A Discriminative Latent Variable Chinese Segmenter with Hybrid Word/Character Information

Xu Sun
Department of Computer Science
University of Tokyo
sunxu@is.s.u-tokyo.ac.jp

Yaozhong Zhang
Department of Computer Science
University of Tokyo
yaozhong.zhang@is.s.u-tokyo.ac.jp

Takuya Matsuzaki
Department of Computer Science
University of Tokyo
matuzaki@is.s.u-tokyo.ac.jp

Yoshimasa Tsuruoka
School of Computer Science
University of Manchester
yoshimasa.tsuruoka@manchester.ac.uk

Jun'ichi Tsujii
Department of Computer Science, University of Tokyo, Japan
School of Computer Science, University of Manchester, UK
National Centre for Text Mining, UK
tsujii@is.s.u-tokyo.ac.jp

Abstract

Conventional approaches to Chinese word segmentation treat the problem as a character-based tagging task. Recently, semi-Markov models have been applied to the problem, incorporating features based on complete words. In this paper, we propose an alternative, a latent variable model, which uses hybrid information based on both word sequences and character sequences. We argue that the use of latent variables can help capture long range dependencies and improve the recall on segmenting long words, e.g., named-entities. Experimental results show that this is indeed the case. With this improvement, evaluations on the data of the second SIGHAN CWS bakeoff show that our system is competitive with the best ones in the literature.

1 Introduction

For most natural language processing tasks, words are the basic units to process. Since Chinese sentences are written as continuous sequences of characters, segmenting a character sequence into a word sequence is the first step for most Chinese processing applications. In this paper, we study the problem of Chinese word segmentation (CWS), which aims to find these basic units (words[1]) for a given sentence in Chinese.

Chinese character sequences are normally ambiguous, and out-of-vocabulary (OOV) words are a major source of the ambiguity. Typical examples of OOV words include named entities (e.g., organization names, person names, and location names). Those named entities may be very long, and a difficult case occurs when a long word W ($|W| \geq 4$) consists of some words which can be separate words on their own; in such cases an automatic segmenter may split the OOV word into individual words. For example, 国际自动控制联合会计算机委员会 (Computer Committee of International Federation of Automatic Control) is one of the organization names in the Microsoft Research corpus. Its length is 13 and it contains more than 6 individual words, but it should be treated as a single word. Proper recognition of long OOV words are meaningful not only for word segmentation, but also for a variety of other purposes, e.g., full-text indexing. However, as is illustrated, recognizing long words (without sacrificing the performance on short words) is challenging.

Conventional approaches to Chinese word segmentation treat the problem as a character-based la-

[1] Following previous work, in this paper, *words* can also refer to multi-word expressions, including proper names, long named entities, idioms, etc.

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 56–64,
Boulder, Colorado, June 2009.

beling task (Xue, 2003). Labels are assigned to each character in the sentence, indicating whether the character x_i is the start ($Label_i = B$), middle or end of a multi-character word ($Label_i = C$). A popular discriminative model that have been used for this task is the conditional random fields (CRFs) (Lafferty et al., 2001), starting with the model of Peng et al. (2004). In the Second International Chinese Word Segmentation Bakeoff (the second SIGHAN CWS bakeoff) (Emerson, 2005), two of the highest scoring systems in the closed track competition were based on a CRF model (Tseng et al., 2005; Asahara et al., 2005).

While the CRF model is quite effective compared with other models designed for CWS, it may be limited by its restrictive independence assumptions on non-adjacent labels. Although the window can in principle be widened by increasing the Markov order, this may not be a practical solution, because the complexity of training and decoding a linear-chain CRF grows exponentially with the Markov order (Andrew, 2006).

To address this difficulty, a choice is to relax the Markov assumption by using the semi-Markov conditional random field model (semi-CRF) (Sarawagi and Cohen, 2004). Despite the theoretical advantage of semi-CRFs over CRFs, however, some previous studies (Andrew, 2006; Liang, 2005) exploring the use of a semi-CRF for Chinese word segmentation did not find significant gains over the CRF ones. As discussed in Andrew (2006), the reason may be that despite the greater representational power of the semi-CRF, there are some valuable features that could be more naturally expressed in a character-based labeling model. For example, on a CRF model, one might use the feature "the current character x_i is X and the current label $Label_i$ is C". This feature may be helpful in CWS for generalizing to new words. For example, it may rule out certain word boundaries if X were a character that normally occurs only as a suffix but that combines freely with some other basic forms to create new words. This type of features is slightly less natural in a semi-CRF, since in that case local features $\varphi(y_i, y_{i+1}, x)$ are defined on pairs of adjacent words. That is to say, information about which characters are not on boundaries is only implicit. Notably, except the hybrid Markov/semi-Markov system in Andrew (2006)[2], no other studies using the semi-CRF (Sarawagi and Cohen, 2004; Liang, 2005; Daumé III and Marcu, 2005) experimented with features of segmenting *non*-boundaries.

In this paper, instead of using semi-Markov models, we describe an alternative, a latent variable model, to learn long range dependencies in Chinese word segmentation. We use the discriminative probabilistic latent variable models (DPLVMs) (Morency et al., 2007; Petrov and Klein, 2008), which use latent variables to carry additional information that may not be expressed by those original labels, and therefore try to build more complicated or longer dependencies. This is especially meaningful in CWS, because the used labels are quite coarse: $Label(y) \in \{B, C\}$, where B signifies *beginning a word* and C signifies *the continuation of a word*.[3] For example, by using DPLVM, the aforementioned feature may turn to "the current character x_i is X, $Label_i = C$, and $LatentVariable_i = LV$". The current latent variable LV may strongly depend on the previous one or many latent variables, and therefore we can model the long range dependencies which may not be captured by those very coarse labels. Also, since character and word information have their different advantages in CWS, in our latent variable model, we use hybrid information based on both character and word sequences.

2 A Latent Variable Segmenter

2.1 Discriminative Probabilistic Latent Variable Model

Given data with latent structures, the task is to learn a mapping between a sequence of observations $\mathbf{x} = x_1, x_2, \ldots, x_m$ and a sequence of labels $\mathbf{y} = y_1, y_2, \ldots, y_m$. Each y_j is a class label for the j'th character of an input sequence, and is a member of a set $\mathbf{Y}$ of possible class labels. For each sequence, the model also assumes a sequence of latent variables $\mathbf{h} = h_1, h_2, \ldots, h_m$, which is unobservable in training examples.

The DPLVM is defined as follows (Morency et al.,

[2]The system was also used in Gao et al. (2007), with an improved performance in CWS.

[3]In practice, one may add a few extra labels based on linguistic intuitions (Xue, 2003).

2007):

$$P(\mathbf{y}|\mathbf{x},\mathbf{\Theta}) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h},\mathbf{x},\mathbf{\Theta})P(\mathbf{h}|\mathbf{x},\mathbf{\Theta}), \quad (1)$$

where $\mathbf{\Theta}$ are the parameters of the model. DPLVMs can be seen as a natural extension of CRF models, and CRF models can be seen as a special case of DPLVMs that have only one latent variable for each label.

To make the training and inference efficient, the model is restricted to have disjoint sets of latent variables associated with each class label. Each h_j is a member in a set $\mathbf{H}_{y_j}$ of possible latent variables for the class label y_j. $\mathbf{H}$ is defined as the set of all possible latent variables, i.e., the union of all $\mathbf{H}_{y_j}$ sets. Since sequences which have any $\mathbf{h}_j \notin \mathbf{H}_{y_j}$ will by definition have $P(\mathbf{y}|\mathbf{x},\mathbf{\Theta}) = 0$, the model can be further defined[4] as:

$$P(\mathbf{y}|\mathbf{x},\mathbf{\Theta}) = \sum_{\mathbf{h}\in\mathbf{H}_{y_1}\times\ldots\times\mathbf{H}_{y_m}} P(\mathbf{h}|\mathbf{x},\mathbf{\Theta}), \quad (2)$$

where $P(\mathbf{h}|\mathbf{x},\mathbf{\Theta})$ is defined by the usual conditional random field formulation:

$$P(\mathbf{h}|\mathbf{x},\mathbf{\Theta}) = \frac{\exp \mathbf{\Theta}\cdot\mathbf{f}(\mathbf{h},\mathbf{x})}{\sum_{\forall\mathbf{h}} \exp \mathbf{\Theta}\cdot\mathbf{f}(\mathbf{h},\mathbf{x})}, \quad (3)$$

in which $\mathbf{f}(\mathbf{h},\mathbf{x})$ is a feature vector. Given a training set consisting of n labeled sequences, $(\mathbf{x}_i,\mathbf{y}_i)$, for $i = 1\ldots n$, parameter estimation is performed by optimizing the objective function,

$$L(\mathbf{\Theta}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{\Theta}) - R(\mathbf{\Theta}). \quad (4)$$

The first term of this equation is the conditional log-likelihood of the training data. The second term is a regularizer that is used for reducing overfitting in parameter estimation.

For decoding in the test stage, given a test sequence $\mathbf{x}$, we want to find the most probable label sequence, $\mathbf{y}^*$:

$$\mathbf{y}^* = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x},\mathbf{\Theta}^*). \quad (5)$$

For latent conditional models like DPLVMs, the best label path $\mathbf{y}^*$ cannot directly be produced by the Viterbi algorithm because of the incorporation of hidden states. In this paper, we use a technique based on A^* search and dynamic programming described in Sun and Tsujii (2009), for producing the most probable label sequence $\mathbf{y}^*$ on DPLVM.

In detail, an A^* search algorithm[5] (Hart et al., 1968) with a Viterbi heuristic function is adopted to produce top-n latent paths, $\mathbf{h}_1, \mathbf{h}_2, \ldots \mathbf{h}_n$. In addition, a forward-backward-style algorithm is used to compute the exact probabilities of their corresponding label paths, $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n$. The model then tries to determine the optimal label path based on the top-n statistics, without enumerating the remaining low-probability paths, which could be exponentially enormous.

The optimal label path $\mathbf{y}^*$ is ready when the following "exact-condition" is achieved:

$$P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta}) - (1 - \sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})) \geq 0, \quad (6)$$

where $\mathbf{y}_1$ is the most probable label sequence in current stage. It is straightforward to prove that $\mathbf{y}^* = \mathbf{y}_1$, and further search is unnecessary. This is because the remaining probability mass, $1 - \sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})$, cannot beat the current optimal label path in this case. For more details of the inference, refer to Sun and Tsujii (2009).

2.2 Hybrid Word/Character Information

We divide our main features into two types: character-based features and word-based features. The character-based features are indicator functions that fire when the latent variable label takes some value and some predicate of the input (at a certain position) corresponding to the label is satisfied. For each latent variable label h_i (the latent variable label at position i), we use the predicate templates as follows:

- Input characters/numbers/letters locating at positions $i-2$, $i-1$, i, $i+1$ and $i+2$
- The character/number/letter bigrams locating at positions $i-2, i-1$, i and $i+1$

[4] It means that Eq. 2 is from Eq. 1 with *additional* definition.

[5] A^* search and its variants, like beam-search, are widely used in statistical machine translation. Compared to other search techniques, an interesting point of A^* search is that it can produce top-n results one-by-one in an efficient manner.

- Whether x_j and x_{j+1} are identical, for $j = (i-2) \ldots (i+1)$
- Whether x_j and x_{j+2} are identical, for $j = (i-3) \ldots (i+1)$

The latter two feature templates are designed to detect character or word reduplication, a morphological phenomenon that can influence word segmentation in Chinese.

The word-based features are indicator functions that fire when the local character sequence matches a word or a word bigram. A dictionary containing word and bigram information was collected from the training data. For each latent variable label unigram h_i, we use the set of predicate template checking for word-based features:

- The identity of the string $x_j \ldots x_i$, if it matches a word A from the word-dictionary of training data, with the constraint $i-6 < j < i$; multiple features will be generated if there are multiple strings satisfying the condition.
- The identity of the string $x_i \ldots x_k$, if it matches a word A from the word-dictionary of training data, with the constraint $i < k < i+6$; multiple features could be generated.
- The identity of the word bigram ($x_j \ldots x_{i-1}$, $x_i \ldots x_k$), if it matches a word bigram in the bigram dictionary and satisfies the aforementioned constraints on j and k; multiple features could be generated.
- The identity of the word bigram ($x_j \ldots x_i$, $x_{i+1} \ldots x_k$), if it matches a word bigram in the bigram dictionary and satisfies the aforementioned constraints on j and k; multiple features could be generated.

All feature templates were instantiated with values that occur in positive training examples. We found that using low-frequency features that occur only a few times in the training set improves performance on the development set. We hence do not do any thresholding of the DPLVM features: we simply use all those generated features.

The aforementioned word based features can incorporate word information naturally. In addition, following Wang et al. (2006), we found using a very simple heuristic can further improve the segmentation quality slightly. More specifically, two operations, *merge* and *split*, are performed on the DPLVM/CRF outputs: if a bigram $A \_ B$ was not observed in the training data, but the merged one AB was, then $A \_ B$ will be simply merged into AB; on the other hand, if AB was not observed but $A \_ B$ appeared, then it will be split into $A \_ B$. We found this simple heuristic on word information slightly improved the performance (e.g., for the PKU corpus, +0.2% on the F-score).

3 Experiments

We used the data provided by the second International Chinese Word Segmentation Bakeoff to test our approaches described in the previous sections. The data contains three corpora from different sources: Microsoft Research Asia (MSR), City University of Hong Kong (CU), and Peking University (PKU).

Since the purpose of this work is to evaluate the proposed latent variable model, we did not use extra resources such as common surnames, lexicons, parts-of-speech, and semantics. For the generation of word-based features, we extracted a word list from the training data as the vocabulary.

Four metrics were used to evaluate segmentation results: recall (R, the percentage of gold standard output words that are correctly segmented by the decoder), precision (P, the percentage of words in the decoder output that are segmented correctly), balanced F-score (F) defined by $2PR/(P+R)$, recall of OOV words (R-oov). For more detailed information on the corpora and these metrics, refer to Emerson (2005).

3.1 Training the DPLVM Segmenter

We implemented DPLVMs in C++ and optimized the system to cope with large scale problems, in which the feature dimension is beyond millions. We employ the feature templates defined in Section 2.2, taking into account those 3,069,861 features for the MSR data, 2,634,384 features for the CU data, and 1,989,561 features for the PKU data.

As for numerical optimization, we performed gradient decent with the Limited-Memory BFGS

(L-BFGS)[6] optimization technique (Nocedal and Wright, 1999). L-BFGS is a second-order Quasi-Newton method that numerically estimates the curvature from previous gradients and updates. With no requirement on specialized Hessian approximation, L-BFGS can handle large-scale problems in an efficient manner.

Since the objective function of the DPLVM model is non-convex, we randomly initialized parameters for the training.[7] To reduce overfitting, we employed an L_2 Gaussian weight prior[8] (Chen and Rosenfeld, 1999). During training, we varied the L_2-regularization term (with values 10^k, k from -3 to 3), and finally set the value to 1. We use 4 hidden variables per label for this task, compromising between accuracy and efficiency.

3.2 Comparison on Convergence Speed

First, we show a comparison of the convergence speed between the objective function of DPLVMs and CRFs. We apply the L-BFGS optimization algorithm to optimize the objective function of DPLVM and CRF models, making a comparison between them. We find that the number of iterations required for the convergence of DPLVMs are fewer than for CRFs. Figure 1 illustrates the convergence-speed comparison on the MSR data. The DPLVM model arrives at the plateau of convergence in around 300 iterations, with the penalized loss of $95K$ when $\#passes = 300$; while CRFs require 900 iterations, with the penalized loss of $98K$ when $\#passes = 900$.

However, we should note that the time cost of the DPLVM model in each iteration is around four times higher than the CRF model, because of the incorporation of hidden variables. In order to speed up the training speed of the DPLVM model in the future, one solution is to use the stochastic learning technique[9]. Another solution is to use a distributed version of L-BFGS to parallelize the batch training.

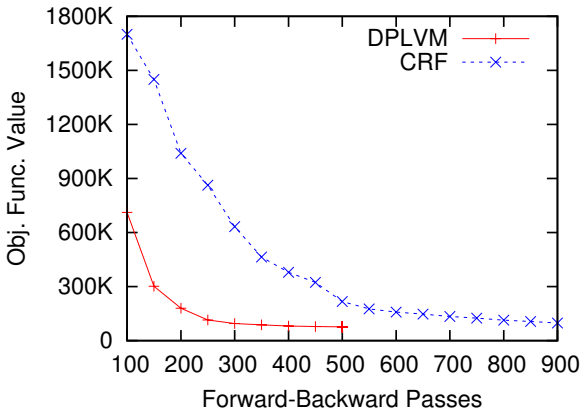

Figure 1: The value of the penalized loss based on the number of iterations: DPLVMs vs. CRFs on the MSR data.

	Style	#W.T.	#Word	#C.T.	#Char
MSR	S.C.	88K	2,368K	5K	4,050K
CU	T.C.	69K	1,455K	5K	2,403K
PKU	S.C.	55K	1,109K	5K	1,826K

Table 1: Details of the corpora. *W.T.* represents *word types*; *C.T.* represents *character types*; *S.C.* represents *simplified Chinese*; *T.C.* represents *traditional Chinese*.

4 Results and Discussion

Since the CRF model is one of the most successful models in Chinese word segmentation, we compared DPLVMs with CRFs. We tried to make experimental results comparable between DPLVMs and CRF models, and have therefore employed the same feature set, optimizer and fine-tuning strategy between the two. We also compared DPLVMs with semi-CRFs and other successful systems reported in previous work.

4.1 Evaluation Results

Three training and test corpora were used in the test, including the MSR Corpus, the CU Corpus, and the

[6]For numerical optimization on latent variable models, we also experimented the conjugate-gradient (CG) optimization algorithm and stochastic gradient decent algorithm (SGD). We found the L-BFGS with L_2 Gaussian regularization performs slightly better than the CG and the SGD. Therefore, we adopt the L-BFGS optimizer in this study.

[7]For a non-convex objective function, different parameter initializations normally bring different optimization results. Therefore, to approach closer to the global optimal point, it is recommended to perform multiple experiments on DPLVMs with random initialization and then select a good start point.

[8]We also tested the L-BFGS with L_1 regularization, and we found the L-BFGS with L_2 regularization performs better in this task.

[9]We have tried stochastic gradient decent, as described previously. It is possible to try other stochastic learning methods, e.g., stochastic meta decent (Vishwanathan et al., 2006).

MSR data	P	R	F	R-oov
DPLVM (*)	97.3	97.3	**97.3**	**72.2**
CRF (*)	97.1	96.8	97.0	72.0
semi-CRF (A06)	N/A	N/A	96.8	N/A
semi-CRF (G07)	N/A	N/A	97.2	N/A
CRF (Z06-a)	96.5	96.3	96.4	71.4
Z06-b	97.2	96.9	97.1	71.2
ZC07	N/A	N/A	97.2	N/A
Best05 (T05)	96.2	96.6	96.4	71.7
CU data	P	R	F	R-oov
DPLVM (*)	94.7	94.4	94.6	68.8
CRF (*)	94.3	93.9	94.1	65.8
CRF (Z06-a)	95.0	94.2	94.6	73.6
Z06-b	95.2	94.9	**95.1**	**74.1**
ZC07	N/A	N/A	**95.1**	N/A
Best05 (T05)	94.1	94.6	94.3	69.8
PKU data	P	R	F	R-oov
DPLVM (*)	95.6	94.8	**95.2**	**77.8**
CRF (*)	95.2	94.2	94.7	76.8
CRF (Z06-a)	94.3	94.6	94.5	75.4
Z06-b	94.7	95.5	95.1	74.8
ZC07	N/A	N/A	94.5	N/A
Best05 (C05)	95.3	94.6	95.0	63.6

Table 2: Results from DPLVMs, CRFs, semi-CRFs, and other systems.

PKU Corpus (see Table 1 for details). The results are shown in Table 2. The results are grouped into three sub-tables according to different corpora. Each row represents a CWS model. For each group, the rows marked by $*$ represent our models with hybrid word/character information. *Best05* represents the best system of the Second International Chinese Word Segmentation Bakeoff on the corresponding data; *A06* represents the semi-CRF model in Andrew (2006)[10], which was also used in Gao et al. (2007) (denoted as *G07*) with an improved performance; *Z06-a* and *Z06-b* represents the pure subword CRF model and the confidence-based combination of CRF and rule-based models, respectively (Zhang et al., 2006); *ZC07* represents the word-based perceptron model in Zhang and Clark (2007); *T05* represents the CRF model in Tseng et al. (2005); *C05* represents the system in Chen et al. (2005). The best F-score and recall of OOV words of each group is shown in bold.

As is shown in the table, we achieved the best F-score in two out of the three corpora. We also achieved the best recall rate of OOV words on those two corpora. Both of the MSR and PKU Corpus use simplified Chinese, while the CU Corpus uses the traditional Chinese.

On the MSR Corpus, the DPLVM model reduced more than 10% error rate over the CRF model using exactly the same feature set. We also compared our DPLVM model with the semi-CRF models in Andrew (2006) and Gao et al. (2007), and demonstrate that the DPLVM model achieved slightly better performance than the semi-CRF models. Andrew (2006) and Gao et al. (2007) only reported the results on the MSR Corpus.

In summary, tests for the Second International Chinese Word Segmentation Bakeoff showed competitive results for our method compared with the best results in the literature. Our discriminative latent variable models achieved the best F-scores on the MSR Corpus (97.3%) and PKU Corpus (95.2%); the latent variable models also achieved the best recalls of OOV words over those two corpora. We will analyze the results by varying the word-length in the following subsection.

4.2 Effect on Long Words

One motivation of using a latent variable model for CWS is to use latent variables to more adequately learn long range dependencies, as we argued in Section 1. In the test data of the MSR Corpus, 19% of the words are longer than 3 characters; there are also 8% in the CU Corpus and 11% in the PKU Corpus, respectively. In the MSR Corpus, there are some extremely long words ($Length > 10$), while the CU and PKU corpus do not contain such extreme cases.

Figure 2 shows the recall rate on different groups of words categorized by their lengths (the number of characters). As we expected, the DPLVM model performs much better on long words ($Length \geq 4$) than the CRF model, which used exactly the same feature set. Compared with the CRF model, the DPLVM model exhibited almost the same level of performance on short words. Both models have the best performance on segmenting the words with the length of two. The performance of the CRF

[10]It is a hybrid Markov/semi-Markov CRF model which outperforms conventional semi-CRF models (Andrew, 2006). However, in general, as discussed in Andrew (2006), it is essentially still a semi-CRF model.

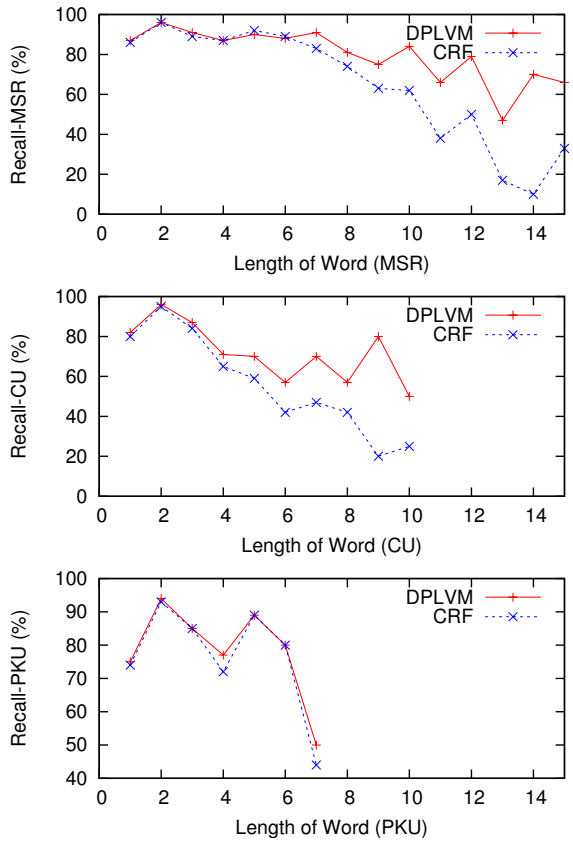

Figure 2: The recall rate on words grouped by the length.

model deteriorates rapidly as the word length increases, which demonstrated the difficulty on modeling long range dependencies in CWS. Compared with the CRF model, the DPLVM model performed quite well in dealing with long words, without sacrificing the performance on short words. All in all, we conclude that the improvement of using the DPLVM model came from the improvement on modeling long range dependencies in CWS.

4.3 Error Analysis

Table 3 lists the major errors collected from the latent variable segmenter. We examined the collected errors and found that many of them can be grouped into four types: over-generalization (the top row), errors on named entities (the following three rows), errors on idioms (the following three rows) and errors from inconsistency (the two rows at the bottom).

Our system performed reasonably well on very complex OOV words, such as 中国农业银行石家庄分行第二营业部 (Agricultural Bank of China, Shijiazhuang-city Branch, the second sales department) and 国家科委中国科技信息研究所 (Science and Technology Commission of China, National Institution on Scientific Information Analysis). However, it sometimes over-generalized to long words. For example, as shown in the top row, 国家环保局 (National Department of Environmental Protection) and 中宣部 (The Central Propaganda Department) are two organization names, but they are incorrectly merged into a single word.

Gold Segmentation	Segmenter Output
国家环保局//中宣部	国家环保局中宣部
Co-allocated org. names	
陈耀 (Chen Yao)	陈//耀
陈飞 (Chen Fei)	陈//飞
瓦西里斯 (Vasillis)	瓦//西里斯
通宵达旦	通宵//达旦
好高骛远	好//高骛远
一蹶不振	一//蹶//不振
Idioms	
宣传//家 (propagandist)	宣传家
沙漠化 (desertification)	沙漠//化

Table 3: Error analysis on the latent variable segmenter. The errors are grouped into four types: over-generalization, errors on named entities, errors on idioms and errors from data-inconsistency.

As for the following three rows, 陈耀 (Chen Yao) and 陈飞 (Chen Fei) are person names. They are wrongly segmented because we lack the features to capture the information of person names (such useful knowledge, e.g., common surname list, are currently not used in our system). In the future, such errors may be solved by integrating open resources into our system. 瓦西里斯 (Vasillis) is a transliterated foreign location name and is also wrongly segmented.

For the corpora that considered 4 character idioms as a word, our system successfully combined most of new idioms together. This differs greatly from the results of CRFs. However, there are still a number of new idioms that failed to be correctly segmented, as listed from the fifth row to the seventh row.

Finally, some errors are due to inconsistencies in the gold segmentation. For example, 宣传//家 (propagandist) is two words, but a word with similar

structure, 理论家 (theorist), is one word. 沙漠化 (desertification) is one word, but its synonym, 荒漠//化 (desertification), is two words in the gold segmentation.

5 Conclusion and Future Work

We presented a latent variable model for Chinese word segmentation, which used hybrid information based on both word and character sequences. We discussed that word and character information have different advantages, and could be complementary to each other. Our model is an alternative to the existing word based models and character based models.

We argued that using latent variables can better capture long range dependencies. We performed experiments and demonstrated that our model can indeed improve the segmentation accuracy on long words. With this improvement, tests on the data of the Second International Chinese Word Segmentation Bakeoff show that our system is competitive with the best in the literature.

Since the latent variable model allows a wide range of features, so the future work will consider how to integrate open resources into our system. The latent variable model handles latent-dependencies naturally, and can be easily extended to other labeling tasks.

Acknowledgments

We thank Kun Yu, Galen Andrew and Xiaojun Lin for the enlightening discussions. We also thank the anonymous reviewers who gave very helpful comments. This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan).

References

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation. *Proceedings of EMNLP'06*, pages 465–472.

Masayuki Asahara, Kenta Fukuoka, Ai Azuma, Chooi-Ling Goh, Yotaro Watanabe, Yuji Matsumoto, and Takahashi Tsuzuki. 2005. Combination of machine learning methods for optimum chinese word segmentation. *Proceedings of the fourth SIGHAN workshop*, pages 134–137.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. *Technical Report CMU-CS-99-108, CMU*.

Aitao Chen, Yiping Zhou, Anne Zhang, and Gordon Sun. 2005. Unigram language model for chinese word segmentation. *Proceedings of the fourth SIGHAN workshop*.

Hal Daumé III and Daniel Marcu. 2005. Learning as search optimization: approximate large margin methods for structured prediction. *Proceedings of ICML'05*.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. *Proceedings of the fourth SIGHAN workshop*, pages 123–133.

Jianfeng Gao, Galen Andrew, Mark Johnson, and Kristina Toutanova. 2007. A comparative study of parameter estimation methods for statistical natural language processing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, pages 824–831.

P.E. Hart, N.J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost path. *IEEE Trans. On System Science and Cybernetics*, SSC-4(2):100–107.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML'01*, pages 282–289.

Percy Liang. 2005. Semi-supervised learning for natural language. *Master's thesis, Massachusetts Institute of Technology*.

Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. 2007. Latent-dynamic discriminative models for continuous gesture recognition. *Proceedings of CVPR'07*, pages 1–8.

Jorge Nocedal and Stephen J. Wright. 1999. Numerical optimization. *Springer*.

F. Peng and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. *Proceedings of COLING'04*.

Slav Petrov and Dan Klein. 2008. Discriminative log-linear grammars with latent variables. *Proceedings of NIPS'08*.

Sunita Sarawagi and William Cohen. 2004. Semi-markov conditional random fields for information extraction. *Proceedings of ICML'04*.

Xu Sun and Jun'ichi Tsujii. 2009. Sequential labeling with latent variables: An exact inference algorithm and its efficient approximation. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL'09)*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff

2005. *Proceedings of the fourth SIGHAN workshop*, pages 168–171.

S.V.N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. 2006. Accelerated training of conditional random fields with stochastic meta-descent. *Proceedings of ICML'06*, pages 969–976.

Xinhao Wang, Xiaojun Lin, Dianhai Yu, Hao Tian, and Xihong Wu. 2006. Chinese word segmentation with maximum entropy and n-gram language model. In *Proceedings of the fifth SIGHAN workshop*, pages 138–141, July.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, 8(1).

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. *Proceedings of ACL'07*.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging by conditional random fields for chinese word segmentation. *Proceedings of HLT/NAACL'06 companion volume short papers*.