Fast Online Training with Frequency-Adaptive Learning Rates for Chinese Word Segmentation and New Word Detection

Xu Sun[†], **Houfeng Wang**[‡], **Wenjie Li**[†]
[†]Department of Computing, The Hong Kong Polytechnic University
[‡]Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, China
{csxsun, cswjli}@comp.polyu.edu.hk wanghf@pku.edu.cn

Abstract

We present a joint model for Chinese word segmentation and new word detection. We present high dimensional new features, including word-based features and enriched edge (label-transition) features, for the joint modeling. As we know, training a word segmentation system on large-scale datasets is already costly. In our case, adding high dimensional new features will further slow down the training speed. To solve this problem, we propose a new training method, adaptive online gradient descent based on feature frequency information, for very fast online training of the parameters, even given large-scale datasets with high dimensional features. Compared with existing training methods, our training method is an order magnitude faster in terms of training time, and can achieve equal or even higher accuracies. The proposed fast training method is a general purpose optimization method, and it is not limited in the specific task discussed in this paper.

1 Introduction

Since Chinese sentences are written as continuous sequences of characters, segmenting a character sequence into words is normally the first step in the pipeline of Chinese text processing. The major problem of Chinese word segmentation is the ambiguity. Chinese character sequences are normally ambiguous, and new words (out-of-vocabulary words) are a major source of the ambiguity. A typical category of new words is named entities, including organization names, person names, location names, and so on.

In this paper, we present high dimensional new features, including word-based features and enriched edge (label-transition) features, for the joint modeling of Chinese word segmentation (CWS) and new word detection (NWD). While most of the state-of-the-art CWS systems used semi-Markov conditional random fields or latent variable conditional random fields, we simply use a single first-order conditional random fields (CRFs) for the joint modeling. The semi-Markov CRFs and latent variable CRFs relax the Markov assumption of CRFs to express more complicated dependencies, and therefore to achieve higher disambiguation power. Alternatively, our plan is not to relax Markov assumption of CRFs, but to exploit more complicated dependencies via using refined high-dimensional features. The advantage of our choice is the simplicity of our model. As a result, our CWS model can be more efficient compared with the heavier systems, and with similar or even higher accuracy because of using refined features.

As we know, training a word segmentation system on large-scale datasets is already costly. In our case, adding high dimensional new features will further slow down the training speed. To solve this challenging problem, we propose a new training method, adaptive online gradient descent based on feature frequency information (ADF), for very fast word segmentation with new word detection, even given large-scale datasets with high dimensional features. In the proposed training method, we try to use more refined learning rates. Instead of using a single learning rate (a scalar) for all weights, we extend the learning rate scalar to a learning rate vector based on feature frequency information in the updating. By doing so, each weight has

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 253–262,
Jeju, Republic of Korea, 8-14 July 2012.

its own learning rate adapted on feature frequency information. We will show that this can significantly improve the convergence speed of online learning. We approximate the learning rate vector based on *feature frequency information in the updating process*. Our proposal is based on the intuition that a feature with higher frequency in the training process should be with a learning rate that is decayed faster. Based on this intuition, we will show the formalized training algorithm later. We will show in experiments that our solution is an order magnitude faster compared with exiting learning methods, and can achieve equal or even higher accuracies.

The contribution of this work is as follows:

- We propose a general purpose fast online training method, ADF. The proposed training method requires only a few passes to complete the training.
- We propose a joint model for Chinese word segmentation and new word detection.
- Compared with prior work, our system achieves better accuracies on both word segmentation and new word detection.

2 Related Work

First, we review related work on word segmentation and new word detection. Then, we review popular online training methods, in particular stochastic gradient descent (SGD).

2.1 Word Segmentation and New Word Detection

Conventional approaches to Chinese word segmentation treat the problem as a sequential labeling task (Xue, 2003; Peng et al., 2004; Tseng et al., 2005; Asahara et al., 2005; Zhao et al., 2010). To achieve high accuracy, most of the state-of-the-art systems are heavy probabilistic systems using semi-Markov assumptions or latent variables (Andrew, 2006; Sun et al., 2009b). For example, one of the state-of-the-art CWS system is the latent variable conditional random field (Sun et al., 2008; Sun and Tsujii, 2009) system presented in Sun et al. (2009b). It is a heavy probabilistic model and it is slow in training. A few other state-of-the-art CWS systems are using semi-Markov perceptron methods or voting systems based on multiple semi-Markov perceptron segmenters (Zhang and Clark, 2007; Sun, 2010). Those semi-Markov perceptron systems are moderately faster than the heavy probabilistic systems using semi-Markov conditional random fields or latent variable conditional random fields. However, a disadvantage of the perceptron style systems is that they can not provide probabilistic information.

On the other hand, new word detection is also one of the important problems in Chinese information processing. Many statistical approaches have been proposed (J. Nie and Jin, 1995; Chen and Bai, 1998; Wu and Jiang, 2000; Peng et al., 2004; Chen and Ma, 2002; Zhou, 2005; Goh et al., 2003; Fu and Luke, 2004; Wu et al., 2011). New word detection is normally considered as a separate process from segmentation. There were studies trying to solve this problem jointly with CWS. However, the current studies are limited. Integrating the two tasks would benefit both segmentation and new word detection. Our method provides a convenient framework for doing this. Our new word detection is not a stand-alone process, but an integral part of segmentation.

2.2 Online Training

The most representative online training method is the SGD method. The SGD uses a small randomly-selected subset of the training samples to approximate the gradient of an objective function. The number of training samples used for this approximation is called the batch size. By using a smaller batch size, one can update the parameters more frequently and speed up the convergence. 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:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \gamma_t \nabla_{\boldsymbol{w}_t} \mathcal{L}_{stoch}(\boldsymbol{z}_i, \boldsymbol{w}_t), \qquad (1)$$

where t is the update counter, γ_t is the learning rate, and $\mathcal{L}_{stoch}(\boldsymbol{z}_i, \boldsymbol{w}_t)$ is the stochastic loss function based on a training sample $\boldsymbol{z}_i$.

There were accelerated versions of SGD, including stochastic meta descent (Vishwanathan et al., 2006) and periodic step-size adaptation online learning (Hsu et al., 2009). Compared with those two methods, our proposal is fundamentally

different. Those two methods are using 2nd-order gradient (Hessian) information for accelerated training, while our accelerated training method does not need such 2nd-order gradient information, which is costly and complicated. Our ADF training method is based on feature frequency adaptation, and there is no prior work on using feature frequency information for accelerating online training.

Other online training methods includes averaged SGD with feedback (Sun et al., 2010; Sun et al., 2011), latent variable perceptron training (Sun et al., 2009a), and so on. Those methods are less related to this paper.

3 System Architecture

3.1 A Joint Model Based on CRFs

First, we briefly review CRFs. CRFs are proposed as a method for structured classification by solving "the label bias problem" (Lafferty et al., 2001). Assuming a feature function that maps a pair of observation sequence $\boldsymbol{x}$ and label sequence $\boldsymbol{y}$ to a global feature vector $\boldsymbol{f}$, the probability of a label sequence $\boldsymbol{y}$ conditioned on the observation sequence $\boldsymbol{x}$ is modeled as follows (Lafferty et al., 2001):

$$P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w}) = \frac{\exp\left\{\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y},\boldsymbol{x})\right\}}{\sum_{\forall \boldsymbol{y}'} \exp\left\{\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}',\boldsymbol{x})\right\}}, \quad (2)$$

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

Given a training set consisting of n labeled sequences, $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i)$, for $i = 1 \ldots n$, parameter estimation is performed by maximizing the objective function,

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n} \log P(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{w}) - R(\boldsymbol{w}). \quad (3)$$

The first term of this equation represents a conditional log-likelihood of a training data. The second term is a regularizer for reducing overfitting. We employed an L2 prior, $R(\boldsymbol{w}) = \frac{||\boldsymbol{w}||^2}{2\sigma^2}$. In what follows, we denote the conditional log-likelihood of each sample $\log P(\boldsymbol{y}_i|\boldsymbol{x}_i,\boldsymbol{w})$ as $\ell(\boldsymbol{z}_i,\boldsymbol{w})$. The final objective function is as follows:

$$\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n} \ell(\boldsymbol{z}_i,\boldsymbol{w}) - \frac{||\boldsymbol{w}||^2}{2\sigma^2}. \quad (4)$$

Since no word list can be complete, new word identification is an important task in Chinese NLP. New words in input text are often incorrectly segmented into single-character or other very short words (Chen and Bai, 1998). This phenomenon will also undermine the performance of Chinese word segmentation. We consider here new word detection as an integral part of segmentation, aiming to improve both segmentation and new word detection: detected new words are added to the word list lexicon in order to improve segmentation. Based on our CRF word segmentation system, we can compute a probability for each segment. When we find some word segments are of reliable probabilities yet they are not in the existing word list, we then treat those "confident" word segments as new words and add them into the existing word list. Based on preliminary experiments, we treat a word segment as a new word if its probability is larger than 0.5. Newly detected words are re-incorporated into word segmentation for improving segmentation accuracies.

3.2 New Features

Here, we will describe high dimensional new features for the system.

3.2.1 Word-based Features

There are two ideas in deriving the refined features. The first idea is to exploit word features for node features of CRFs. Note that, although our model is a Markov CRF model, we can still use word features to learn word information in the training data. To derive word features, first of all, our system automatically collect a list of word unigrams and bigrams from the training data. To avoid overfitting, we only collect the word unigrams and bigrams whose frequency is larger than 2 in the training set. 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. The word-based feature templates derived for the label y_i are as follows:

- `unigram1`$(\boldsymbol{x},\ y_i) \leftarrow [x_{j,i},\ y_i]$, 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 $x_j \dots x_i$. $\mathbb{U}$ represents the unigram-dictionary collected from the training data.

- `unigram2`$(\boldsymbol{x},\ y_i) \leftarrow [x_{i,k},\ y_i]$, if the character sequence $x_{i,k}$ matches a word $w \in \mathbb{U}$, with the constraint $i < k < i + 6$.

- `bigram1`$(\boldsymbol{x},\ y_i) \leftarrow [x_{j,i-1},\ x_{i,k},\ y_i]$, 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`$(\boldsymbol{x},\ y_i) \leftarrow [x_{j,i},\ x_{i+1,k},\ y_i]$, 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.

We also employ the traditional character-based features. For each label y_i, we use the feature 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 $j = i - 2,\ \dots,\ i + 1$

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

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

3.2.2 High Dimensional Edge Features

The node features discussed above are based on a single label y_i. CRFs also have edge features that are based on label transitions. The second idea is to incorporate local observation information of $\boldsymbol{x}$ in edge features. For traditional implementation of CRF systems (e.g., the HCRF package), usually the edges features contain only the information of y_{i-1} and y_i, and without the information of the observation sequence (i.e., $\boldsymbol{x}$). The major reason for this simple realization of edge features in traditional CRF implementation is for reducing the dimension of features. Otherwise, there can be an explosion of edge features in some tasks. For example, in part-of-speech tagging tasks, there can be more than 40 labels and more than 1,600 types of label transitions. Therefore, incorporating local observation information into the edge feature will result in an explosion of edge features, which is 1,600 times larger than the number of feature templates.

Fortunately, for our task, the label set is quite small, $\mathbb{Y} = \{\mathtt{B}, \mathtt{I}, \mathtt{E}\}$[1]. There are only nine possible label transitions: $\mathbb{T} = \mathbb{Y} \times \mathbb{Y}$ and $|\mathbb{T}| = 9$.[2] As a result, the feature dimension will have nine times increase over the feature templates, if we incorporate local observation information of $\boldsymbol{x}$ into the edge features. In this way, we can effectively combine observation information of $\boldsymbol{x}$ with label transitions $y_{i-1}y_i$. We simply used the same templates of node features for deriving the new edge features. We found adding new edge features significantly improves the disambiguation power of our model.

4 Adaptive Online Gradient Descent based on Feature Frequency Information

As we will show in experiments, the training of the CRF model with high-dimensional new features is quite expensive, and the existing training method is not good enough. To solve this issue, we propose a fast online training method: adaptive online gradient descent based on feature frequency information (ADF). The proposed method is easy to implement.

For high convergence speed of online learning, we try to use more refined learning rates than the SGD training. Instead of using a single learning rate (a scalar) for all weights, we extend the learning rate scalar to a learning rate vector, which has the same dimension of the weight vector $\boldsymbol{w}$. The learning rate vector is automatically adapted based on feature frequency information. By doing so, each weight

[1] `B` means *beginning of a word*, `I` means *inside a word*, and `E` means *end of a word*. The `B`, `I`, `E` labels have been widely used in previous work of Chinese word segmentation (Sun et al., 2009b).

[2] The operator $\times$ means a Cartesian product between two sets.

ADF learning algorithm

procedure ADF(q, c, α, β)
  $\boldsymbol{w} \leftarrow 0, t \leftarrow 0, \boldsymbol{v} \leftarrow 0, \boldsymbol{\gamma} \leftarrow c$
  repeat until convergence
  . Draw a sample $\boldsymbol{z}_i$ at random
  . $\boldsymbol{v} \leftarrow$ UPDATE($\boldsymbol{v}, \boldsymbol{z}_i$)
  . **if** $t > 0$ and $t \bmod q = 0$
  . . $\boldsymbol{\gamma} \leftarrow$ UPDATE($\boldsymbol{\gamma}, \boldsymbol{v}$)
  . . $\boldsymbol{v} \leftarrow 0$
  . $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{w}} \mathcal{L}_{stoch}(\boldsymbol{z}_i, \boldsymbol{w})$
  . $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\gamma} \cdot \boldsymbol{g}$
  . $t \leftarrow t + 1$
  return $\boldsymbol{w}$

procedure UPDATE($\boldsymbol{v}, \boldsymbol{z}_i$)
  for $k \in$ features used in sample $\boldsymbol{z}_i$
  . $\boldsymbol{v}_k \leftarrow \boldsymbol{v}_k + 1$
  return $\boldsymbol{v}$

procedure UPDATE($\boldsymbol{\gamma}, \boldsymbol{v}$)
  for $k \in$ all features
  . $u \leftarrow \boldsymbol{v}_k / q$
  . $\eta \leftarrow \alpha - u(\alpha - \beta)$
  . $\boldsymbol{\gamma}_k \leftarrow \eta \boldsymbol{\gamma}_k$
  return $\boldsymbol{\gamma}$

Figure 1: The proposed ADF online learning algorithm. q, c, α, and β are hyper-parameters. q is an integer representing window size. c is for initializing the learning rates. α and β are the upper and lower bounds of a scalar, with $0 < \beta < \alpha < 1$.

has its own learning rate, and we will show that this can significantly improve the convergence speed of online learning.

In our proposed online learning method, the update formula is as follows:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{\gamma}_t \cdot \boldsymbol{g}_t. \tag{5}$$

The update term $\boldsymbol{g}_t$ is the gradient term of a randomly sampled instance:

$$\boldsymbol{g}_t = \nabla_{\boldsymbol{w}_t} \mathcal{L}_{stoch}(\boldsymbol{z}_i, \boldsymbol{w}_t) = \nabla_{\boldsymbol{w}_t} \left\{ \ell(\boldsymbol{z}_i, \boldsymbol{w}_t) - \frac{||\boldsymbol{w}_t||^2}{2n\sigma^2} \right\}.$$

In addition, $\boldsymbol{\gamma}_t \in \mathbb{R}_+^f$ is a positive vector-valued learning rate and $\cdot$ denotes component-wise (Hadamard) product of two vectors.

We learn the learning rate vector $\boldsymbol{\gamma}_t$ based on *feature frequency information in the updating process*. Our proposal is based on the intuition that a feature with higher frequency in the training process should be with a learning rate that decays faster. In other words, we assume a high frequency feature observed in the training process should have a small learning rate, and a low frequency feature should have a relatively larger learning rate in the training. Our assumption is based on the intuition that a weight with higher frequency is more adequately trained, hence smaller learning rate is preferable for fast convergence.

Given a window size q (number of samples in a window), we use a vector $\boldsymbol{v}$ to record the feature frequency. The k'th entry $\boldsymbol{v}_k$ corresponds to the frequency of the feature k in this window. Given a feature k, we use u to record the normalized frequency:

$$u = \boldsymbol{v}_k / q.$$

For each feature, an adaptation factor η is calculated based on the normalized frequency information, as follows:

$$\eta = \alpha - u(\alpha - \beta),$$

where α and β are the upper and lower bounds of a scalar, with $0 < \beta < \alpha < 1$. As we can see, a feature with higher frequency corresponds to a smaller scalar via linear approximation. Finally, the learning rate is updated as follows:

$$\boldsymbol{\gamma}_k \leftarrow \eta \boldsymbol{\gamma}_k.$$

With this setting, different features will correspond to different adaptation factors based on feature frequency information. Our ADF algorithm is summarized in Figure 1.

The ADF training method is efficient, because the additional computation (compared with SGD) is only the derivation of the learning rates, which is simple and efficient. As we know, the regularization of SGD can perform efficiently via the optimization based on sparse features (Shalev-Shwartz et al., 2007). Similarly, the derivation of $\boldsymbol{\gamma}_t$ can also perform efficiently via the optimization based on sparse features.

4.1 Convergence Analysis

Prior work on convergence analysis of existing online learning algorithms (Murata, 1998; Hsu et

Data	Method	Passes	Train-Time (sec)	NWD Rec	Pre	Rec	CWS F-score
MSR	Baseline	50	4.7e3	72.6	96.3	95.9	96.1
	+ New features	50	1.2e4	75.3	97.2	97.0	97.1
	+ New word detection	50	1.2e4	78.2	97.5	96.9	97.2
	+ ADF training	10	2.3e3	77.5	97.6	97.2	**97.4**
CU	Baseline	50	2.9e3	68.5	94.0	93.9	93.9
	+ New features	50	7.5e3	68.0	94.4	94.5	94.4
	+ New word detection	50	7.5e3	68.8	94.8	94.5	94.7
	+ ADF training	10	1.5e3	68.8	94.8	94.7	**94.8**
PKU	Baseline	50	2.2e3	77.2	95.0	94.0	94.5
	+ New features	50	5.2e3	78.4	95.5	94.9	95.2
	+ New word detection	50	5.2e3	79.1	95.8	94.9	95.3
	+ ADF training	10	1.2e3	78.4	95.8	94.9	**95.4**

Table 2: Incremental evaluations, by incrementally adding *new features* (word features and high dimensional edge features), *new word detection*, and *ADF training* (replacing SGD training with ADF training). Number of passes is decided by empirical convergence of the training methods.

	#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

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

al., 2009) can be extended to the proposed ADF training method. We can show that the proposed ADF learning algorithm has reasonable convergence properties.

When we have the smallest learning rate $\boldsymbol{\gamma}_{t+1} = \beta\boldsymbol{\gamma}_t$, the expectation of the obtained $\boldsymbol{w}_t$ is

$$E(\boldsymbol{w}_t) = \boldsymbol{w}^* + \prod_{m=1}^{t} (\boldsymbol{I} - \boldsymbol{\gamma}_0 \beta^m \boldsymbol{H}(\boldsymbol{w}^*))(\boldsymbol{w}_0 - \boldsymbol{w}^*),$$

where $\boldsymbol{w}^*$ is the optimal weight vector, and $\boldsymbol{H}$ is the Hessian matrix of the objective function. The rate of convergence is governed by the largest eigenvalue of the function $\boldsymbol{C}_t = \prod_{m=1}^{t}(\boldsymbol{I} - \boldsymbol{\gamma}_0 \beta^m \boldsymbol{H}(\boldsymbol{w}^*))$. Then, we can derive a bound of rate of convergence.

Theorem 1 *Assume ϕ is the largest eigenvalue of the function $\boldsymbol{C}_t = \prod_{m=1}^{t}(\boldsymbol{I} - \boldsymbol{\gamma}_0 \beta^m \boldsymbol{H}(\boldsymbol{w}^*))$. For the proposed ADF training, its convergence rate is bounded by ϕ, and we have*

$$\phi \leq \exp\{\frac{\boldsymbol{\gamma}_0 \lambda \beta}{\beta - 1}\},$$

where λ is the minimum eigenvalue of $\boldsymbol{H}(\boldsymbol{w}^*)$.

5 Experiments

5.1 Data and Metrics

We used benchmark datasets provided by the second International Chinese Word Segmentation Bakeoff to test our proposals. The datasets are from Microsoft Research Asia (MSR), City University of Hongkong (CU), and Peking University (PKU). Details of the corpora are listed in Table 1. We did not use any extra resources such as common surnames, parts-of-speech, and semantics.

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 defined by $2PR/(P + R)$, and recall of new word detection (NWD recall). For more detailed information on the corpora, refer to Emerson (2005).

5.2 Features, Training, and Tuning

We employed the feature templates defined in Section 3.2. The feature sets are huge. There are 2.4×10^7 features for the MSR data, 4.1×10^7 features for the CU data, and 4.7×10^7 features for the PKU data. To generate word-based features, we extracted high-frequency word-based unigram and bigram lists from the training data.

As for training, we performed gradient descent

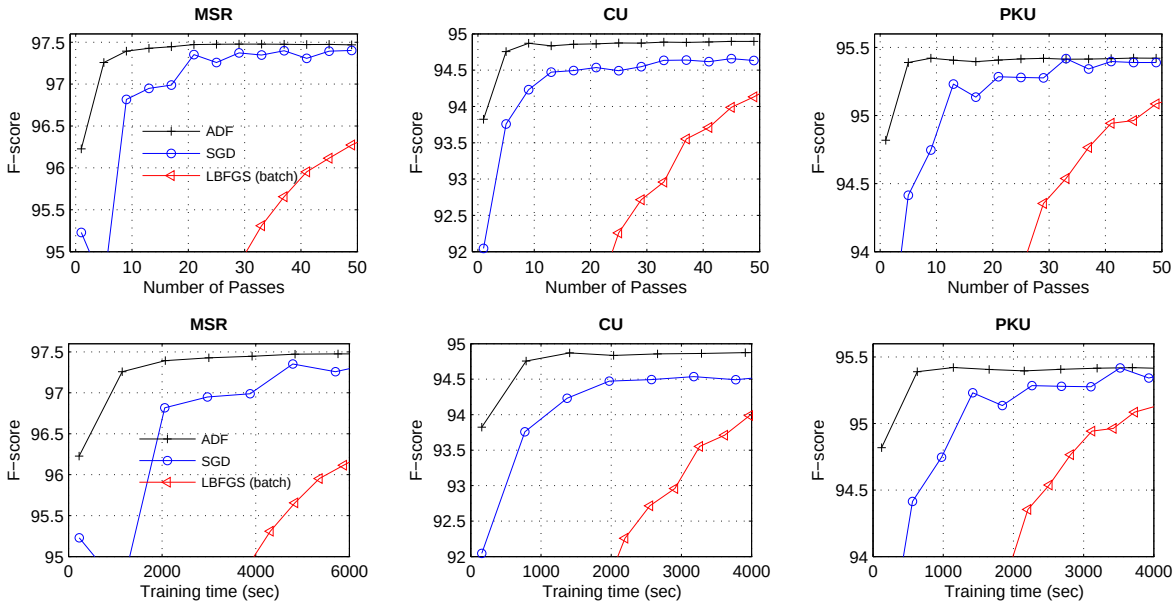

Figure 2: F-score curves on the MSR, CU, and PKU datasets: ADF learning *vs.* SGD and LBFGS training methods.

with our proposed training method. To compare with existing methods, we chose two popular training methods, a batch training one and an online training one. The batch training method is the Limited-Memory BFGS (LBFGS) method (Nocedal and Wright, 1999). The online baseline training method is the SGD method, which we have introduced in Section 2.2.

For the ADF training method, we need to tune the hyper-parameters q, c, α, and β. Based on automatic tuning within the training data (validation in the training data), we found it is proper to set $q = n/10$ (n is the number of training samples), $c = 0.1$, $\alpha = 0.995$, and $\beta = 0.6$. To reduce overfitting, we employed an L_2 Gaussian weight prior (Chen and Rosenfeld, 1999) for all training methods. We varied the σ with different values (e.g., 1.0, 2.0, and 5.0), and finally set the value to 1.0 for all training methods.

5.3 Results and Discussion

First, we performed incremental evaluation in this order: Baseline (word segmentation model with SGD training); Baseline + New features; Baseline + New features + New word detection; Baseline + New features + New word detection + ADF training (replacing SGD training). The results are shown in Table 2.

As we can see, the new features improved performance on both word segmentation and new word detection. However, we also noticed that the training cost became more expensive via adding high dimensional new features. Adding new word detection function further improved the segmentation quality and the new word recognition recall. Finally, by using the ADF training method, the training speed is much faster than the SGD training method. The ADF method can achieve empirical optimum in only a few passes, yet with better segmentation accuracies than the SGD training with 50 passes.

To get more details of the proposed training method, we compared it with SGD and LBFGS training methods based on an identical platform, by varying the number of passes. The comparison was based on the same platform: Baseline + New features + New word detection. The F-score curves of the training methods are shown in Figure 2. Impressively, the ADF training method reached empirical convergence in only a few passes, while the SGD and LBFGS training converged much slower, requiring more than 50 passes. The ADF training is about an order magnitude faster than the SGD online training and more than an order magnitude faster than the LBFGS batch training.

Finally, we compared our method with the state-

Data	Method	Prob.	Pre	Rec	F-score
MSR	Best05 (Tseng et al., 2005)	$\surd$	96.2	96.6	96.4
	CRF + rule-system (Zhang et al., 2006)	$\surd$	97.2	96.9	97.1
	Semi-Markov perceptron (Zhang and Clark, 2007)	$\times$	N/A	N/A	97.2
	Semi-Markov CRF (Gao et al., 2007)	$\surd$	N/A	N/A	97.2
	Latent-variable CRF (Sun et al., 2009b)	$\surd$	97.3	97.3	97.3
	Our method (A Single CRF)	$\surd$	97.6	97.2	**97.4**
CU	Best05 (Tseng et al., 2005)	$\surd$	94.1	94.6	94.3
	CRF + rule-system (Zhang et al., 2006)	$\surd$	95.2	94.9	95.1
	Semi-perceptron (Zhang and Clark, 2007)	$\times$	N/A	N/A	95.1
	Latent-variable CRF (Sun et al., 2009b)	$\surd$	94.7	94.4	94.6
	Our method (A Single CRF)	$\surd$	94.8	94.7	94.8
PKU	Best05 (Chen et al., 2005)	N/A	95.3	94.6	95.0
	CRF + rule-system (Zhang et al., 2006)	$\surd$	94.7	95.5	95.1
	semi-perceptron (Zhang and Clark, 2007)	$\times$	N/A	N/A	94.5
	Latent-variable CRF (Sun et al., 2009b)	$\surd$	95.6	94.8	95.2
	Our method (A Single CRF)	$\surd$	95.8	94.9	**95.4**

Table 3: Comparing our method with the state-of-the-art CWS systems.

of-the-art systems reported in the previous papers. The statistics are listed in Table 3. *Best05* represents the best system of the Second International Chinese Word Segmentation Bakeoff on the corresponding data; *CRF + rule-system* represents confidence-based combination of CRF and rule-based models, presented in Zhang et al. (2006). *Prob.* indicates whether or not the system can provide probabilistic information. As we can see, our method achieved similar or even higher F-scores, compared with the best systems reported in previous papers. Note that, our system is a single Markov model, while most of the state-of-the-art systems are complicated heavy systems, with model-combinations (e.g., voting of multiple segmenters), semi-Markov relaxations, or latent-variables.

6 Conclusions and Future Work

In this paper, we presented a joint model for Chinese word segmentation and new word detection. We presented new features, including word-based features and enriched edge features, for the joint modeling. We showed that the new features can improve the performance on the two tasks.

On the other hand, the training of the model, especially with high-dimensional new features, became quite expensive. To solve this problem, we proposed a new training method, ADF training, for very fast training of CRFs, even given large-scale datasets with high dimensional features. We performed experiments and showed that our new training method is an order magnitude faster than existing optimization methods. Our final system can learn highly accurate models with only a few passes in training. The proposed fast learning method is a general algorithm that is not limited in this specific task. As future work, we plan to apply this fast learning method on other large-scale natural language processing tasks.

Acknowledgments

We thank Yaozhong Zhang and Weiwei Sun for helpful discussions on word segmentation techniques. The work described in this paper was supported by a Hong Kong RGC Project (No. PolyU 5230/08E), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), and National Natural Science Foundation of China (No.91024009, No.60973053).

References

Galen Andrew. 2006. A hybrid markov/semi-markov conditional random field for sequence segmentation.

In *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. In *Proceedings of The Fourth SIGHAN Workshop*, pages 134–137.

K.J. Chen and M.H. Bai. 1998. Unknown word detection for chinese by a corpus-based learning method. *Computational Linguistics and Chinese Language Processing*, 3(1):27–44.

Keh-Jiann Chen and Wei-Yun Ma. 2002. Unknown word extraction for chinese documents. In *Proceedings of COLING'02*.

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. In *Proceedings of the fourth SIGHAN workshop*, pages 138–141.

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

Guohong Fu and Kang-Kwong Luke. 2004. Chinese unknown word identification using class-based lm. In *Proceedings of IJCNLP'04*, volume 3248 of *Lecture Notes in Computer Science*, pages 704–713. Springer.

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.

Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. 2003. Chinese unknown word identification using character-based tagging and chunking. In Kotaro Funakoshi, Sandra Kbler, and Jahna Otterbacher, editors, *Proceedings of ACL (Companion)'03*, pages 197–200.

Chun-Nan Hsu, Han-Shen Huang, Yu-Ming Chang, and Yuh-Jye Lee. 2009. Periodic step-size adaptation in second-order gradient descent for single-pass on-line structured learning. *Machine Learning*, 77(2-3):195–224.

M. Hannan J. Nie and W. Jin. 1995. Unknown word detection and segmentation of chinese using statistical and heuristic knowledge. *Communications of the Chinese and Oriental Languages Information Processing Society*, 5:47C57.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 282–289.

Noboru Murata. 1998. A statistical study of on-line learning. *In On-line learning in neural networks, Cambridge University Press*, pages 63–92.

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

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of Coling 2004*, pages 562–568, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of ICML'07*.

Xu Sun and Jun'ichi Tsujii. 2009. Sequential labeling with latent variables: An exact inference algorithm and its efficient approximation. In *Proceedings of EACL'09*, pages 772–780, Athens, Greece, March.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. In *Proceedings of COLING'08*, pages 841–848, Manchester, UK.

Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun'ichi Tsujii. 2009a. Latent variable perceptron algorithm for structured classification. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 1236–1242.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009b. A discriminative latent variable chinese segmenter with hybrid word/character information. In *Proceedings of NAACL-HLT'09*, pages 56–64, Boulder, Colorado, June.

Xu Sun, Hisashi Kashima, Takuya Matsuzaki, and Naonori Ueda. 2010. Averaged stochastic gradient descent with feedback: An accurate, robust, and fast training method. In *Proceedings of the 10th International Conference on Data Mining (ICDM'10)*, pages 1067–1072.

Xu Sun, Hisashi Kashima, Ryota Tomioka, and Naonori Ueda. 2011. Large scale real-life action recognition using conditional random fields with stochastic training. In *Proceedings of the 15th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'11)*.

Weiwei Sun. 2010. Word-based and character-based word segmentation models: Comparison and combination. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING'10 (Posters)*, pages 1211–1219. Chinese Information Processing Society of China.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A

conditional random field word segmenter for sighan bakeoff 2005. In *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. In *Proceedings of ICML'06*, pages 969–976.

A. Wu and Z. Jiang. 2000. Statistically-enhanced new word identification in a rule-based chinese system. In *Proceedings of the Second Chinese Language Processing Workshop*, page 46C51, Hong Kong, China.

Yi-Lun Wu, Chaio-Wen Hsieh, Wei-Hsuan Lin, Chun-Yi Liu, and Liang-Chih Yu. 2011. Unknown word extraction from multilingual code-switching sentences (in chinese). In *Proceedings of ROCLING (Posters)'11*, pages 349–360.

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

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847, Prague, Czech Republic, June. Association for Computational Linguistics.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 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*, pages 193–196, New York City, USA, June. Association for Computational Linguistics.

Hai Zhao, Changning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for chinese word segmentation. *ACM Trans. Asian Lang. Inf. Process.*, 9(2).

Guodong Zhou. 2005. A chunking strategy towards unknown word detection in chinese word segmentation. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *Proceedings of IJCNLP'05*, volume 3651 of *Lecture Notes in Computer Science*, pages 530–541. Springer.