final weight vector w is the average of the weight vectors after each iteration.

5.2 Loss Function

For the joint segmentation and labeling task, there are two alternative loss functions: 0-1 loss and F1 loss. 0-1 loss gives credit only when the entire output sequence is correct: there is no notion of partially correct solutions. The most common loss function for joint segmentation and labeling problems is F1 measure over chunks. This is the geometric mean of precision and recall over the (properly-labeled) chunk identification task, defined as follows.

$$L^F(y,\hat{y}) \triangleq 1 - \frac{2|y \cap y'|}{|y| + |y'|} \quad (3)$$

where the cardinality of y is simply the number of chunks identified. The cardinality of the intersection is the number of chunks in common. As can be seen in the definition, one is penalized both for identifying too many chunks (penalty in the denominator) and for identifying too few (penalty in the numerator).

In our experiments, we will compare the performance of the systems with different loss functions.

5.3 Features

Table 1 shows the feature templates for the joint segmentation and labeling model. In the row for feature templates, c, t, w and p are used to represent a chunk, a chunk type, a word and a POS tag, respectively. And c_0 and c_{-1} represent the current chunk and the previous chunk respectively. Similarly, w_{-1}, w_0 and w_1 represent the previous word, the current word and the next word, respectively.

Although it is slightly less natural to do so, part of the features used in the sequence labeling models can also be represented in our approach. Therefore the features employed in our model can be divided into three types: the features similar to those used in the sequence labeling models (called SL-type features), the features describing internal structure of a chunk (called Internal-type features), and the features capturing the correlations between the adjacent chunks (called Correlation-type features).

Firstly, some features associated with a single label (here refers to label "B" and "I") used in the sequence labeling models are also represented in our model. In Table 1, templates 1-4 are SL-type features, where *label*(w) denotes the label indicating the position of the word w in the current chunk; *len*(c) denotes the length of chunk c. For example, given an NP chunk "北京(Beijing) 机场(Airport)", which includes two words, the value of *label*("北京") is "B" and the value of *label*("机场") is "I". *Bigram*(w) denotes the word bigrams formed by combining the word to the left of w and the one to the right of w. And the same meaning is for *biPOS*(w). Template *specitermMatch*(c) is used to check the punctuation matching within chunk c for the special terms, as illustrated in *section* 1.

Secondly, in our model, we have a chance to treat the chunk candidate as a whole during decoding, which means that we can employ more expressive features in our model than in the sequence labeling models. In Table 1, templates 5-13 concern the Internal-type features, where *start_word*(c) and *end_word*(c) represent the first word and the last word of chunk c, respectively. Similarly, *start_POS*(c) and *end_POS*(c) represent the POS tags associated with the first word and the last word of chunk c, respectively. These features aim at expressing the formation patterns of the current chunk with respect to words and POS tags. Template *internalWords*(c) denotes the concatenation of words in chunk c, while *internalPOSs*(c) denotes the sequence of POS tags in chunk c using regular expression-like form, as illustrated in *section* 1.

Finally, in Table 1, templates 14-28 concern the Correlation-type features, where *head*(c) denotes the headword extracted from chunk c, and *headPOS*(c) denotes the POS tag associated with the headword in chunk c. These features take into account various aspects of correlations between adjacent chunks. For example, we extracted the headwords located in adjacent chunks to form headword bigrams to express semantic dependency between adjacent chunks. To find the headword within every chunk, we referred to the head-finding rules from (Bikel, 2004), and made a simple modification to them. For instance, the head-finding rule for NP in (Bikel, 2004) is as follows:

(NP (r NP NN NT NR QP) (r))

Since the phrases are non-overlapping in our task, we simply remove the overlapping phrase tags NP