added to the seed set. In the next iteration, we retrain the classifier on the augmented labeled data, apply it to the unlabeled data, and add to the labeled data those instances whose predicted confidence is above the current threshold. If none of the instances has a predicted confidence above the current threshold, we reduce the threshold by 0.1. (For instance, if the original thresholds are 1 and -1, they will be changed to 0.9 and -0.9.) We then repeat the above procedure until the thresholds reach 0.5 and -0.5. [11] Finally, we apply the resulting bootstrapped classifier to label all of the unlabeled words that have a corpus frequency of at least five, using a threshold of 0.

In our implementation of the self-training algorithm, rather than train a multi-class classifier in each bootstrapping iteration, we train pairwise classifiers (recall that for English, 45 classifiers are formed from 10 POS tags) using the morphological and distributional features described in the previous section. Again, since we employ distributional features, we apply the 45 pairwise classifiers only to the distributionally reliable words (i.e., words with corpus frequency at least 5). To classify an unlabeled word *w*, we apply the 45 pairwise classifiers to independently assign a label to *w*.[12] We then count how many times *w* is tagged with each of the ten POS tags. If there is a POS tag whose count is nine and all of these nine votes are associated with confidence that exceeds the current threshold, then we add *w* to the labeled data together with its assigned tag.

## 7 Evaluation

### 7.1 Experimental Setup

**Corpora.** Recall that our bootstrapping algorithm assumes as input an unannotated corpus from which we (1) extract our *vocabulary* (i.e., the set of words to be labeled) and (2) collect the statistics needed in morphological and distributional clustering. We use as our English corpus the Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus et al., 1993). Our Bengali corpus is composed of five years of articles taken from the Bengali newspaper *Prothom Alo*.

**Vocabulary creation**. To extract our English vocabulary, we pre-processed each document in the WSJ corpus by first tokenizing them and then removing the most frequent 500 words (as they are mostly closed class words), capitalized words, punctuations, numbers, and unwanted character sequences (e.g., "***"). The resulting English vocabulary consists of approximately 35K words. We applied similar pre-processing steps to the Prothom Alo articles to generate our Bengali vocabulary, which consists of 80K words.

**Test set preparation**. Our English test set is composed of the 25K words in the vocabulary that appear at least five times in the WSJ corpus. The gold-standard POS tags for each word *w* are derived automatically from the parse trees in which *w* appears. To create the Bengali test set, we randomly chose 5K words from the vocabulary that appear at least five times in Prothom Alo. Each word in the test set was then labeled with its POS tags by two of our linguists.

**Evaluation metric**. Following Schütze (1995), we report performance in terms of recall, precision, and F1. Recall is the percentage of POS tags correctly proposed, precision is the percentage of POS tags proposed that are correct, and F1 is simply the harmonic mean of recall and precision. To exemplify, suppose the correct tagset for "crowd" is {NN, VB}; if our system outputs {VB, JJ, RB}, then recall is 50%, precision is 33%, and F1 is 40%. Importantly, all of our results will be reported on *word types*. This prevents the frequently occurring words from having a higher influence on the results than their infrequent counterparts.

### 7.2 Results and Discussion

**The baseline system.** We use as our baseline system one of the best existing unsupervised POS induction algorithms (Clark, 2003). More specifically, we downloaded from Clark's website[13] the code that implements a set of POS induction algorithms he proposed. Among these implementations, we chose *cluster_neyessenmorph*, which combines morphological and distributional infor-

[11] We decided to stop the bootstrapping procedure at thresholds of 0.5 and -0.5, because the more bootstrapping iterations we use, the lower are the quality of the bootstrapped data as well as the accuracy of the bootstrapped classifier.

[12] As in purification, each pairwise classifier is implemented as a set of 10 classifiers, each of which is trained on an equal number of instances from both classes. Testing also proceeds as before: the label of an instance is derived from the average of the confidence values returned by the 10 classifiers, and the confidence value associated with the label is just the average of the 10 confidence values.

[13] http://www.cs.rhul.ac.uk/home/alexc/