Figure 3.

Table XXII gives the segmentation results of various cross-domain testing-sets under different conditions. The performance of in-domain testing-set is shown in the first row in Table XX, labeled PKU(cvt.). For cross-domain testing-sets, the performance *without* conducting domain adaptation is given in the rows labeled "Joint-Plus-2"; in contrast, the performance *with* domain adaptation is given in the rows labeled "Joint-Plus-2+S". Marked entries denote that the performance with domain adaptation is significantly better than without.

In comparison with the in-domain performance (SIGHAN 2005), the cross-domain performance (except for the Finance domain) degrades significantly, because many technical terms in different domains cannot be covered by the training-set of the News domain. (This analysis is supported by the dramatic increment of OOV Rate in Table XXII.) Among the various domains, the Finance domain obtains the best result, which is even a bit better than the in-domain performance. The reason is that most OOV words in the Finance domain are numerical expressions, which can be easily handled by exploiting character-type information. The worst domain is Medicine,

because there are many OOV medical terms. Table XXII also shows that conducting domain adaptation significantly improves the performance of the Computer domain (raising $F$-score from 0.935 to 0.940, which is statistically significant); however, performance improves only slightly in the Literature domain (from 0.937 to 0.939, also statistically significant). This is because the difference between News and Computer is much greater than that between News and Literature. The unlabeled training data from the Computer domain is thus more effective in providing information concerning computer-related technical terms, which are not found in the original News domain.

Table XXII also gives the OOV rates for cross-domain testing-sets with and without domain adaptation. In the case with domain adaptation, the OOV words in the testing set are checked against the union of the pre-labeled and the unlabeled training corpora. For the unlabeled training corpus, the OOV words are checked against its final segmentation results after convergence. We see that the OOV rates of the Computer domain decreases sharply after domain adaptation (decreasing from 0.152 to 0.079). This decrease is much more obvious than that seen in the Literature domain (from 0.069 to 0.056). However, the improvement (of $F$-score) with domain adaptation in the Computer domain is only 0.5%. Upon analyzing, we find that most of the OOV words in the Computer domain, which are covered by the unlabeled training data, are numbers and English strings; and most of these words have already been correctly handled by the Joint-Plus-2 model before the semi-supervised learning. Thus, even though the OOV rates decrease sharply, the improvement with domain adaptation in the Computer domain is still minor.

On the other hand, the Computer domain performs better than the Literature domain after domain adaptation (0.940 versus 0.939 in $F$-score). Even its OOV rate is higher (0.079 versus 0.056, after adaptation). This observation can be explained as follows: after numbers and English strings are excluded, the number of remaining OOV words covered by the unlabeled training data in the Literature domain is only 177, while the number in the Computer is 1,011. Further, the unlabeled corpora provided by the SIGHAN Bakeoff-2010 are very small compared with the pre-labeled corpora. The gain from the unlabeled data should thus be greater if a larger unlabeled training corpus is provided.

## 7. RELATED WORK

The word-based generative model is a well-known approach used in many successful applications [Gao et al. 2003; Zhang et al. 2003]. However, Zhang et al. [2006] has shown that, while this approach performs excellently for IV words, it is quite weak for OOV words. To handle OOV words appropriately, Zhang et al. [2003] adopted a procedure for incorporating other knowledge of various sorts. In contrast to the word-based generative model, the word-based discriminative model is adopted only in [Zhang and Clark 2007]. This study utilizes a discriminative perceptron algorithm [Collins 2002] to generate word candidates with features related to both words and characters. This model reportedly achieves state-of-the-art performance on some of the corpora tested.

We can now consider the character-based tagging model [Xue 2003]. This method has become dominant because it can tolerate OOV words. As a consequence, in the SIGHAN Bakeoff 2005 [Emerson 2005], all systems ranked in the first tier [Asahara et al. 2005; Tseng et al. 2005] are based upon it. However, the performance of this approach is still quite unsatisfactory [Huang et al. 2007], and many studies have tried to improve it. For example, Peng et al. [2004] integrates domain knowledge, such as additional word lists, character lists, and "part-of-speech character lexicons" (including title prefixes, title suffixes, Chinese surnames, etc.) into the framework

of the *conditional random field* (CRF) [Lafferty et al. 2001]. In a similar spirit, Tseng et al. [2005] adopts a large number of linguistic features, such as features representing morphological and character reduplication. Zhao et al. [2006] reports that the six-tag set using a three-character window outperforms the standard four-tag set with a five-character window. Finally, Li and Sun [2009] uses punctuation as implicit annotation to improve OOV word recognition.

In general, the character-based tagging model yields high recall of OOV words ($R_{OOV}$) but unsatisfactory recall of IV words ($R_{IV}$). To overcome this weakness, Zhang et al. [2006] propose a sub-word tagging approach, and Fu et al. [2008] adopts a morpheme-based chunking approach. Sun et al. [2009] incorporates hybrid information, based on both word and character sequences, with a latent variable model. Sun [2010] compares the performance of the word-based discriminative model and the character-based discriminative model, and then uses a bagging approach to combine the outputs of these two models. Recently, Zhang and Clark [2011] use a single discriminative model to adopt both word-based and character-based features. Notably, while the character-based model can be associated with the generative form as argued here, there are no related papers in the literature. (The only exceptions are our own conference articles [Wang et al. 2009, 2010], which are greatly reduced versions of the current article.)

With regard to integration of generative and discriminative models, a hybrid generative/discriminative approach was proposed by Jaakkola and Haussler [1999]. In that study, the kernel function for the discriminative model was extracted from a generative model. Also, Raina et al. [2004] divide the feature vector into sub-vectors based on naïve Bayesian assumptions, and then combines these sub-generative models with discriminative learning. More recently, Jiampojamarn et al. [2010] integrated a generative joint n-gram model as binary features into the discriminative training. Specifically for WS, Andrew [2006] improves performance by adding generative features into a semi-Markov CRF framework. However, the gain from adopting these additional generative components has been insignificant. As compared with these various approaches, our experiments have shown that our proposed log-linear interpolation is still the most effective way to combine the generative and the discriminative models for the WS problem, simple though it is.

Since some ambiguities of word segmentation require even the information of subsequent phases (e.g., POS tagging, parsing, etc.) to solve, some researchers do word segmentation jointly with subsequent tasks. For example, Shi and Wang [2007] incorporate the Part-of-Speech (POS) information in the WS procedure, and the best outputs are searched with the overall joint WS and POS probabilistic score. Similar works also include Jiang et al. [2008], Zhang and Clark [2008], Kruengkrai et al. [2009], Sun [2011], and Zhang and Clark [2011]. Furthermore, Li [2011] proposes a new paradigm for Chinese word segmentation, which do word segmentation, word inter-structure, and phrase parsing at the same time in a unified way. However, all of them require additional linguistic resources (e.g., a corpus annotated with POS or a Chinese Treebank).

This article differs from previous approaches in several ways. First, we propose a new model form including the character-tag-pair (as opposed to simply adding a few new features under the same discriminative framework). Second, we propose and test a simple but effective way to integrate the generative and discriminative models. Third, all state-of-the-art systems reported in the literature are checked and compared with our systems. Fourth, a complete and detailed error analysis is conducted, which clearly points out directions for future research. Fifth, we show the effect of adding character-type information on the SIGHAN Bakeoff 2005. Last, a semi-supervised learning method is proposed to conduct domain adaptation for word segmentation.

## 8. CONCLUSION

Since word segmentation is the first step for most Chinese NLP applications, WS errors will be carried forward into subsequent phases. Thus WS accuracy is crucial for Chinese NLP and should be raised as much as possible. The traditional word-trigram generative model can identify IV words quite well, but cannot handle OOV words. To address this issue, this article first proposes a new character-based generative model, which replaces word-based $n$-grams with character-tag-pair $n$-grams. As the vocabulary of characters is a closed-set, as opposed to the open-set of words, there will be no more unseen candidates if the training set is large enough. Thus the character-based approach can handle OOV words much better than the word $n$-gram approach ($R_{OOV}$ is significantly raised from 0.053 to 0.511). Experiments conducted on the second SIGHAN Bakeoff 2005 corpora have shown that the proposed character-based generative model not only achieves a good balance between IV words and OOV words, but also obtains competitive results with the widely adopted character-based discriminative model.

On the other hand, although the character-based discriminative approach handles OOV words better, given its ability to incorporate the future context as features (as generative models cannot), it fails to model the adhesion and dependency between adjacent characters within words (as the generative model does, and as humans are believed to do when segmenting words). It thus gives unsatisfactory performance for IV words. That is, the generative and discriminative approaches complement each other in handling IV words and OOV words. To take advantage of these complementary capacities, a joint model is thus further proposed to combine the character-based discriminative approach and the proposed character-based generative approach. A closed-test on the SIGHAN Bakeoff 2005 corpora shows that this joint model significantly outperforms all the state-of-the-art systems reported in the literature. Although the proposed approaches have been tested on Chinese corpora only, we believe that they should also be applicable to other languages with similar characteristics (e.g., Japanese and Korean), since no Chinese-specific features (e.g., prefixes, suffixes, or Chinese family names) are adopted in the models.

After the remaining errors of the Joint model were analyzed, we observed that many of them (24.2% of the OOV Critical section for the CITYU corpus, and 36.1% for MSR) were caused by failure to take the character-type into account – that is, failure to distinguish punctuation-symbols, Arabic numbers, and English characters from common Chinese characters. And indeed, after we incorporated such character-type information, most errors related to numerical expressions, English character sequences, and punctuations were corrected. We thus suspect that more character-related features (showing for example whether the character is a prefix or suffix, a surname-character, etc) should be added in the future if further improvement is required.

Finally, cross-domain performance has been evaluated on the SIGHAN 2010 corpora, and a semi-supervised method has been proposed for conducting domain adaptation. The results show that this approach is effective, especially when the mismatch between two domains is large.

## ACKNOWLEDGMENTS

The authors would like to extend their sincere thanks to Wenbin Jiang, Yue Zhang, and Ruiqiang Zhang for discussion and help with experiments. Also, many thanks go to Dekang Lin and another anonymous reviewer for their comments on weighting various features differently in ME training. In addition, we thank Behavior Design Corporation for allowing us to use their Generic-Beam-Search code. Last, sincere thanks to Mark Seligman for his careful revision work.

## REFERENCES


ANDREW, G. 2006. A hybrid markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*. 465–472.

ASAHARA, M., FUKUOKA, K., AZUMA, A., GOH, C.-L., WATANABE, Y., MATSUMOTO, Y., AND TSUZUKI, T. 2005. Combination of machine learning methods for optimum Chinese word segmentation. In *Proceedings of the 4th Workshop on Chinese Language Processing (SIGHAN'05)*. 134–137.

BERGER, A. L., DELLA PIETRA, V. J., AND DELLA PIETRA, S. A. 1996. A maximum entropy approach to natural language processing. *Comp. Linguist. 22*, 39–71.

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.

BISHOP, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer New York.

CHANG, J.-S. AND LAI, Y.-T. 2004. A preliminary study on probabilistic models for Chinese abbreviations. In *Proceedings of the 3rd Workshop on Chinese Language Learning (SIGHAN'04)*. 9–16.

CHEN, S. F. AND GOODMAN, J. 1998. An empirical study of smoothing techniques for language modeling. Tech. rep. TR-10-98, Harvard University Center for Research in Computing Technology.

COLLINS, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*. 1-8.

CULP, M. AND MICHAILIDIS, G. 2008. An iterative algorithm for extending learners to a semi-supervised setting. *J. Comp. Graph. Stat. 17*, 545–571.

EMERSON, T. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the 4th Workshop on Chinese Language Processing (SIGHAN'05)*. 123–133.

FU, G., KIT, C., AND WEBSTER, J. J. 2008. Chinese word segmentation as morpheme-based lexical chunking. *Inf. Sci. 178*, 2282–2296.

FUJINO, A., UEDA, N., AND SAITO, K. 2005. A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'05)*. 764–769.

GAO, J., LI, M., AND HUANG, C.-N. 2003. Improved source-channel models for Chinese word segmentation. In *Proceedings of the 41th Annual Meeting of Association of Computational Linguistics (ACL'03)*. 272–279.

GAO, J., LI, M., WU, A., AND HUANG, C.-N. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Comp. Linguist. 31*, 531–574.

HAFFARI, G. AND SARKAR, A. 2007. Analysis of semi-supervised learning with the Yarowsky algorithm. In *Proceedings of the 23rd Conference on Uncertainty in Arti?cial Intelligence (UAI'07)*.

HUANG, C.-R., ŠIMON, P., HSIEH, S.-K., AND PRVOT, L. 2007. Rethinking Chinese word segmentation: Tokenization, character classification, or word break identification. In *Proceedings of the 45th ACL Conference on Interactive Poster and Demonstration Sessions (ACL'07)*. 69–72.

JAAKKOLA, T. S. AND HAUSSLER, D. 1999. Exploiting generative models in discriminative classifiers. *Adv. Neural Inf. Proc. Syst.* 487–493.

JIAMPOJAMARN, S., CHERRY, C., AND KONDRAK, G. 2010. Integrating joint n-gram features into a discriminative training framework. In *Proceedings of the Language Technology Conference (NAACL'10)*. 697–700.

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 Association for Computational Linguistics (ACL'08)*. 897–904.

JOHNSON, M. 2001. Joint and conditional estimation of tagging and parsing models. In *Proceedings of the Association for Computational Linguistics (ACL'01)*. 314–321.

KOEHN, P. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*. 388–395.

KRUENGKRAI, C., UCHIMOTO, K., KAZAMA, J. I., 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 47th Annual Meetings of the ACL and the 4th IJCNLP of the AFNLP (ACL'09)*. 513–521.

LAFFERTY, J., MCCALLUM, A., AND PEREIRA, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (CML'01)*. 282–289.

LEE, H. W. D. 2005. *A Study of Automatic Expansion of Chinese Abbreviations*. The University of Hong Kong.

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 (ACL'11)*. 1405–1414.

LI, Z. AND SUN, M. 2009. Punctuation as implicit annotations for chinese word segmentation. *Comp. Linguist. 35*, 505–512.

LIANG, P. AND JORDAN, M. I. 2008. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. 584–591.

MCCLOSKY, D., CHARNIAK, E., AND JOHNSON, M. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT-NAACL'06)*. 152–159.

NG, A. Y. AND JORDAN, M. I. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Proceedings of the Conference on the Advances in Neural Information Processing Systems (NIPS'02)*. 14, 841–848.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*. 277–284.

PALMER, D. D. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter (ACL'97)*. 321–328.

PENG, F., FENG, F., AND MCCALLUM, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of International Conference on Computer Linguistics (COLING'04)*. 562–568.

RAINA, R., SHEN, Y., NG, A. Y., AND MCCALLUM, A. 2004. Classification with hybrid generative/discriminative models. *Adv. Neural Inf. Proc. Syst*.

SHI, Y. AND WANG, M. 2007. A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of IJCAI*.

STOLCKE, A. 2002. SRILM-an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'02)*. 311–318.

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 (ACL'11)*. 1385–1394.

SUN, X. AND WANG, H. 2006. Chinese abbreviation identification using abbreviation-template features and context information. Lecture Notes in Computer Science, vol. 4285, 245–255.

SUN, X., ZHANG, Y., MATSUZAKI, T., TSURUOKA, Y., AND TSUJII, J. I. 2009. A discriminative latent variable Chinese segmenter with hybrid word/character information. In *Proceedings of the Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL'09)*. 56–64.

TOUTANOVA, K. 2006. Competitive generative models with structure learning for NLP classification tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*. 576–584.

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 Workshop on Chinese Language Processing (SIGHAN'05)*. 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'09)*. 827–834.

WANG, K., ZONG, C., AND SU, K.-Y. 2010. A character-based joint model for Chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. 1173–1181.

Wu, Y., Zhao, J., Xu, B., and Yu, H. 2005. Chinese named entity recognition based on multiple features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*. 427–434.

XIONG, Y., ZHU, J., HUANG, H., AND XU, H. 2009. Minimum tag error for discriminative training of conditional random fields. *Inf. Sci. 179*, 169–179.

XUE, J. AND TITTERINGTON, D. M. 2008. Comment on "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes". *Neur. Proc. Lett. 28*, 169–187.

XUE, N. 2003. Chinese word segmentation as character tagging. *Comp. Linguist. Chinese Lang. Proc. 8*, 29–48.

XUE, N. AND SHEN, L. 2003. Chinese word segmentation as LMR tagging. In *Proceedings of the 2nd Workshop on Chinese Language Processing (SIGHAN'03)*. 176–179.

YEH, C.-L. AND LEE, H.-J. 1991. Rule-based word identification for Mandarin Chinese sentences-A unification approach. *Comp. Proc. Chinese Orient. Lang. 5*, 97–118.

ZHANG, H., YU, H., XIONG, D., AND LIU, Q. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the 2nd Workshop on Chinese Language Processing (SIGHAN'03)*. 184–187.

ZHANG, R., KIKUI, G., AND SUMITA, E. 2006. Subword-based tagging for confidence-dependent Chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'06)*. 961–968.

ZHANG, Y. AND CLARK, S. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'07)*. 840–847.

ZHANG, Y. AND CLARK, S. 2008. Joint Word Segmentation and POS Tagging using a Single Perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'08)*.

ZHANG, Y. AND CLARK, S. 2011. Syntactic processing using the generalized perceptron and beam search. *Comp. Linguist. 37*, 105–151.

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'04)*. 2051–2054.

ZHAO, H., HUANG, C.-N., LI, M., AND LU, B.-L. 2006. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC'06)*. 87–94.

ZHAO, H., HUANG, C.-N., LI, M., AND LU, B.-L. 2010. A unified character-based tagging framework for Chinese word segmentation. *Trans. Asian Lang. Inform. Process. 9*, 1–32.

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

ZONG, C. 2008. *Statistical Natural Language Processing*. Tsinghua University Press.

Received March 2011; revised June 2011; accepted August 2011