labeling can also be done directly (Sarawagi and Cohen, 2004). However, Semi-CRF just models label dependency, and it cannot capture more correlations between adjacent chunks, as is done in our approach. The limitation of Semi-CRF leads to its relatively low performance.

3 Problem Formulation

3.1 Chunk Types

Unlike English chunking, there is not a benchmarking corpus for Chinese chunking. We follow the studies in (Chen et al. 2006) so that a more direct comparison with state-of-the-art systems for Chinese chunking would be possible. There are 12 types of chunks: ADJP, ADVP, CLP, DNP, DP, DVP, LCP, LST, NP, PP, QP and VP in the chunking corpus (Xue et al., 2000). The training and test corpus can be extracted from CTB4 with a public tool, as depicted in (Chen et al. 2006).

3.2 Sequence Labeling Approaches to Phrase Chunking

The standard approach to phrase chunking is to use tagging techniques with a BIO tag set. Words in the input text are tagged with one of B for the beginning of a contiguous segment, I for the inside of a contiguous segment, or O for outside a segment. For instance, the sentence (word segmented and POS tagged) "他/NR(He) 到达/VV(reached) 北京/NR(Beijing) 机场/NN(airport) 。/PU" will be tagged as follows:

Example 1:
S1: [NP 他][VP 到达][NP 北京/机场][O 。]
S2: 他/B-NP 到达/B-VP 北京/B-NP 机场/I-NP 。/O

Here S1 denotes that the sentence is tagged with chunk types, and S2 denotes that the sentence is tagged with chunk tags based on the BIO-based model. With the data representation like the S2, the problem of phrase chunking can be reduced to a sequence labeling task.

3.3 Phrase Chunking via a Joint Segmentation and Labeling Approach

To tackle the problems with the sequence labeling approaches to phrase chunking, we formulate it as a joint problem, which maps a Chinese sentence *x* with segmented words and POS tags to an output *y* with tagged chunk types, like the S1 in Example 1. The joint model considers all possible chunk boundaries and corresponding chunk types in the sentence, and chooses the overall best output. This kind of parser reads the input sentences from left to right, predicts whether current segment of continuous words is some type of chunk. After one chunk is found, parser move on and search for next possible chunk.

Given a sentence *x*, let *y* denote an output tagged with chunk types, and GEN a function that enumerates a set of segmentation and labeling candidates GEN(*x*) for *x*. A parser is to solve the following "argmax" problem:

$$\begin{aligned}\hat{y} &= \arg\max_{y \in GEN(x)} w^T \cdot \Phi(y) \\ &= \arg\max_{y \in GEN(x)} w^T \cdot \sum_{i=1}^{|y|} \phi(y_{[1..i]})\end{aligned} \qquad (1)$$

where Φ and ϕ are global and local feature maps and w is the parameter vector to learn. The inner product $w^T \cdot \phi(y_{[1..i]})$ can be seen as the confidence score of whether y_i is a chunk. The parser takes into account confidence score of each chunk, by using the sum of local scores as its criteria. Markov assumption is necessary for computation, so ϕ is usually defined on a limited history.

The main advantage of the joint segmentation and labeling approach to phrase chunking is to allow for integrating both the internal structural features and the correlations between the adjacent chunks for prediction. The two basic components of our model are decoding and learning algorithms, which are described in the following sections.

4 Decoding

The inference technique is one of the most important components for a joint segmentation and labeling model. In this section, we propose a dynamic programming algorithm with pruning to efficiently produce the optimal output.

4.1 Algorithm Description

Given an input sentence *x*, the decoding algorithm searches for the highest-scored output with recognized chunks. The search space of combined candidates in the joint segmentation and labeling task is very large, which is an exponential growth