Information Processing and Management 49 (2013) 626–636

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Probabilistic Chinese word segmentation with non-local information and stochastic training

Xu Sun [a,b,*], Yaozhong Zhang [c], Takuya Matsuzaki [d], Yoshimasa Tsuruoka [e], Jun'ichi Tsujii [f]

[a] Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, Beijing, China
[b] School of EECS, Peking University, Beijing, China
[c] Dept of Computer Science, The University of Tokyo, Tokyo, Japan
[d] National Institute of Informatics, Tokyo, Japan
[e] Dept of EEIS, The University of Tokyo, Tokyo, Japan
[f] Microsoft Research Asia, Haidian District, Beijing, China

ARTICLE INFO

Article history:
Received 29 May 2011
Received in revised form 15 October 2012
Accepted 10 December 2012

Keywords:
Word segmentation
Natural language processing
Conditional random fields
Latent conditional random fields
Online training

ABSTRACT

In this article, we focus on Chinese word segmentation by systematically incorporating non-local information based on latent variables and word-level features. Differing from previous work which captures non-local information by using semi-Markov models, we propose an alternative method for modeling non-local information: a latent variable word segmenter employing word-level features. In order to reduce computational complexity of learning non-local information, we further present an improved online training method, which can arrive the same objective optimum with a significantly accelerated training speed. We find that the proposed method can help the learning of long range dependencies and improve the segmentation quality of long words (for example, complicated named entities). Experimental results demonstrate that the proposed method is effective. With this improvement, evaluations on the data of the second SIGHAN CWS bakeoff show that our system is competitive with the state-of-the-art systems.

1. Introduction

In 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.

1.1. Ambiguities and long words

Chinese character sequences are often ambiguous in their segmentation, 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 ($|W| \geqslant 4$) 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. Named entities are hard to segment. First, named entities are typically long character strings. In some criteria of word segmentation (e.g., MSR standard), a long named entity is supposed to be a single word, and in this

* Corresponding author at: Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, Beijing, China.
E-mail address: xusun@pku.edu.cn (X. Sun).

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

Computer Committee of International Federation of Automatic Control
国际自动控制联合会计算机委员会

Fig. 1. An example of a complex single "word".

setting we should not segment a long named entity into small segments. A typical error in word segmentation is that the long character strings of named entities are incorrectly segmented into small segments. Second, named entities may consist of many lexicon words, which are already registered in dictionaries. Such lexicon words can frequently appear in other sentences, not necessarily with named entities. Hence, it is probable that a named entity will be incorrectly segmented into multiple lexicon words.

For example, the word shown in Fig. 1 is one of the organization names in the Microsoft Research corpus. Its length is 13 and it contains more than six individual words, but it should be treated as a single word. Proper recognition of long OOV words is important 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.

1.2. Existing methods

Conventional approaches to Chinese word segmentation treat the problem as a character-based labeling task (Xue, 2003; Zhao, Huang, Li, & Lu, 2010; Zhao & Kit, 2011). 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 has been used for this task is the conditional random fields (CRFs) (Lafferty, McCallum, & Pereira, 2001), starting with the work of Peng, Feng, and McCallum (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 (Asahara et al., 2005; Tseng, Chang, & Andrew, 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 & Cohen, 2004). Despite the theoretical advantage of semi-CRFs over CRFs, however, some previous studies (Andrew, 2006; Liang, 2005) exploring the use of semi-CRFs for Chinese word segmentation did not find significant gains over the standard CRF models. 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 can 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. This type of features is 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.

1.3. Proposals

In this paper, instead of using semi-Markov models, we describe an alternative based on latent variables and word-level features to learn long range dependencies in Chinese word segmentation. We use the latent conditional random fields (LDCRFs) (Morency, Quattoni, & Darrell, 2007; Petrov & 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 few: $Label(y) \in \{B, C\}$, where B signifies *beginning a word* and C signifies *the continuation of a word*.[2] For example, by using LDCRFs, 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 the system can model the long range dependencies which may not be captured by those very simple labels.

Further, we use word-level features to model non-local information. Compared with the traditional character-based features, the word-based features can learn non-local information better. The word-based features are extracted from the training data only, and there is no need to use extra resources like lexicons. Since character and word information have their different advantages in word segmentation, we use both character-based features and word-based features.

Not surprisingly, learning non-local information increases computational complexity in the training. The use of latent variables can increase the inference lattice and slow down the training speed. The use of non-local features can increase the feature dimension and further slow down the training speed. Traditional batch training methods are very slow on training LDCRFs (Petrov & Klein, 2008; Sun, Matsuzaki, Okanohara, & Tsujii, 2009a). For example, Petrov and Klein (2008) mentioned both time and memory cost problems on training LDCRFs for natural language parsing. Also, Sun et al. (2009a) showed that the training of LDCRFs is computationally expensive on large scale problems. To accelerate training speed, we further present an improved online training method for optimizing LDCRFs. We demonstrate that the improved online training

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

method can arrive the similar objective optimum with significantly accelerated training speed. It is interesting to study online optimization methods for non-convex log-linear models like LDCRFs, because existing studies on this topic is limited.

2. Models

2.1. Conditional random fields

Conditional random fields (CRFs) are proposed as an alternative solution to structured classification by solving the "label bias problem" (Lafferty et al., 2001). Assuming a feature function that maps a pair of observation sequence $\mathbf{x}$ and label sequence $\mathbf{y}$ to a global feature vector $\mathbf{f}$, the probability of a label sequence $\mathbf{y}$ conditioned on the observation sequence $\mathbf{x}$ is modeled as follows (Lafferty et al., 2001):

$$P(\mathbf{y}|\mathbf{x},\mathbf{w}) = \frac{\exp[\mathbf{w}^\top \mathbf{f}(\mathbf{y},\mathbf{x})]}{\sum_{\forall \mathbf{y}'} \exp[\mathbf{w}^\top \mathbf{f}(\mathbf{y}',\mathbf{x})]}, \tag{1}$$

where $\mathbf{w}$ is a parameter vector.

Typically, computing $\sum_{\forall \mathbf{y}'} \exp[\mathbf{w}^\top \mathbf{f}(\mathbf{y}',\mathbf{x})]$ is computationally intractable. This summation can be computed using dynamic programming techniques (Lafferty et al., 2001). To make the dynamic programming techniques applicable, the dependencies of labels are chosen to obey the Markov property. This has a computational complexity of $O(NK^M)$. N is the length of the sequence; K is the cardinality of the label set; M is the length of the Markov order used by local features.

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 maximizing the objective function,

$$L(\mathbf{w}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w}) - R(\mathbf{w}). \tag{2}$$

The first term of this equation represents a conditional log-likelihood of the training data. The second term is a regularizer for reducing overfitting. Since our preliminary experiments suggested that an L_2 prior performed slightly better than an L_1 prior in the tasks discussed in this paper, we employed an L_2 prior, $R(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2\sigma^2}$. In what follows, we denote the conditional log-likelihood of each sample, log $P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w})$, as $\ell(i,\mathbf{w})$. The final objective function is as follows:

$$L(\mathbf{w}) = \sum_{i=1}^{n} \ell(i,\mathbf{w}) - \frac{\|\mathbf{w}\|^2}{2\sigma^2}. \tag{3}$$

2.2. Latent conditional random fields

Given the training data, 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 token of a word sequence, and is a member of a set $\mathbb{Y}$ 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 LDCRF model is defined as follows (Morency et al., 2007):

$$P(\mathbf{y}|\mathbf{x},\mathbf{w}) \triangleq \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h},\mathbf{x},\mathbf{w}) P(\mathbf{h}|\mathbf{x},\mathbf{w}),$$

where $\mathbf{w}$ represents the parameter vector of the model. To make the training efficient, a restriction is made for the model: for each label, the latent variables associated with it have no intersection with the latent variables from other labels (Morency et al., 2007; Petrov & Klein, 2008). This simplification is also a popular practice in other latent conditional models, including hidden-state conditional random fields (HCRFs) (Quattoni, Wang, Morency, Collins, & Darrell, 2007). Each h is a member in a set $\mathbb{H}(y)$ of possible latent variables for the class label y, and $\mathbb{H}(y^j) \cap \mathbb{H}(y^k) = \varnothing$ if $y^j \neq y^k$. $\mathbb{H}$ is defined as the set of all possible latent variables; i.e., the union of all $\mathbb{H}(y)$ sets: $\mathbb{H} = \cup_{y \in \mathbb{Y}} \mathbb{H}(y)$. In other words, h can have any value from $\mathbb{H}$, but $P(y|h)$ is zero except for only one of y in $\mathbb{Y}$. The disjoint restriction indicates a discrete simplification of $P(\mathbf{y}|\mathbf{h},\mathbf{x},\mathbf{w})$:

$$\begin{aligned} P(\mathbf{y}|\mathbf{h},\mathbf{x},\mathbf{w}) &= 1 \quad \textit{if} \quad \mathbf{h} \in \mathbb{H}(y_1) \times \ldots \times \mathbb{H}(y_m) \\ P(\mathbf{y}|\mathbf{h},\mathbf{x},\mathbf{w}) &= 0 \quad \textit{if} \quad \mathbf{h} \notin \mathbb{H}(y_1) \times \ldots \times \mathbb{H}(y_m) \end{aligned}$$

where m is the length of the labeling: $m = |\mathbf{y}|$. The formula $\mathbf{h} \in \mathbb{H}(y_1) \times \ldots \times \mathbb{H}(y_m)$ indicates that the latent-labeling $\mathbf{h}$ is a latent-labeling of the labeling $\mathbf{y}$, which can be more formally defined as follows:

$$\mathbf{h} \in \mathbb{H}(y_1) \times \ldots \times \mathbb{H}(y_m) \Longleftrightarrow h_j \in \mathbb{H}(y_j),\ j = 1, \ldots, m.$$

Since sequences that have any $h_j \notin \mathbb{H}(y_j)$ will by definition have $P(\mathbf{y}|h_j,\mathbf{x},\mathbf{w}) = 0$, the model can be simplified as:

$$P(\mathbf{y}|\mathbf{x},\mathbf{w}) \triangleq \sum_{\mathbf{h} \in \mathbb{H}(y_1) \times \ldots \times \mathbb{H}(y_m)} P(\mathbf{h}|\mathbf{x},\mathbf{w}). \tag{4}$$

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

$$P(\mathbf{h}|\mathbf{x},\mathbf{w}) = \frac{\exp[\mathbf{w}^\top \mathbf{f}(\mathbf{h},\mathbf{x})]}{\sum_{\mathbf{h}' \in \mathbb{H}\times\ldots\times\mathbb{H}} \exp[\mathbf{w}^\top \mathbf{f}(\mathbf{h}',\mathbf{x})]}, \tag{5}$$

where $\mathbf{f}(\mathbf{h},\mathbf{x})$ is a global 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

$$L(\mathbf{w}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w}) - R(\mathbf{w}) = \sum_{i=1}^{n} \log \left\{ \sum_{\mathbf{h}\in\mathbb{H}(y_1^i)\times\ldots\times\mathbb{H}(y_m^i)} P(\mathbf{h}|\mathbf{x}_i,\mathbf{w}) \right\} - R(\mathbf{w}). \tag{6}$$

We denote $\log P(\mathbf{y}_i|\mathbf{x}_i,\mathbf{w})$, as $\ell(i,\mathbf{w})$ and we used L_2 regularization. Then, the objective function will be like this:

$$L(\mathbf{w}) = \sum_{i=1}^{n} \ell(i,\mathbf{w}) - \frac{\|\mathbf{w}\|^2}{2\sigma^2}.$$

For details of the decoding on LDCRFs, refer to Sun, Morency, Okanohara, and Tsujii (2008) and Sun and Tsujii (2009). Recent related work on latent conditional models also includes (Sun, Okazaki, & Tsujii, 2009b).

2.3. Improved SGD training (I-SGD)

Training LDCRFs is challenging. Traditional gradient-based batch training methods, such as Limited-memory BFGS (LBFGS) (Nocedal & Wright, 1999), are quite slow on training LDCRFs (Petrov & Klein, 2008; Sun et al., 2009a). To accelerate training speed, we employ online optimization methods for training LDCRFs. First, we review the literature on stochastic gradient descent (SGD).

2.3.1. Stochastic gradient descent

The SGD uses a small randomly-selected subset of the training samples to approximate the gradient of the objective function given by Eq. (3). The number of training samples used for this approximation is called the batch size. The extreme case is a batch size of 1, and it gives the maximum frequency of updates, which we adopt in this work. Then, the model parameters are updated in such a way:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \gamma_k \nabla_{\mathbf{w}_k} \left\{ \ell(i,\mathbf{w}_k) - \frac{\|\mathbf{w}_k\|^2}{2n\sigma^2} \right\}, \tag{7}$$

where k is the update counter, γ_k is the learning rate, and n is the number of data samples in the training data set.

2.3.2. Modifications

In preliminary experiments, we find the standard SGD training is still slow in this task. The reason of the slowness is from the setting of the learning rates. The setting of learning rates often has a big impact on the convergence speed of the SGD training. A typical choice of learning rate is as follows (Collins, Globerson, Koo, Carreras, & Bartlett, 2008):

$$\gamma_k = \frac{\gamma_0}{1 + k/n}, \tag{8}$$

where γ_0 is a constant. Although this scheduling guarantees theoretical ultimate convergence, the actual convergence speed can be slow in practice. For the CWS task, we want a more effective scheduling of learning rates to achieve faster convergence speed of the SGD training. We tested a simple exponential decay (Tsuruoka, Tsujii, & Ananiadou, 2009):

$$\gamma_k = \gamma_0 \alpha^{k/n}, \tag{9}$$

where α is constant, with $0 < \alpha < 1$. Our experiments demonstrate that this exponential decay is more effective than the traditional decay (Eq. (8)), and arrives empirical convergence state faster.

The main reason is that the exponential decay is more "smooth" than the traditional decay. To achieve fast empirical convergence, the online training should have a "smooth" decay on the learning rate. The traditional decay schedule drops too quickly at the beginning and too slowly at the end. First, since the traditional decay drops too quickly at the beginning stage, the model weights are not adequately trained in the beginning stage and it is far away from convergence. Second, since the traditional decay drops too slowly at the end, the learning rate is not small enough at the ending stage, so that the model weights can meet severe fluctuations (i.e., the weights are unstable) from one sample to another sample in the online setting. On the other hand, the exponential decay can partially solve those two issues and will have faster empirical convergence.

There is no prior work on convergence analysis for the SGD training with the exponential decaying rate Eq. (9). We can show that the SGD training with the exponential decaying rate has reasonable convergence properties. That is, the SGD training with the exponential decaying rate Eq. (9) is asymptotically convergent. This convergence analysis statement can be derived from the assumptions and convergence analysis discussed in Sun, Wang, and Li (2012). The SGD training with the exponential decaying rate Eq. (9) converges to a fixed point, but different runs of the algorithm may converge to different fixed points if use random shuffling of the order of the training samples.

When the update gradients are sparse and the overall feature dimension is high, it will be wasteful to use the standard SGD updates, because at each update we need to apply regularization to all features, including those features that are not used in the current training sample. To make training faster, we can use a traditional lazy regularization schedule (Shalev-Shwartz, Singer, & Srebro, 2007) for the SGD. Nevertheless, it is interesting that we find a very simple heuristic lazy regularization schedule (called *simple lazy regularization* below) has as good empirical performance as the traditional lazy regularization. Compared with the traditional lazy regularization schedule, the simple lazy regularization is easier to implement and has moderately faster training speed. Hence, we use the simple lazy regularization instead of the traditional lazy regularization. Fig. 2 shows the I-SGD training method with new decaying rate and the simple lazy regularization.

3. Hybrid word and character features

To better capture non-local information, we employ features based on words and word bigrams. To derive word features, our system first automatically collects a list of word unigrams and bigrams from the training data. This list of word unigrams and bigrams are then used as a unigram-dictionary and a bigram-dictionary to generate word-based *unigram* and *bigram* features. The word-based features are indicator functions that fire when the local character sequence matches a word unigram or bigram occurred in the training data. For each label at position i, we use the predicate templates as follows:

- `unigram1`$(\mathbf{x}, y_i) \leftarrow [x_{j,i}, y_i]$, if the character sequence $x_{j,i}$ matches a word $w \in \mathbb{U}$, with the constraint $i - 6 < j < i$. The item $x_{j,i}$ represents the character sequence $x_j \ldots x_i$. $\mathbb{U}$ represents the unigram word dictionary compiled from the training data.
- `unigram2`$(\mathbf{x}, y_i) \leftarrow [x_{i,k}, y_i]$, if the character sequence $x_{i,k}$ matches a word $w \in \mathbb{U}$, with the constraint $i < k < i + 6$.
- `bigram1`$(\mathbf{x}, y_i) \leftarrow [x_{j,i-1}, x_{i,k}, y_i]$, if the word bigram candidate $[x_{j,i-1}, x_{i,k}]$ hits a word bigram $[w_i, w_j] \in \mathbb{B}$, and satisfies the aforementioned constraints on j and k. $\mathbb{B}$ represents the word bigram dictionary collected from the training data.
- `bigram2`$(\mathbf{x}, y_i) \leftarrow [x_{j,i}, x_{i+1,k}, y_i]$, if the word bigram candidate $[x_{j,i}, x_{i+1,k}]$ hits a word bigram $[w_i,\ w_j] \in \mathbb{B}$, and satisfies the aforementioned constraints on j and k.

Fig. 3 presents a brief piece of pseudocode for extracting word-based unigram and bigram features.

We also use traditional character-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 label at position i, we use the predicate templates as follows:

- Character unigrams locating at positions $i - 2$, $i - 1$, i, $i + 1$ and $i + 2$.
- Character bigrams locating at positions $i - 2$, $i - 1$, i and $i + 1$.
- 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 $j = i - 3, \ldots, i + 1$.

All feature templates were instantiated with values that occur in positive training examples. The aforementioned word based features can incorporate non-local information naturally.

In addition, we found that using a very simple heuristic can slightly improve segmentation quality. Two operations, *merge* and *split*, are performed on the LDCRF/CRF outputs: if a bigram *A B* was not observed in the training data, but the merged one *AB* was, then *AB* will be simply merged into *AB*; on the other hand, if *AB* was not observed but *AB* appeared, then it will be split into *AB*. We found this simple heuristic on word information moderately improved the performance.

```
1: Procedure Improved SGD
2: for #iter ← 1 to Convergence
3:     for $k \leftarrow 1$ to $n$
4:         pick a sample $(\mathbf{x}_i,\ \mathbf{y}_i)$ randomly
5:         $\gamma \leftarrow \gamma_0 \alpha^{k/n}$
6:         $\mathbf{w} \leftarrow \mathbf{w} + \gamma \nabla_{\mathbf{w}} \ell(i, \mathbf{w})$
7:     $\mathbf{w} \leftarrow \mathbf{w} - \gamma \nabla_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2\sigma^2}$
8: Return w
```

Fig. 2. Improved SGD Training with exponential decaying learning rate and *simple lazy regularization*.

```
//word unigram features
for (int L = maxWordLength; L >= minWordLength; L--){
   string chars = sentence.getChars(i - L, i);
   if (trainWordSet.Contains(chars)){
      string feature = "word.left" + chars;
      featureSet.Add(feature);
      leftWordSet.Add(chars);
   }
}
for (int L = maxWordLength; L >= minWordLength; L--){
   string chars = sentence.getChars(i, L);
   if (trainWordSet.Contains(chars)){
      string feature = "word.right" + chars;
      featureSet.Add(feature);
      rightWordSet.Add(chars);
   }
}
//word bigram features
foreach (string lWord in leftWordSet)
   foreach (string rWord in rightWordSet){
      string bigram = lWord + " " + rWord;
      if (trainBigramSet.Contains(bigram)){
         string feature = "word.bigram" + bigram;
         featureSet.Add(feature);
      }
   }
```

Fig. 3. Pseudocode for extracting word-based unigram and bigram features.

Table 1
Details of the corpora. *W.T.* represents *word types*; *C.T.* represents *character types*.

	#W.T.	#Word	#C.T.	#Char
MSR	8.8×10^4	2.4×10^6	5×10^3	4.1×10^6
CU	6.9×10^4	1.5×10^6	5×10^3	2.4×10^6
PKU	5.5×10^4	1.1×10^6	5×10^3	1.8×10^6

4. Experiments

4.1. Data and metrics

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). The detailed properties of the there data sets are shown in Table 1. 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 list of word unigrams and bigrams from the training data.

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).

4.2. Experimental settings

Since the CRF model is one of the most successful models in Chinese word segmentation, we compare LDCRFs with CRFs. We make experimental results comparable between LDCRFs and CRF models, and have therefore employed the same feature set and optimizer. We employ the feature templates defined in Section 3, taking into account those 3.0×10^6 features for the MSR data, 2.6×10^6 features for the CU data, and 1.9×10^6 features for the PKU data.

As for numerical optimization, we adopt the Limited-Memory BFGS (L-BFGS) optimization technique (Nocedal & Wright, 1999), standard SGD training (SGD), and improved SGD training (I-SGD). The SGD and I-SGD training methods are online training methods, as have been introduced in previous sections.

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

	Rec-OOV	Pre	Rec	F-score
MSR data				
LDCRF ($*$)	72.2	97.3	97.3	97.3
CRF ($*$)	72.0	97.1	96.8	97.0
Zhao'10 Zhao et al. (2010)	N/A	N/A	N/A	97.35
Zhao'11 Zhao and Kit (2011)	N/A	N/A	N/A	97.58
Andrew'06 Andrew (2006)	N/A	N/A	N/A	96.8
Gao'07 Gao et al. (2007)	N/A	N/A	N/A	97.2
Zhang'06 Zhang et al. (2006)	71.4	96.5	96.3	96.4
Zhang'07 Zhang and Clark (2007)	N/A	N/A	N/A	97.2
Best05 Tseng et al. (2005)	71.7	96.2	96.6	96.4
CU data				
LDCRF ($*$)	68.8	94.7	94.4	94.6
CRF ($*$)	65.8	94.3	93.9	94.1
Zhao'10 Zhao et al. (2010)	N/A	N/A	N/A	94.76
Zhao'11 Zhao and Kit (2011)	N/A	N/A	N/A	96.10
Zhang'06 Zhang et al. (2006)	73.6	95.0	94.2	94.6
Zhang'07 Zhang and Clark (2007)	N/A	N/A	N/A	95.1
Best05 Tseng et al. (2005)	69.8	94.1	94.6	94.3
PKU data				
LDCRF ($*$)	77.8	95.6	94.8	95.2
CRF ($*$)	76.8	95.2	94.2	94.7
Zhao'10 Zhao et al. (2010)	N/A	N/A	N/A	95.15
Zhao'11 Zhao and Kit (2011)	N/A	N/A	N/A	95.40
Zhang'06 Zhang et al. (2006)	75.4	94.3	94.6	94.5
Zhang'07 Zhang and Clark (2007)	N/A	N/A	N/A	94.5
Best05 Chen et al. (2005)	63.6	95.3	94.6	95.0

Since the objective function of the LDCRF model is non-convex, we randomly initialized parameters for the training.[3] To reduce overfitting, we employed an L_2 Gaussian weight prior (Chen et al., 1999).

4.3. Segmentation quality

The results are shown in Table 2. The results are based on convergence of batch training. 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; *Andrew'06* represents the semi-CRF model in Andrew (2006), which was also used in Gao, Andrew, Johnson, and Toutanova (2007) (denoted as *Gao'07*) with improved performance; *Zhang'06* represents the sub-word based CRF model (Zhang, Kikui, & Sumita, 2006); *Zhang'07* represents the word-based perceptron model in Zhang and Clark (2007).

More recent development of Chinese word segmentation included the work in Zhao et al. (2010) and Zhao and Kit (2011). *Zhao'10* (Zhao et al., 2010) had similar accuracy level compared with our results. *Zhao'11* (Zhao & Kit, 2011) shared a similar idea with this work by trying to find non-local information to improve the performance of word segmentation, and achieved good performance. *Zhao'10* (Zhao et al., 2010) and *Zhao'11* (Zhao & Kit, 2011) used more complex labels/tags than our method. Their methods used six labels while our method used two labels. Using more labels will increase the segmentation accuracies, but its disadvantage is the slowing down of the training speed. The slowing down of the training speed has a quadratic relation with the number of labels.

On the MSR corpus, the LDCRF model reduced more than 10% error rate over the CRF model using exactly the same feature set. We also compared our LDCRF model with the semi-CRF models in Andrew (2006) and Gao et al. (2007), and the results suggest that the LDCRF 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.

We also performed cross validation on the training data sets to compare the performance of CRFs and LDCRFs. The experimental results are shown in Table 3. As we can see, the LDCRF segmentation method (without tuning on the weight initialization) already outperformed the CRF segmentation method on all of the three datasets with the cross validation setting. This confirmed that the LDCRF model, even not fully optimized, is better than the CRF model in word segmentation. With multiple random initialization for weight vectors and choosing a good one, the performance of LDCRF can be further improved.

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

Table 3
Comparisons between LDCRFs (without multiple random initialization on weights) and CRFs via 4-fold cross validation on the training sets.

	CV1	CV2	CV3	CV4	Overall F-score
MSR data					
LDCRF	95.4	95.1	95.5	95.8	**95.4**
CRF	94.9	94.8	95.0	95.4	95.0
CU data					
LDCRF	94.9	96.3	96.1	94.2	**95.4**
CRF	94.9	96.3	95.9	94.1	95.3
PKU data					
LDCRF	95.2	95.1	95.4	94.8	**95.1**
CRF	95.2	95.1	95.4	94.6	95.0

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.

4.4. Accelerated training

Here, we perform experiments on optimizing objective functions of CRFs and LDCRFs on different data sets. We compare the improved SGD (I-SGD) training method with the L-BFGS batch training method and the standard SGD training method. For the online training method, we need to set the value of γ_0 and α. Typically, the setting $\gamma_0 = 0.1$ and $\alpha = [0.85, 0.95]$ works well for online training.

The curves of the objective functions by varying training iterations are shown in Fig. 4. As we can see, in all cases compared with the L-BFGS batch training method, the I-SGD method achieved the same level of objective values on convergence. Most importantly, the I-SGD method converges much faster than the L-BFGS batch training method and the existing SGD training method.

It is noteworthy that the weights produced by the I-SGD and the L-BFGS training are similar when the two methods optimized the objective function to the similar optimum. Hence, in spite of the significant acceleration of training speed, word segmentation accuracies are similar when we use the I-SGD training instead of the L-BFGS training.

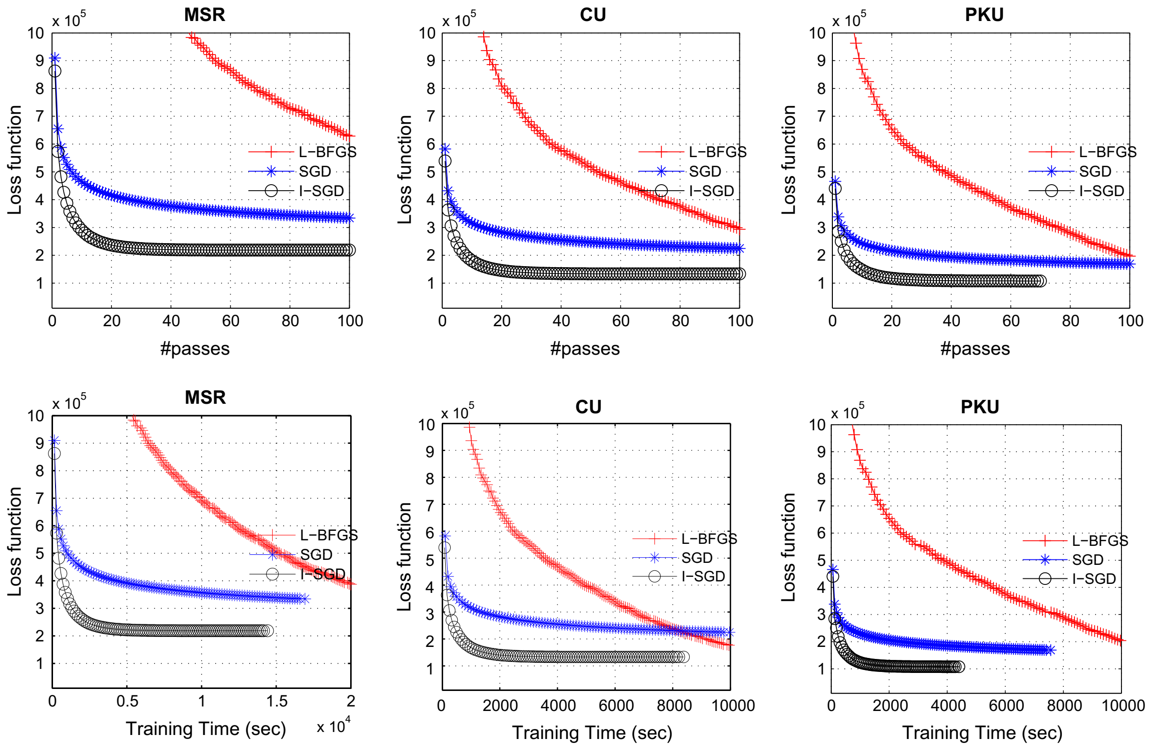


Fig. 4. Convergence speed of different training methods on LDCRFs.

4.5. Effect on long words

One motivation of modeling non-local information for CWS is to use non-local information to 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 three 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 (e.g., *Length* > 10), while the CU and PKU corpus do not contain such extreme cases.

Fig. 5 shows the recall rate on different groups of words categorized by their lengths (the number of characters). As we expected, the LDCRF model performs much better on long words (*Length* $\geqslant$ 4) than the CRF model, which used exactly the same feature set. Compared with the CRF model, the LDCRF 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 model deteriorates rapidly as the word length increases, which demonstrates the difficulty of modeling long range dependencies in CWS. Compared with the CRF model, the LDCRF model performed quite well in dealing with long words, without sacrificing the performance on short words. This indicates that the improvement of using the LDCRF model came from the improvement on modeling long range dependencies in CWS.

4.6. Error analysis

Fig. 6 lists the major types of 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).

Among the four types of errors, co-allocated organization names are difficult for segmentation. The example shown in Table 6 is a case that those two organization names are independent to each other, and therefore should be segmented. There are also many cases that two organizations names should be annotated into a single word. For example, when the two organization names have a relation of "*B* is a branch of *A*". We found the LDCRF model is not suitable for disambiguating such cases. Compared with CRFs, the LDCRF model is more likely to merge co-allocated names.

For the human names, they are wrongly segmented because we lack features to capture the information of person names. A collection of common surnames can be helpful, but such extra knowledge sources are currently not used in our system. Correctly segmenting transliterated location names is also difficult and it requires the technique on recognizing transliterated location names. In the future, such errors can be solved via using extra resources.

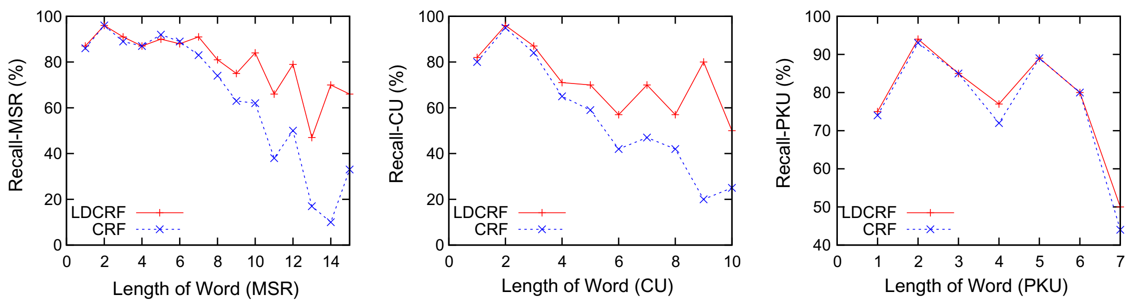
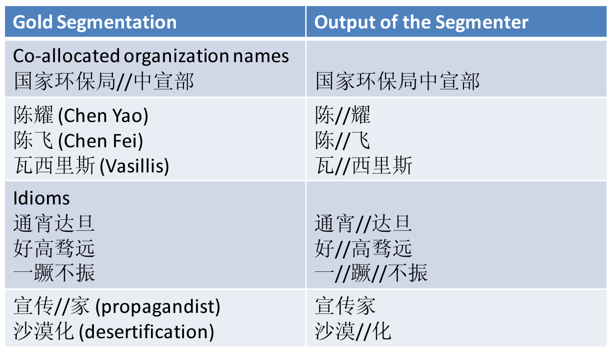


Fig. 5. The recall rate on words grouped by the length.



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

Fig. 6. 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.

Gold Segmentation (Training)	Gold Segmentation (Testing)
理论家 (theorist)	宣传//家 (propagandist)
荒漠//化 (desertification)	沙漠化 (desertification)

Fig. 7. Examples of annotation inconsistencies between training and evaluation data.

Our system can successfully recognize many new idioms consisting of four characters. Nevertheless, there are still a number of new idioms that failed to be correctly segmented, as listed in the table. Finally, some errors are due to inconsistencies of the data itself, i.e., the annotation errors in the training and testing data. The annotation errors are illustrated in Fig. 7.

5. Conclusions, discussion, and future work

We presented a non-local information based model for Chinese word segmentation. Non-local information are learned via latent variables and new features including word-level features. From experimental evaluation, we found that modeling non-local information can better capture long range dependencies and improve the word segmentation quality. Our system is competitive with the state-of-the-art methods.

On the other hand, we showed that learning non-local information can slow down the training speed. To solve this problem, we further presented an improved online training method for training the segmentation model with non-local information. We showed that the proposed online training method has significantly accelerated training speed.

There are different segmentation standards among different datasets. Even with different segmentation standards, long words are expected to be important. There are different kinds of long words. For example, some transliterated names, location names, human names, and Chinese idioms. Those long words are not *compound* words because they contain no subwords. In such cases, modeling non-local information is also meaningful.

Most recently, Sun, Wang, and Li (2012) presented a very accurate and fast online training method for Chinese word segmentation. Although the experiments in Sun et al. (2012) were based on CRFs, the proposed feature-frequency-adaptive online training method in Sun et al. (2012) is a general purpose algorithm and can be applied to different machine learning models and different tasks. As the future work, we will consider applying the feature-frequency-adaptive online training to latent conditional models with non-local features for word segmentation and other natural language processing tasks.

Acknowledgments

Part of this work was presented at *NAACL-HLT 2009* (Sun, Zhang, Matsuzaki, Tsuruoka, & Tsujii, 2009c). This work is supported by National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101) and National Natural Science Foundation of China (Nos. 91024009 and 60973053).

References

Andrew, G. (2006). A hybrid markov/semi-markov conditional random field for sequence segmentation. In *Proceedings of EMNLP'06* (pp. 465–472).
Asahara, M., Fukuoka, K., Azuma, A., Goh, C. -L., Watanabe, Y., Matsumoto, Y., et al. (2005). Combination of machine learning methods for optimum chinese word segmentation. In *Proceedings of the fourth SIGHAN workshop* (pp. 134–137).
Chen, A., Zhou, Y., Zhang, A., & Sun, G. (2005). Unigram language model for chinese word segmentation. In *Proceedings of the fourth SIGHAN workshop* (pp. 138–141).
Chen, Stanley F., & Rosenfeld, R. (1999). *A gaussian prior for smoothing maximum entropy models*. Technical Report CMU-CS-99-108, CMU.
Collins, M., Globerson, A., Koo, T., Carreras, X., & Bartlett, Peter L. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin markov networks. *Journal of Machine Learning Research (JMLR), 9*, 1775–1822.
Emerson, T. (2005). The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop* (pp. 123–133).
Gao, J., Andrew, G., Johnson, M., & Toutanova, K. (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)* (pp. 824–831).
Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th international conference on machine learning (ICML'01)* (pp. 282–289).
Liang, P. (2005). *Semi-supervised learning for natural language*. Master's thesis, Massachusetts Institute of Technology.
Morency, L. -P., Quattoni, A., & Darrell, T. (2007). Latent-dynamic discriminative models for continuous gesture recognition. In *Proceedings of CVPR'07* (pp. 1–8).
Nocedal, J., & Wright, Stephen J. (1999). *Numerical optimization*. Springer.
Peng, F., Feng, F., & McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of coling 2004* (pp. 562–568). Geneva, Switzerland; August 23–27.
Petrov, Slav., and Klein, Dan., 2008. Discriminative log-linear grammars with latent variables. In *Advances in neural information processing systems 20 (NIPSs)* (pp. 1153–1160).
Quattoni, A., Wang, S., Morency, L. -P., Collins, M., & Darrell, T. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(10), 1848–1852 (ISSN:0162-8828).
Sarawagi, S., & Cohen, W. (2004). Semi-markov conditional random fields for information extraction. In *Proceedings of NIPS'04*.
Shalev-Shwartz, S., Singer, Y., & Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of ICML'07*.
Sun, X., & Tsujii, J. (2009). Sequential labeling with latent variables: An exact inference algorithm and its efficient approximation. In *Proceedings of EACL'09* (pp. 772–780). Athens, Greece.
Sun, X., Morency, L. -P., Okanohara, D., & Tsujii, J. (2008). Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. In *Proceedings of COLING'08* (pp. 841–848). Manchester, UK.

Sun, X., Matsuzaki, T., Okanohara, D., & Tsujii, J. (2009a). Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st international joint conference on artificial intelligence (IJCAI 2009)* (pp. 1236–1242).

Sun, X., Okazaki, N., & Tsujii, J. (2009b). Robust approach to abbreviating terms: A discriminative latent variable model with global information. In *Proceedings of the ACL'09* (pp. 905–913). Suntec, Singapore.

Sun, X., Zhang, Y., Matsuzaki, T., Tsuruoka, Y., & Tsujii, J. (2009c). A discriminative latent variable chinese segmenter with hybrid word/character information. In *Proceedings of NAACL-HLT'09* (pp. 56–64). Boulder, Colorado.

Sun, X., Wang, H., & Li, W. (2012). Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of ACL'12* (pp. 253–262). The Association for Computational Linguistics.

Sun, X., Wang, H., & Li, W. (2012). Feature-frequency-adaptive online training for structured classification with high dimensional features. In *Submission*.

Tseng, H., Chang, P., Andrew, G. (2005). Daniel Jurafsky, and Christopher Manning. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop* (pp. 168–171).

Tsuruoka, Y., Tsujii, J., & Ananiadou, S. (2009). Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of ACL'09* (pp. 477–485). Suntec, Singapore.

Xue, Nianwen (2003). Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing, 8*(1), 29–48.

Zhang, R., Kikui, G., & Sumita, E. (2006). Subword-based tagging by conditional random fields for chinese word segmentation. In *Proceedings of the human language technology conference of the NAACL, companion volume: Short papers* (pp. 193–196). New York City, USA: Association for Computational Linguistics.

Zhang, Y., & Clark, S. (2007). Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 840–847). Prague, Czech Republic: Association for Computational Linguistics.

Zhao, Hai, & Kit, Chunyu (2011). Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Science, 181*(1), 163–183.

Zhao, Hai, Huang, Changning, Li, Mu, & Lu, Bao-Liang (2010). A unified character-based tagging framework for chinese word segmentation. *ACM Transactions on Asian Language Information Processing, 9*(2).