2 The approach

As mentioned above, a two-stage cascade routine for parsing is used in this work for the shared task. The two stages, are conducted one by one independently. The PCFG parsing is performed by Berkeley parser (Petrov et al., 2006);(Petrov and Klein, 2007). For the head constituent recognition, according to (Zhou, 2007), the outside and inside syntactic constituents for a single syntactic production can provide enough information to locate the head constituent(s). Thus, to design a proper framework to incorporate the syntactic features could help to label the head constituents correctly. We propose a sequence labeling approach as CRF learning and tagging. The following subsections present the details of our approach.

2.1 PCFG parsing

The Berkeley parser is used as our syntactic constituent parsing tool for the first stage. It is a PCFG parser, written in Java, that can be trained on standardized collection of hand parsed sentences. It uses EM training to estimate the probabilities involved in the context-free grammars in use, usually begining with the barest possible initial structure and then refining the grammars via a hierarchical coarse-to-fine scheme until the predicted syntactic structures fit the training data well enough to a certain degree. Specifically, the parser provides horizontal and vertical markovization[1] options to support integrating more syntactic information. The parser is originally designed for parsing Penn Treebank (PTB) sentence. Although it provides Chinese grammar package, it is still not capable of dealing with TCT sentences. Thus we need to have a few modifications on it for our task, e.g., add multiple root support based on PTB parsing configuration for TCT sentences since the TCT roots can be *-adjp*, *-vp* or *-np* and so on.

2.2 Head recognition

To recognize the head constituents(s), an extra step is needed since ordinary parsing could not provide a straight forward way for this. Consider that head constituents are always determined by their syntactic symbol and their neighbors, whose order and relations strongly affects the head labeling. Like chunking (Sha and Pereira, 2003);(Tsuruoka et al., 2009), it is natural to apply a sequence labeling strategy to tackle this problem. We adopt the linear-chain CRF (Lafferty et al., 2001), the most successful sequence

Table 1: Feature templates used in head constituent recognition

Description	Template
Constituent Unigrams	$c_{-2}, c_{-1}, c_0, c_{+1}, c_{+2}$
Constituent Bigrams	$c_{-2}c_{-1}, c_{-1}c_0, c_0c_{+1}, c_{+1}c_{+2}$
Constituent Trigrams	$c_{-3}c_{-2}c_{-1}, c_{-2}c_{-1}c_0, c_{-1}c_0c_{+1}, c_0c_{+1}c_{+2}, c_{+1}c_{+2}c_{+3}$
Type Unigrams	t_{-1}, t_0, t_{+1}
Type Bigrams	$t_{-1}t_0, t_0t_{+1}$
Type Trigram	$t_{-1}t_0t_{+1}$
Combined Bigram	c_0t_0

Table 2: Of cial scores for CIPS-Parseval parsing task

	Score
Without-head match F1	0.8364
Partial-head match F1	0.7914
Complete-head match F1	0.7600*

* Our calculation using the of cial score program gives 0.7723 for complete-head match F1, with 0.7737 for precision and 0.7708 for recall rate.

We split one fourth of the data from the part with syntactic information as our development set. On the other hand, there are 16210 clauses in released testing data. We only focus on those lines of two or more words. 7939 clauses are then extracted as the nal parsing set. Because of the integrated annotation, we need to remove all of the head information from the training sentences in order to properly train the Berkeley parser. Once a test sentence is parsed, each production on the whole syntax tree is analyzed and then the tagged head position is attached to the parent constituent symbol.

3.2 Results

There is only one submission for our parsing task, whose of cial scores are shown in Table 2. The F1 score is combined via geometry average of precision and recall rate of accurate syntactic constituents. It shows the effectiveness of our head constituent recognition.

In further experiments, we improve the output of the PCFG parser by adjusting some parameters for the training process in order to avoid over tting, which was found after our submission, and adding the 1st order horizontal markovization to incorporate more information. Consequently, the head recognition get a promising improvement, as shown in Table 3.

In addition, since the head constituent recognition works well as long as there is a parsed syntax tree for a given sentence, we could compare our tagging performance objectively against the parsing outputs by other methods. For an of cial submitted result, generated by a shift-reduce approach, provided by another CIPS-Parseval participant, we work on it just as we did on our development data, by removing

Table 3: Improved performance on CIPS-Parseval parsing task

	Score	Increment
Without-head match F1	0.8582	+2.60%
Complete-head match F1	0.7937	+2.77%*

* It is the increment from our recalculated complete-head match F1 score, not the of cially released one.