Joint Training and Decoding Using Virtual Nodes for Cascaded Segmentation and Tagging Tasks

Xian Qian, Qi Zhang, Yaqian Zhou, Xuanjing Huang, Lide Wu
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, P.R.China
{qianxian, qz, zhouyaqian, xjhuang, ldwu}@fudan.edu.cn

Abstract

Many sequence labeling tasks in NLP require solving a cascade of segmentation and tagging subtasks, such as Chinese POS tagging, named entity recognition, and so on. Traditional pipeline approaches usually suffer from error propagation. Joint training/decoding in the cross-product state space could cause too many parameters and high inference complexity. In this paper, we present a novel method which integrates graph structures of two subtasks into one using virtual nodes, and performs joint training and decoding in the factorized state space. Experimental evaluations on CoNLL 2000 shallow parsing data set and Fourth SIGHAN Bakeoff CTB POS tagging data set demonstrate the superiority of our method over cross-product, pipeline and candidate reranking approaches.

1 Introduction

There is a typical class of sequence labeling tasks in many natural language processing (NLP) applications, which require solving a cascade of segmentation and tagging subtasks. For example, many Asian languages such as Japanese and Chinese which do not contain explicitly marked word boundaries, word segmentation is the preliminary step for solving part-of-speech (POS) tagging problem. Sentences are firstly segmented into words, then each word is assigned with a part-of-speech tag. Both syntactic parsing and dependency parsing usually start with a textual input that is tokenized, and POS tagged.

The most commonly approach solves cascaded subtasks in a pipeline, which is very simple to implement and allows for a modular approach. While, the key disadvantage of such method is that errors propagate between stages, significantly affecting the quality of the final results. To cope with this problem, Shi and Wang (2007) proposed a reranking framework in which N-best segment candidates generated in the first stage are passed to the tagging model, and the final output is the one with the highest overall segmentation and tagging probability score. The main drawback of this method is that the interaction between tagging and segmentation is restricted by the number of candidate segmentation outputs. Razvan C. Bunescu (2008) presented an improved pipeline model in which upstream subtask outputs are regarded as hidden variables, together with their probabilities are used as probabilistic features in the downstream subtasks. One shortcoming of this method is that calculation of marginal probabilities of features may be inefficient and some approximations are required for fast computation. Another disadvantage of these two methods is that they employ separate training and the segmentation model could not take advantages of tagging information in the training procedure.

On the other hand, joint learning and decoding using cross-product of segmentation states and tagging states does not suffer from error propagation problem and achieves higher accuracy on both subtasks (Ng and Low, 2004). However, two problems arises due to the large state space, one is that the amount of parameters increases rapidly, which is apt to overfit on the training corpus, the other is that the inference by dynamic programming could be inefficient. Sutton (2004) proposed Dynamic Conditional Random Fields (DCRFs) to perform joint training/decoding of subtasks using much fewer parameters than the cross-product approach. How-

ever, DCRFs do not guarantee non-violation of hard-constraints that nodes within the same segment get a single consistent tagging label. Another drawback of DCRFs is that exact inference is generally time consuming, some approximations are required to make it tractable.

Recently, perceptron based learning framework has been well studied for incorporating node level and segment level features together (Kazama and Torisawa, 2007; Zhang and Clark, 2008). The main shortcoming is that exact inference is intractable for those dynamically generated segment level features, so candidate based searching algorithm is used for approximation. On the other hand, Jiang (2008) proposed a cascaded linear model which has a two layer structure, the inside-layer model uses node level features to generate candidates with their weights as inputs of the outside layer model which captures non-local features. As pipeline models, error propagation problem exists for such method.

In this paper, we present a novel graph structure that exploits joint training and decoding in the factorized state space. Our method does not suffer from error propagation, and guards against violations of those hard-constraints imposed by segmentation subtask. The motivation is to integrate two Markov chains for segmentation and tagging subtasks into a single chain, which contains two types of nodes, then standard dynamic programming based exact inference is employed on the hybrid structure. Experiments are conducted on two different tasks, CoNLL 2000 shallow parsing and SIGHAN 2008 Chinese word segmentation and POS tagging. Evaluation results of shallow parsing task show the superiority of our proposed method over traditional joint training/decoding approach using cross-product state space, and achieves the best reported results when no additional resources at hand. For Chinese word segmentation and POS tagging task, a strong baseline pipeline model is built, experimental results show that the proposed method yields a more substantial improvement over the baseline than candidate reranking approach.

The rest of this paper is organized as follows: In Section 2, we describe our novel graph structure. In Section 3, we analyze complexity of our proposed method. Experimental results are shown in Section 4. We conclude the work in Section 5.

2 Multi-chain integration using Virtual Nodes

2.1 Conditional Random Fields

We begin with a brief review of the Conditional Random Fields(CRFs). Let $\mathbf{x} = x_1 x_2 \dots x_l$ denote the observed sequence, where x_i is the i^{th} node in the sequence, l is sequence length, $\mathbf{y} = y_1 y_2 \dots y_l$ is a label sequence over $\mathbf{x}$ that we wish to predict. CRFs (Lafferty et al., 2001) are undirected graphic models that use Markov network distribution to learn the conditional probability. For sequence labeling task, linear chain CRFs are very popular, in which a first order Markov assumption is made on the labels:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \phi(\mathbf{x}, \mathbf{y}, i)$$

,where

$$\phi(\mathbf{x}, \mathbf{y}, i) = \exp\left(\mathbf{w}^T \mathbf{f}(\mathbf{x}, y_{i-1}, y_i, i)\right)$$
$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_i \phi(\mathbf{x}, \mathbf{y}, i)$$

$\mathbf{f}(\mathbf{x}, y_{i-1}, y_i, i) = [f_1(\mathbf{x}, y_{i-1}, y_i, i), \dots, f_m(\mathbf{x}, y_{i-1}, y_i, i)]^T$, each element $f_j(\mathbf{x}, y_{i-1}, y_i, i)$ is a real valued feature function, here we simplify the notation of state feature by writing $f_j(\mathbf{x}, y_i, i) = f_j(\mathbf{x}, y_{i-1}, y_i, i)$, m is the cardinality of feature set $\{f_j\}$. $\mathbf{w} = [w_1, \dots, w_m]^T$ is a weight vector to be learned from the training set. $Z(\mathbf{x})$ is the normalization factor over all label sequences for $\mathbf{x}$.

In the traditional joint training/decoding approach for cascaded segmentation and tagging task, each label y_i has the form s_i-t_i, which consists of segmentation label s_i and tagging label t_i. Let $\mathbf{s} = s_1 s_2 \dots s_l$ be the segmentation label sequence over $\mathbf{x}$. There are several commonly used label sets such as BI, BIO, IOE, BIES, etc. To facilitate our discussion, in later sections we will use BIES label set, where B,I,E represents *Beginning, Inside* and *End* of a multi-node segment respectively, S denotes a single node segment. Let $\mathbf{t} = t_1 t_2 \dots t_l$ be the tagging label sequence over $\mathbf{x}$. For example, in named entity recognition task, $t_i \in$ {PER, LOC, ORG, MISC, O} represents an entity type (person name, location name, organization name, miscellaneous entity

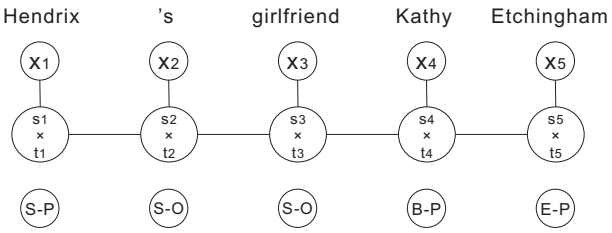

Figure 1: Graphical representation of linear chain CRFs for traditional joint learning/decoding

name and other). Graphical representation of linear chain CRFs is shown in Figure 1, where tagging label "P" is the simplification of "PER". For nodes that are labeled as other, we define s_i =S, t_i =O.

2.2 Hybrid structure for cascaded labeling tasks

Different from traditional joint approach, our method integrates two linear markov chains for segmentation and tagging subtasks into one that contains two types of nodes. Specifically, we first regard segmentation and tagging as two independent sequence labeling tasks, corresponding chain structures are built, as shown in the top and middle sub-figures of Figure 2. Then a chain of twice length of the observed sequence is built, where nodes $x_1, \ldots, x_l$ on the even positions are original observed nodes, while nodes $v_1, \ldots, v_l$ on the odd positions are virtual nodes that have no content information. For original nodes x_i, the state space is the tagging label set, while for virtual nodes, their states are segmentation labels. The label sequence of the hybrid chain is $\mathbf{y} = y_1 \ldots y_{2l} = s_1 t_1 \ldots s_l t_l$, where combination of consecutive labels $s_i t_i$ represents the full label for node x_i.

Then we let s_i be connected with s_{i-1} and s_{i+1} , so that first order Markov assumption is made on segmentation states. Similarly, t_i is connected with t_{i-1} and t_{i+1}. Then neighboring tagging and segmentation states are connected as shown in the bottom sub-figure of Figure 2. Non-violation of hard-constraints that nodes within the same segment get a single consistent tagging label is guaranteed by introducing second order transition features $f(t_{i-1}, s_i, t_i, i)$ that are true if $t_{i-1} \neq t_i$ and $s_i \in \{\text{I,E}\}$. For example, $f_j(t_{i-1}, s_i, t_i, i)$ is defined as true if t_{i-1} =PER, s_i =I and t_i =LOC. In other words, it is true, if a segment is partially tagging as PER, and partially tagged as LOC. Since such features are always false in the training corpus, their corresponding weights will be very low so that inconsistent label assignments impossibly appear in decoding procedure. The hybrid graph structure can be regarded as a special case of second order Markov chain.

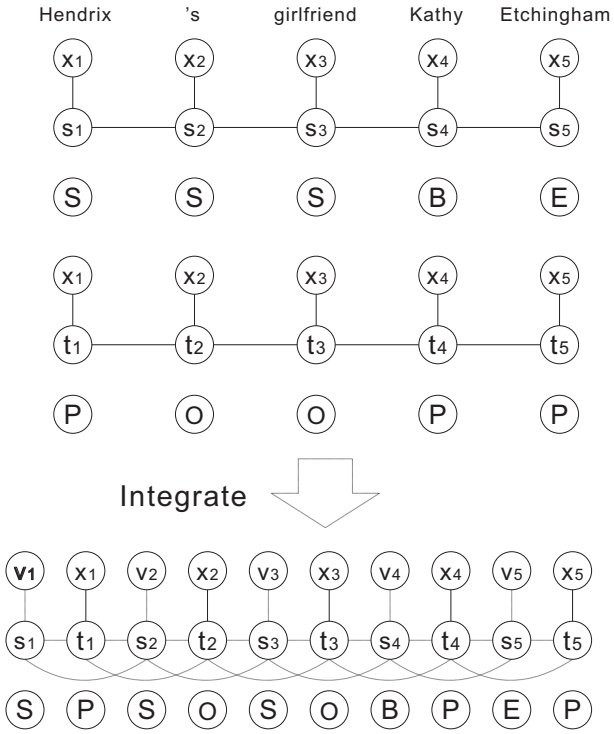

Figure 2: Multi-chain integration using Virtual Nodes

2.3 Factorized features

Compared with traditional joint model that exploits cross-product state space, our hybrid structure uses factorized states, hence could handle more flexible features. Any state feature $g(\mathbf{x}, y_i, i)$ defined in the cross-product state space can be replaced by a first order transition feature in the factorized space: $f(\mathbf{x}, s_i, t_i, i)$. As for the transition features, we use $f(s_{i-1}, t_{i-1}, s_i, i)$ and $f(t_{i-1}, s_i, t_i, i)$ instead of $g(y_{i-1}, y_i, i)$ in the conventional joint model.

Features in cross-product state space require that segmentation label and tagging label take on particular values simultaneously, however, sometimes we

want to specify requirement on only segmentation or tagging label. For example, "Smith" may be an end of a person name, "Speaker: John Smith"; or a single word person name "Professor Smith will . . . ". In such case, our observation is that "Smith" is likely a (part of) person name, we do not care about its segmentation label. So we could define state feature $f(\mathbf{x}, t_i, i) = true$, if x_i is "Smith" with tagging label t_i=PER.

Further more, we could define features like $f(\mathbf{x}, t_{i-1}, t_i, i)$, $f(\mathbf{x}, s_{i-1}, s_i, i)$, $f(\mathbf{x}, t_{i-1}, s_i, i)$, etc. The hybrid structure facilitates us to use varieties of features. In the remainder of the paper, we use notations $f(\mathbf{x}, t_{i-1}, s_i, t_i, i)$ and $f(\mathbf{x}, s_{i-1}, t_{i-1}, s_i, i)$ for simplicity.

2.4 Hybrid CRFs

A hybrid CRFs is a conditional distribution that factorizes according to the hybrid graphical model, and is defined as:

$$p(\mathbf{s}, \mathbf{t}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \phi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i) \prod_i \psi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i)$$

Where

$$\phi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i) = \exp\left(\mathbf{w}_1^T \mathbf{f}(\mathbf{x}, s_{i-1}, t_{i-1}, s_i)\right)$$
$$\psi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i) = \exp\left(\mathbf{w}_2^T \mathbf{f}(\mathbf{x}, t_{i-1}, s_i, t_i)\right)$$
$$Z(\mathbf{x}) = \sum_{\mathbf{s},\mathbf{t}} \left(\prod_i \phi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i) \prod_i \psi(\mathbf{x}, \mathbf{s}, \mathbf{t}, i) \right)$$

Where $\mathbf{w}_1$, $\mathbf{w}_2$ are weight vectors.

Luckily, unlike DCRFs, in which graph structure can be very complex, and the cross-product state space can be very large, in our cascaded labeling task, the segmentation label set is often small, so far as we known, the most complicated segmentation label set has only 6 labels (Huang and Zhao, 2007). So exact dynamic programming based algorithms can be efficiently performed.

In the training stage, we use second order forward backward algorithm to compute the marginal probabilities $p(\mathbf{x}, s_{i-1}, t_{i-1}, s_i)$ and $p(\mathbf{x}, t_{i-1}, s_i, t_i)$, and the normalization factor $Z(\mathbf{x})$. In decoding stage, we use second order Viterbi algorithm to find the best label sequence. The Viterbi decoding can be used to label a new sequence, and marginal computation is used for parameter estimation.

Table 1: Time Complexity

Method	Training	Decoding
Pipeline	$(\|S\|^2 c_s + \|T\|^2 c_t)L$	$(\|S\|^2 + \|T\|^2)U$
Cross-Product	$(\|S\|\|T\|)^2 cL$	$(\|S\|\|T\|)^2 U$
Reranking	$(\|S\|^2 c_s + \|T\|^2 c_t)L$	$(\|S\|^2 + \|T\|^2)NU$
Hybrid	$(\|S\| + \|T\|)\|S\|\|T\|cL$	$(\|S\| + \|T\|)\|S\|\|T\|U$

3 Complexity Analysis

The time complexity of the hybrid CRFs training and decoding procedures is higher than that of pipeline methods, but lower than traditional cross-product methods. Let

- $|S|$ = size of the segmentation label set.
- $|T|$ = size of the tagging label set.
- L = total number of nodes in the training data set.
- U = total number of nodes in the testing data set.
- c = number of joint training iterations.
- c_s = number of segmentation training iterations.
- c_t = number of tagging training iterations.
- N = number of candidates in candidate reranking approach.

Time requirements for pipeline, cross-product, candidate reranking and hybrid CRFs are summarized in Table 1. For Hybrid CRFs, original node x_i has features $\{f_j(t_{i-1}, s_i, t_i)\}$, accessing all label subsequences $t_{i-1}s_it_i$ takes $|S||T|^2$ time, while virtual node v_i has features $\{f_j(s_{i-1}, t_{i-1}, s_i)\}$, accessing all label subsequences $s_{i-1}t_{i-1}s_i$ takes $|S|^2|T|$ time, so the final complexity is $(|S| + |T|)|S||T|cL$.

In real applications, $|S|$ is small, $|T|$ could be very large, we assume that $|T| >> |S|$, so for each iteration, hybrid CRFs is about $|S|$ times slower than pipeline and $|S|$ times faster than cross-product

Table 2: Feature templates for shallow parsing task

Cross Product CRFs	Hybrid CRFs
$w_{i-2}y_i$, $w_{i-1}y_i$, w_iy_i $w_{i+1}y_i$, $w_{i+2}y_i$	$w_{i-1}s_i$, w_is_i, $w_{i+1}s_i$
	$w_{i-2}t_i$, $w_{i-1}t_i$, w_it_i, $w_{i+1}t_i$, $w_{i+2}t_i$
$w_{i-1}w_iy_i$, $w_iw_{i+1}y_i$	$w_{i-1}w_is_i$, $w_iw_{i+1}s_i$
	$w_{i-1}w_it_i$, $w_iw_{i+1}t_i$
$p_{i-2}y_i$, $p_{i-1}y_i$, p_iy_i $p_{i+1}y_i$, $p_{i+2}y_i$	$p_{i-1}s_i$, p_is_i, $p_{i+1}s_i$
	$p_{i-2}t_i$, $p_{i-1}t_i$, $p_{i+1}t_i$, $p_{i+2}t_i$
$p_{i-2}p_{i-1}y_i$, $p_{i-1}p_iy_i$, $p_ip_{i+1}y_i$, $p_{i+1}p_{i+2}y_i$	$p_{i-2}p_{i-1}s_i$, $p_{i-1}p_is_i$, $p_ip_{i+1}s_i$, $p_{i+1}p_{i+2}s_i$
	$p_{i-3}p_{i-2}t_i$, $p_{i-2}p_{i-1}t_i$, $p_{i-1}p_it_i$, $p_ip_{i+1}t_i$, $p_{i+1}p_{i+2}t_i$, $p_{i+2}p_{i+3}t_i$, $p_{i-1}p_{i+1}t_i$
$p_{i-2}p_{i-1}p_iy_i$, $p_{i-1}p_ip_{i+1}y_i$, $p_ip_{i+1}p_{i+2}y_i$	$p_{i-2}p_{i-1}p_is_i$, $p_{i-1}p_ip_{i+1}s_i$, $p_ip_{i+1}p_{i+2}s_i$
	$w_ip_it_i$
	$w_is_{i-1}s_i$
	$w_{i-1}t_{i-1}t_i$, $w_it_{i-1}t_i$, $p_{i-1}t_{i-1}t_i$, $p_it_{i-1}t_i$
$y_{i-1}y_i$	$s_{i-1}t_{i-1}s_i$, $t_{i-1}s_it_i$

method. When decoding, candidate reranking approach requires more time if candidate number $N > |S|$.

Though the space complexity could not be compared directly among some of these methods, hybrid CRFs require less parameters than cross-product CRFs due to the factorized state space. This is similar with factorized CRFs (FCRFs) (Sutton et al., 2004).

4 Experiments

4.1 Shallow Parsing

Our first experiment is the shallow parsing task. We use corpus from CoNLL 2000 shared task, which contains 8936 sentences for training and 2012 sentences for testing. There are 11 tagging labels: noun phrase(NP), verb phrase(VP) , ... and other (O), the segmentation state space we used is BIES label set, since we find that it yields a little improvement over BIO set.

We use the standard evaluation metrics, which are precision P (percentage of output phrases that exactly match the reference phrases), recall R (percentage of reference phrases returned by our system), and their harmonic mean, the F1 score $F1 = \frac{2PR}{P+R}$ (which we call F score in what follows).

We compare our approach with traditional cross-product method. To find good feature templates, development data are required. Since CoNLL2000 does not provide development data set, we divide the training data into 10 folds, of which 9 folds for training and 1 fold for developing. After selecting feature templates by cross validation, we extract features and learn their weights on the whole training data set. Feature templates are summarized in Table 2, where w_i denotes the i^{th} word, p_i denotes the i^{th} POS tag.

Notice that in the second row, feature templates of the hybrid CRFs does not contain $w_{i-2}s_i$, $w_{i+2}s_i$, since we find that these two templates degrade performance in cross validation. However, $w_{i-2}t_i$, $w_{i+2}t_i$ are useful, which implies that the proper context window size for segmentation is smaller than tagging. Similarly, for hybrid CRFs, the window size of POS bigram features for segmentation is 5 (from p_{i-2} to p_{i+2}, see the eighth row in the second column); while for tagging, the size is 7 (from p_{i-3} to p_{i+3}, see the ninth row in the second column). However for cross-product method, their window sizes must be consistent.

For traditional cross-product CRFs and our hybrid CRFs, we use fixed gaussian prior $\sigma = 1.0$ for both methods, we find that this parameter does not signifi-

Table 3: Results for shallow parsing task, Hybrid CRFs significantly outperform Cross-Product CRFs (McNemar's test; $p < 0.01$)

Method	Cross-Product CRFs	Hybrid CRFs
Training Time	11.6 hours	6.3 hours
Feature Number	13 million	10 million
Iterations	118	141
F1	93.88	94.31

cantly affect the results when it varies between 1 and 10. LBFGS(Nocedal and Wright, 1999) method is employed for numerical optimization. Experimental results are shown in Table 3. Our proposed CRFs achieve a performance gain of 0.43 points in F-score over cross-product CRFs that use state space while require less training time.

For comparison, we also listed the results of previous top systems, as shown in Table 4. Our proposed method outperforms other systems when no additional resources at hand. Though recently semi-supervised learning that incorporates large mounts of unlabeled data has been shown great improvement over traditional supervised methods, such as the last row in Table 4, supervised learning is fundamental. We believe that combination of our method and semi-supervised learning will achieve further improvement.

4.2 Chinese word segmentation and POS tagging

Our second experiment is the Chinese word segmentation and POS tagging task. To facilitate comparison, we focus only on the closed test, which means that the system is trained only with a designated training corpus, any extra knowledge is not allowed, including Chinese and Arabic numbers, letters and so on. We use the Chinese Treebank (CTB) POS corpus from the Fourth International SIGHAN Bakeoff data sets (Jin and Chen, 2008). The training data consist of 23444 sentences, 642246 Chinese words, 1.05M Chinese characters and testing data consist of 2079 sentences, 59955 Chinese words, 0.1M Chinese characters.

We compare our hybrid CRFs with pipeline and candidate reranking methods (Shi and Wang, 2007)

Table 4: Comparison with other systems on shallow parsing task

Method	F1	Additional Resources
Cross-Product CRFs	93.88	none
Hybrid CRFs	**94.31**	
SVM combination (Kudo and Matsumoto, 2001)	93.91	
Voted Perceptrons (Carreras and Marquez, 2003)	93.74	
ETL (Milidiu et al., 2008)	92.79	
(Wu et al., 2006)	94.21	Extended features such as token features, affixes
HySOL (Suzuki et al., 2007)	94.36	$17M$ words unlabeled data
ASO-semi (Ando and Zhang, 2005)	94.39	$15M$ words unlabeled data
(Zhang et al., 2002)	94.17	full parser output
(Suzuki and Isozaki, 2008)	95.15	$1G$ words unlabeled data

using the same evaluation metrics as shallow parsing. We do not compare with cross-product CRFs due to large amounts of parameters.

For pipeline method, we built our word segmenter based on the work of Huang and Zhao (2007), which uses 6 label representation, 7 feature templates (listed in Table 5, where c_i denotes the i^{th} Chinese character in the sentence) and CRFs for parameter learning. We compare our segmentor with other top systems using SIGHAN CTB corpus and evaluation metrics. Comparison results are shown in Table 6, our segmenter achieved 95.12 F-score, which is ranked 4th of 26 official runs. Except for the first system which uses extra unlabeled data, differences between rest systems are not significant.

Our POS tagging system is based on linear chain CRFs. Since SIGHAN dose not provide development data, we use the 10 fold cross validation described in the previous experiment to turning feature templates and Gaussian prior. Feature templates are listed in Table 5, where w_i denotes the i^{th} word in

Table 5: Feature templates for Chinese word segmentation and POS tagging task

Segmentation feature templates
(1.1) $c_{i-2}s_i$, $c_{i-1}s_i$, c_is_i, $c_{i+1}s_i$, $c_{i+2}s_i$
(1.2) $c_{i-1}c_is_i$, $c_ic_{i+1}s_i$, $c_{i-1}c_{i+1}s_i$
(1.3) $s_{i-1}s_i$
POS tagging feature templates
(2.1) $w_{i-2}t_i$, $w_{i-1}t_i$, w_it_i, $w_{i+1}t_i$, $w_{i+2}t_i$
(2.2) $w_{i-2}w_{i-1}t_i$, $w_{i-1}w_it_i$, $w_iw_{i+1}t_i$, $w_{i+1}w_{i+2}t_i$, $w_{i-1}w_{i+1}t_i$
(2.3) $c_1(w_i)t_i$, $c_2(w_i)t_i$, $c_3(w_i)t_i$, $c_{-2}(w_i)t_i$, $c_{-1}(w_i)t_i$
(2.4) $c_1(w_i)c_2(w_i)t_i$, $c_{-2}(w_i)c_{-1}(w_i)t_i$
(2.5) $l(w_i)t_i$
(2.6) $t_{i-1}t_i$
Joint segmentation and POS tagging feature templates
(3.1) $c_{i-2}s_i$, $c_{i-1}s_i$, c_is_i, $c_{i+1}s_i$, $c_{i+2}s_i$
(3.2) $c_{i-1}c_is_i$, $c_ic_{i+1}s_i$, $c_{i-1}c_{i+1}s_i$
(3.3) $c_{i-3}t_i$, $c_{i-2}t_i$, $c_{i-1}t_i$, c_it_i, $c_{i+1}t_i$, $c_{i+2}t_i$, $c_{i+3}t_i$
(3.4) $c_{i-3}c_{i-2}t_i$, $c_{i-2}c_{i-1}t_i$, $c_{i-1}c_it_i$, $c_ic_{i+1}t_i$ $c_{i+1}c_{i+2}t_i$, $c_{i+2}c_{i+3}t_i$, $c_{i-2}c_it_i$, $c_ic_{i+2}t_i$
(3.5) $c_is_it_i$
(3.6) $c_it_{i-1}t_i$
(3.7) $s_{i-1}t_{i-1}s_i$, $t_{i-1}s_it_i$

Table 6: Word segmentation results on Fourth SIGHAN Bakeoff CTB corpus

Rank	F1	Description
1/26	95.89*	official best, using extra unlabeled data (Zhao and Kit, 2008)
2/26	95.33	official second
3/26	95.17	official third
4/26	95.12	segmentor in pipeline system

Table 7: POS results on Fourth SIGHAN Bakeoff CTB corpus

Rank	Accuracy	Description
1/7	94.29	POS tagger in pipeline system
2/7	94.28	official best
3/7	94.01	official second
4/7	93.24	official third

the sentence, $c_j(w_i), j > 0$ denotes the j^{th} Chinese character of word w_i, $c_j(w_i), j < 0$ denotes the j^{th} last Chinese character, $l(w_i)$ denotes the word length of w_i. We compare our POS tagger with other top systems on Bakeoff CTB POS corpus where sentences are perfectly segmented into words, our POS tagger achieved 94.29 accuracy, which is the best of 7 official runs. Comparison results are shown in Table 7.

For reranking method, we varied candidate numbers n among $n \in \{10, 20, 50, 100\}$. For hybrid CRFs, we use the same segmentation label set as the segmentor in pipeline. Feature templates are listed in Table 5. Experimental results are shown in Figure 3. The gain of hybrid CRFs over the baseline pipeline model is 0.48 points in F-score, about 3 times higher than 100-best reranking approach which achieves 0.13 points improvement. Though larger candidate number can achieve higher performance, such improvement becomes trivial for $n > 20$.

Table 8 shows the comparison between our work and other relevant work. Notice that, such comparison is indirect due to different data sets and re-

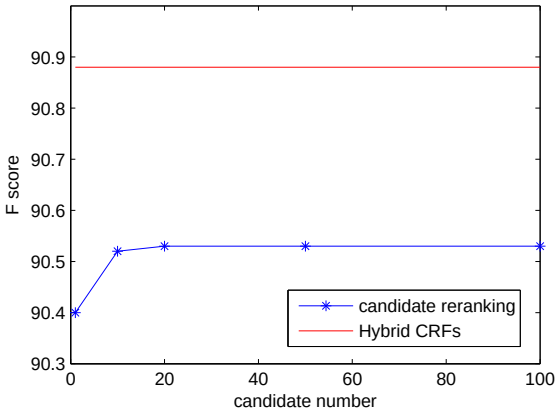

Figure 3: Results for Chinese word segmentation and POS tagging task, Hybrid CRFs significantly outperform 100-Best Reranking (McNemar's test; $p < 0.01$)

Table 8: Comparison of word segmentation and POS tagging, such comparison is indirect due to different data sets and resources.

Model	F1
Pipeline (ours)	90.40
100-Best Reranking (ours)	90.53
Hybrid CRFs (ours)	90.88
Pipeline (Shi and Wang, 2007)	91.67
20-Best Reranking (Shi and Wang, 2007)	91.86
Pipeline (Zhang and Clark, 2008)	90.33
Joint Perceptron (Zhang and Clark, 2008)	91.34
Perceptron Only (Jiang et al., 2008)	92.5
Cascaded Linear (Jiang et al., 2008)	93.4

sources. One common conclusion is that joint models generally outperform pipeline models.

5 Conclusion

We introduced a framework to integrate graph structures for segmentation and tagging subtasks into one using virtual nodes, and performs joint training and decoding in the factorized state space. Our approach does not suffer from error propagation, and guards against violations of those hard-constraints imposed by segmentation subtask. Experiments on shallow parsing and Chinese word segmentation tasks demonstrate our technique.

6 Acknowledgements

The author wishes to thank the anonymous reviewers for their helpful comments. This work was partially funded by 973 Program (2010CB327906), The National High Technology Research and Development Program of China (2009AA01A346), Shanghai Leading Academic Discipline Project (B114), Doctoral Fund of Ministry of Education of China (200802460066), National Natural Science Funds for Distinguished Young Scholar of China (61003092), and Shanghai Science and Technology Development Funds (08511500302).

References

R. Ando and T. Zhang. 2005. A high-performance semi-supervised learning method for text chunking. In *Proceedings of ACL*, pages 1–9.

Razvan C. Bunescu. 2008. Learning with probabilistic features for improved pipeline models. In *Proceedings of EMNLP*, Waikiki, Honolulu, Hawaii.

X Carreras and L Marquez. 2003. Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of RANLP*.

Changning Huang and Hai Zhao. 2007. Chinese word segmentation: A decade review. *Journal of Chinese Information Processing*, 21:8–19.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lu. 2008. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL*, Columbus, Ohio, USA.

Guangjin Jin and Xiao Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Proceedings of Sixth SIGHAN Workshop on Chinese Language Processing*, India.

Junichi Kazama and Kentaro Torisawa. 2007. A new perceptron algorithm for sequence labeling with non-local features. In *Proceedings of EMNLP*, pages 315–324, Prague, June.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of NAACL*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Ruy L. Milidiu, Cicero Nogueira dos Santos, and Julio C. Duarte. 2008. Phrase chunking using entropy guided transformation learning. In *Proceedings of ACL*, pages 647–655.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-ofspeech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of EMNLP*.

J. Nocedal and S. J. Wright. 1999. *Numerical Optimization*. Springer.

Yanxin Shi and Mengqiu Wang. 2007. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of IJCAI*, pages 1707–1712, Hyderabad, India.

C. Sutton, K. Rohanimanesh, and A. McCallum. 2004. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of ICML*.

Jun Suzuki and Hideki Isozaki. 2008. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of ACL*, pages 665–673.

Jun Suzuki, Akinori Fujino, and Hideki Isozaki. 2007. Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *Proceedings of EMNLP*, Prague.

Yu-Chieh Wu, Chia-Hui Chang, and Yue-Shi Lee. 2006. A general and multi-lingual phrase chunking model based on masking method. In *Proceedings of Intelligent Text Processing and Computational Linguistics*, pages 144–155.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proceedings of ACL*, Columbus, Ohio, USA.

T. Zhang, F. Damerau, and D. Johnson. 2002. Text chunking based on a generalization of winnow. machine learning research. *Machine Learning Research*, 2:615–637.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging forword segmentation and named entity recognition. In *Proceedings of Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111.