brackets would need to be determined for the unlabeled sentences together with the latent annotations, this would increase the running time from linear in the number of expansion rules to cubic in the length of the sentence.

Another important decision is how to weight the gold standard and automatically labeled data when training a new parser model. Errors in the automatically labeled data could limit the accuracy of the self-trained model, especially when there is a much greater quantity of automatically labeled data than the gold standard training data. To balance the gold standard and automatically labeled data, one could duplicate the treebank data to match the size of the automatically labeled data; however, the training of the PCFG-LA parser would result in redundant applications of EM computations over the same data, increasing the cost of training. Instead we weight the posterior probabilities computed for the gold and automatically labeled data, so that they contribute equally to the resulting grammar. Our preliminary experiments show that balanced weighting is effective, especially for Chinese (about 0.4% absolute improvement) where the automatic parse trees have a relatively lower accuracy.

The training procedure of the PCFG-LA parser gradually introduces more latent annotations during each split-merge stage, and the self-labeled data can be introduced at any of these stages. Introduction of the self-labeled data in later stages, after some important annotations are learned from the treebank, could result in more effective learning. We have found that a middle stage introduction (after 3 split-merge iterations) of the automatically labeled data has an effect similar to balancing the weights of the gold and automatically labeled trees, possibly due to the fact that both methods place greater trust in the former than the latter. In this study, we introduce the automatically labeled data at the outset and weight it equally with the gold treebank training data in order to focus our experiments to support a deeper analysis.

4 Experimental Setup

For the English experiments, sections from the WSJ Penn Treebank are used as labeled training data: section 2-19 for training, section 22 for development, and section 23 as the test set. We also used 210k[4] sentences of unlabeled news articles in the BLLIP corpus for English self-training.

For the Chinese experiments, the Penn Chinese Treebank 6.0 (CTB6) (Xue et al., 2005) is used as labeled data. CTB6 includes both news articles and transcripts of broadcast news. We partitioned the news articles into train/development/test sets following Huang et al. (2007). The broadcast news section is added to the training data because it shares many of the characteristics of newswire text (e.g., fully punctuated, contains nonverbal expressions such as numbers and symbols). In addition, 210k sentences of unlabeled Chinese news articles are used for self-training. Since the Chinese parsers in our experiments require word-segmented sentences as input, the unlabeled sentences need to be word-segmented first. As shown in (Harper and Huang, 2009), the accuracy of automatic word segmentation has a great impact on Chinese parsing performance. We chose to use the Stanford segmenter (Chang et al., 2008) in our experiments because it is consistent with the treebank segmentation and provides the best performance among the segmenters that were tested. To minimize the discrepancy between the self-training data and the treebank data, we normalize both CTB6 and the self-training data using UW Decatur (Zhang and Kahn, 2008) text normalization.

Table 1 summarizes the data set sizes used in our experiments. We used slightly modified versions of the treebanks; empty nodes and nonterminal-yield unary rules[5], e.g., NP→VP, are deleted using tsurgeon (Levy and Andrew, 2006).

	Train	**Dev**	**Test**	**Unlabeled**
English	39.8k (950.0k)	1.7k (40.1k)	2.4k (56.7k)	210k (5,082.1k)
Chinese	24.4k (678.8k)	1.9k (51.2k)	2.0k (52.9k)	210k (6,254.9k)

Table 1: The number of sentences (and words in parentheses) in our experiments.

We trained parsers on 20%, 40%, 60%, 80%, and 100% of the treebank training data to evaluate

[4] This amount was constrained based on both CPU and memory. We plan to investigate cloud computing to exploit more unlabeled data.

[5] As nonterminal-yield unary rules are less likely to be posited by a statistical parser, it is common for parsers trained on the standard Chinese treebank to have substantially higher precision than recall. This gap between bracket recall and precision is alleviated without loss of parse accuracy by deleting the nonterminal-yield unary rules. This modification similarly benefits both parsers we study here.