SEMI-SUPERVISED LEARNING FOR PART-OF-SPEECH TAGGING OF MANDARIN TRANSCRIBED SPEECH

Wen Wang[1], *Zhongqiang Huang*[2], *Mary Harper*[2,3]

[1]SRI International, Menlo Park, CA 94025, USA
[2]Purdue University, West Lafayette, IN 47907, USA
[3]University Of Maryland, MD 20742, USA
wwang@speech.sri.com, {zqhuang,harper}@purdue.edu

ABSTRACT

In this paper, we investigate bootstrapping part-of-speech (POS) taggers for Mandarin broadcast news (BN) transcripts using co-training, by iteratively retraining two competitive POS taggers from a small set of labeled training data and a large set of unlabeled data. We compare co-training with self-training and our results show that the performance using co-training is significantly better than that from self-training and these semi-supervised learning methods significantly improve tagging accuracy over training only on the small labeled seed corpus. We also investigate a variety of example selection approaches for co-training and find that the computationally expensive, agreement-based selection approach and a more efficient selection approach based on maximizing training utility produce comparable tagging performance from resulting POS taggers. By applying co-training, we are able to build effective POS taggers for Mandarin transcribed speech with the tagging accuracy comparable to that obtained on newswire text.

Index Terms— POS tagging, Co-training, Self-training, Mandarin speech recognition, Active learning

1. INTRODUCTION

Part-of-speech (POS) tagging is a prerequisite for many advanced natural language processing tasks, for example, name entity recognition, parsing, and sentence boundary detection. Much effort has been expended to develop high-quality POS taggers, but the majority has been applied only on newswire text, largely because of the availability of labeled newswire training data. However, many tasks require applying POS taggers to a new domain with little or no manual annotations, for example, transcribed speech. Clark et al. [1] explored using co-training to bootstrap POS taggers using a small in-domain labeled seed corpus and a large set of unlabeled data. Although their work shows the effectiveness of co-training for improving POS tagging, their investigations were conducted within the newswire text genre. Mieskes and Strube [2] used four POS taggers originally trained on newswire text to tag a corpus of transcribed multiparty spoken dialogues. However, their work was focused on using manual corrections on the tags assigned by the taggers to evaluate the taggers and retrain them. To our knowledge, there has been little effort on exploring methods for applying semi-supervised learning methods like co-training to POS tagging on transcribed speech and evaluations of the tagging performance. Our work is aimed at helping to generate rich syntactic and semantic annotations for improving Mandarin broadcast news (BN) ASR and machine translation (MT) performance. To support building language models and translation models employing POS information for ASR and MT, we have been focusing on improving POS tagging on Mandarin BN transcripts. In the rest of the paper, Section 2 describes the co-training and self-training algorithms, as well as example selection approaches used in co-training. Section 3 briefly describes the two POS taggers used in this work. Experimental results, discussion, and conclusions appear in Section 4.

2. CO-TRAINING AND SELF-TRAINING

2.1. General co-training algorithm

Co-training was first introduced by Blum and Mitchell [3] as a weakly supervised method. It can be used for bootstrapping a model from a seed corpus of labeled examples, which is typically quite small, augmented with a much larger amount of unlabeled examples, by exploiting redundancy among multiple statistical models that generate different *views* of the data. Blum and Mitchell [3] showed that co-training is *p*robably approximately correct (PAC) learnable when the two views are individually sufficient for classification and conditionally independent given the class. Abney [4] proved that a weaker independence assumption on the multiple classifiers than Blum and Mitchell's quite restrictive assumption could still allow co-training to work well. There has been much effort on investigating the efficacy of co-training in different domains and applications [4, 5]. The co-training algorithm developed by Pierce and Cardie [5] is presented in Algorithm 1 in this paper.

Input: S is a seed set of labeled data.
L_{h_1} is labeled training data for h_1.
L_{h_2} is labeled training data for h_2.
U is the unlabeled data set.
C is the cache holding a small subset of U.
$L_{h_1} \longleftarrow S$
$L_{h_2} \longleftarrow S$
Train classifier h_1 on L_{h_1}
Train classifier h_2 on L_{h_2}
repeat
  Randomly partition U into C where $|C| = u$ and U'
  Apply h_1, h_2 to assign labels for all examples in C
  Select examples labeled by h_1 and add to L_{h_2}
  Train h_2 on L_{h_2}
  Select examples labeled by h_2 and add to L_{h_1}
  Train h_1 on L_{h_1}
  $U \longleftarrow U'$
until U *is empty*

Algorithm 1: General co-training algorithm.

2.2. Example selection approaches for co-training

In Algorithm 1, when calling the classifier that provides additional training data for the opposite classifier the *teacher* and the opposite classifier the *student*, since the labeled output from both classifiers h_1 and h_2 is noisy, an important question is which newly labeled examples from the teacher should be added to the training data pool of the student. This issue of example selection plays an important role in the learning rate of co-training and the performance of resulting classifiers. In this paper, we investigate four example selection approaches. The first is *naive co-training*, which simply adds all examples in the cache labeled by the teacher to the training data pool of the student. The single parameter that needs to be optimized (on a held-out set) for this example selection approach on the classification accuracy is the cache size, u.

The second approach, *agreement-based co-training* [6, 1], is to select the subset of the labeled cache that maximizes the agreement of the two classifiers on *unlabeled* data. The pseudo-code for agreement-based example selection algorithm is presented in Algorithm 2. The *student* classifier is the one being retrained and the *teacher* classifier is the one remaining static. During the *agreement-based* selection procedure, we repeatedly sample from all possible subsets of the cache, by first choosing the size of the subset and then randomly choosing examples from the labeled cache based on the size. In this algorithm, if h_2 is trained on the updated L_{h_2} after adding output from h_1, then the most recent version of h_1 is used to measure agreement and vice versa. Hence, this approach aims to improve the performance of the two classifiers alternatively, instead of simultaneously. Note that the *agreement rate on* U, denoted A, is the number of times each token in the unlabeled set U is assigned the same label by both classifiers h_1 and h_2.

Input: C is a cache of examples labeled by the *teacher* classifier.
U is a set of examples, used for measuring agreement.
$c_{max} \longleftarrow \emptyset$
$A_{max} \longleftarrow 0$
iter=1
repeat
  Randomly sample $c \subseteq C$
  Retrain *student* classifier using c as additional data
  A=the new agreement rate on U
  if $A > A_{max}$ **then**
    $A_{max} \longleftarrow A$
    $c_{max} \longleftarrow c$
  end
  iter++
until *iter=n*
return c_{max}

Algorithm 2: Agreement-based example selection approach [1].

Besides naive co-training and the agreement-based example selection approach, we proposed two different methods. One is to select the top n examples with highest scores (based on a scoring function) when labeled by the teacher to add to the training pool of the student. This approach has been employed in many co-training applications. We denote it *max-score*. The underlying intuition is to select examples that are reliably labeled by the teacher for the student. However, Hwa's work on active learning [7] has shown that accurately labeled examples may not always been useful for improving a classifier's performance. Instead, examples with high *training utility* could satisfy this request. To combine accuracy and training utility, we defined another example selection criterion, which selects examples with scores within the m percent of top high-scoring labeled examples by the teacher and within the n percent of bottom low-scoring labeled examples by the student. We denote it *max-t-min-s*. The intuition for this approach is that the newly labeled data should not only be reliably labeled by the teacher but also should be as useful and compensatory as possible for the student. During empirical evaluations of these example selection methods, control parameters, e.g., n and m, in these approaches, are optimized on a heldout data set with respect to the performance of resulting classifiers after co-training.

2.3. Self-training algorithm

We also compare the performance of co-training to self-training. There are a variety of definitions of self-training in the literature and we adopted that of Nigam and Ghani [8]. The self-training algorithm is shown in Algorithm 3. Self-training in this work simply adds all examples in the labeled cache to the training pool in each iteration.

Input: S is a seed set of labeled data.
L_{h_1} is labeled training data for h_1.
U is the unlabeled data set.
C is the cache holding a small subset of U.
$L_{h_1} \longleftarrow S$
Train classifier h_1 on L_{h_1}
repeat
  Randomly partition U into C where $|C| = u$ and U'
  Apply h_1 to assign labels for all examples in C
  Select examples labeled by h_1 and add to L_{h_1}
  Train h_1 on L_{h_1}
  $U \longleftarrow U'$
until U *is empty*

Algorithm 3: General self-training algorithm.

3. TWO POS TAGGERS

The two POS taggers we use in this paper are a Hidden Markov Model (HMM) tagger and a maximum-entropy (ME) tagger.

3.1. H1: HMM tagger

The HMM tagger used for this effort is a second-order HMM tagger initially developed by Thede and Harper [9]. This second-order HMM tagger, initially designed for English, used trigram transition probability estimations, $P(t_i|t_{i-2}t_{i-1})$, and trigram emission probability estimations, $P(w_i|t_{i-1}t_i)$. For estimating emission probabilities for unknown words (i.e., a word that does not appear in the training data), a weighted sum of $P(s_i^k|t_{i-1}t_i)$ was used as an approximation, where s_i^k is the k-th suffix of word w_i (the first suffix of word w_i is its last character). The interpolation weights for smoothing transition and emission probabilities were estimated using a log-based function introduced in [9]. In this work, we achieved improvement on Chinese newswire POS tagging accuracy after refining this model by: replacing $P(w_i|t_{i-1}t_i)$ with context-enriched $P(w_i|t_{i-1}t_i)^{\frac{1}{2}} \times P(w_{i-2}|t_{i-2}t_{i-1})^{\frac{1}{2}}$ for both known and unknown words; also, for unknown words, replacing $P(w_i|t_{i-1}t_i)$ by the geometric mean of $P(c_i^k|t_{i-1}t_i)$, where c_i^k is the k-th character of

the word w_i, for all of the characters of the word w_i (note that $P(w_{i-2}|t_{i-2}t_{i-1})$ is calculated similarly). In Chinese, there is no inflection and derivation of words as in English. However, the last few characters of a Chinese word may still provide hints for propernouns, nouns, and verbs. We empirically compared the use of last few characters to all characters in a word for unknown word emission probability estimation and found that using all characters produced the best tagging accuracy and the use of last few characters also provides smaller but consistent improvement on the POS tagging accuracy.

3.2. H2: ME tagger

We built a maximum-entropy POS tagger that uses features from Ratnaparkhi's ME tagger [10] adapted for Mandarin POS tagging. In our ME tagger, the context used for predicting the POS tag of a word w_i among a sentence $w_1 \dots w_n$ with tags $t_1 \dots t_n$ is defined as $h_i = \{w_{i-2}, w_{i-1}, t_{i-2}, t_{i-1}, w_i, w_{i+1}, w_{i+2}\}$. We modified the features for rare words in [10] by collecting a set of the most frequent last one character suffixes (i.e., the last character of a word) and last two character suffixes and created features for rare words with respect to including members in this set of rare-word suffixes, denoted $\mathcal{S}$. The adapted features on the current history h_i are presented in Table 1. Different from the HMM tagger which is a joint model, the ME tagger uses a conditional model. Also, arbitrary features derived from the context can be easily added to the ME tagger, for example, word identities from either side of the target word. Besides this theoretical comparison, in Section 4, we will empirically examine whether the two taggers are sufficiently different based on their tagging output.

Table 1. Features on the current history h_i used in the ME tagger. Given the current history, a feature is an indicator function of certain variables on the history. X, Y, and T in the table are instantiations of word and tag identities that can be automatically collected from the training data.

Condition	Features
w_i is not rare	$w_i = X$ and $t_i = T$
w_i is rare	w_i contains a suffix in $\mathcal{S}$ and $t_i = T$
$\forall w_i$	$t_{i-1} = X$ and $t_i = T$
	$t_{i-2}t_{i-1} = XY$ and $t_i = T$
	$w_{i-1} = X$ and $t_i = T$
	$w_{i-2} = X$ and $t_i = T$
	$w_{i+1} = X$ and $t_i = T$
	$w_{i+2} = X$ and $t_i = T$

4. EXPERIMENTS

4.1. Data

In this paper, the data set of labeled examples is a text corpus with each sentence annotated with POS tags. For Mandarin POS tagging, the most recently released Chinese Penn Treebank 5.2 (denoted **CTB**, released by LDC) contains around 500K words, 800K characters, 18K sentences, and 900 data files, including articles from the Xinhua news agency, Information Services Department of HKSAR, and Sinorama magazine (Taiwan). The format of CTB is similar to the Penn English Treebank and it was carefully annotated. In this work, we first compare the performance of the two POS taggers on the CTB corpus, and for self-training and co-training, we always include CTB in the initial training pool for each tagger. Among the 33 POS tags used in CTB[1], we discriminated punctuation (POS tag as PU) by creating new POS tags for each distinct word type tagged as PU in CTB (e.g., PU-?). In total, we used 92 POS tags.

The creation of the small seed corpus and evaluation test set for Mandarin BN transcripts is a non-trivial procedure. Since basically there was no hand-annotated POS-tagged Mandarin BN corpus available, we created one for this work using the following procedure: first, we selected the set of BN transcripts from the DARPA GALE program[2] Mandarin ASR/MT development test set, where we used the four dev show transcripts from the GALE Year 1 BN audio release. This text set, which includes about 17K words (about 79K characters), is our manual annotation corpus. Second, the HMM tagger was used to automatically assign POS tags to this corpus. Then, three annotators (all are native speakers of Mandarin with expertise on POS tagging) proof-read the pre-tagged corpus and fixed tagging errors. Since automatic word segmentation [11] was conducted on this corpus before the pre-tagging step, we also fixed transcription errors and word segmentation errors. Each annotator independently conducted manual annotations on the corpus and then an Emacs annotation tool adopted from LDC was used to highlight the differences between annotators' decisions. The inter-annotator agreement from the first round was sufficiently high (estimated $\kappa > 0.7$)[3], demonstrating that the automatically assigned tags can be reliably corrected by annotators. Differences between annotators were discussed and all disagreements appearing in the first round were resolved in the end. We extracted three disjoint sets from this tagged corpus: the first served as the small Mandarin BN seed corpus (400 sentences, 8K words), the second was used as the POS-eval test set (400 sentences, 6K words), and the rest set (163 sentences, 3K words) was used as a heldout set for optimizing parameters for co-training, for example, n and m for the *max-t-min-s* approach.

The large set of unlabeled data includes the following sources: the HUB4 1997 Mandarin BN acoustic transcripts, the LDC Chinese TDT2, TDT3, TDT4 corpora, GALE Year 1 Quarter 1, 2, and Interim release of BN audio transcripts, Multiple-Translation Chinese Corpus part 1, 2, and 3, and Chinese Gigaword corpus. Note that the word segmentation algorithm presented in [11] was applied and the unlabeled data was segmented into 750M words (34M sentences) of text.

Table 2. Comparison of the average 10-fold cross-validation tagging accuracy (%) on CTB from the HMM tagger and ME tagger. Known word, unknown word, and overall accuracies are included.

Tagger	Known	Unknown	Overall
HMM	95.0	76.2	94.3
ME	94.2	75.1	93.1

4.2. Co-training and self-training results

Table 2 presents the averaged 10-fold cross-validation tagging accuracy from the two taggers on CTB. The HMM tagger outperforms

[1]The part-of-speech tagging guidelines for the Penn Chinese Treebank (3.0), http://www.cis.upenn.edu/ chinese/posguide.3rd.ch.pdf.

[2]The goal of the GALE program is to develop computer software techniques to analyze, interpret, and distill information from speech and text in multiple languages.

[3]The Kappa Statistics is an index which compares the agreement against that which might be expected by chance. Kappa can be thought of as the chance-corrected proportional agreement, and possible values range from +1 (perfect agreement) via 0 (no agreement above that expected by chance) to -1 (complete disagreement).

the ME tagger, but both taggers are competitive on tagging newswire text. We examined the output of the two taggers on CTB and found that they made quite different errors. Hence, we hypothesized that the two taggers are sufficiently different to allow co-training to produce reasonable performance. Before conducting co-training or self-training, we found when using the two taggers trained on the entire CTB corpus to predict tags on the POS-eval test set, none of them gave satisfactory performance, as shown in Table 3. After adding the small seed corpus for training, the accuracy for both taggers was improved by about 10% absolutely. These results demonstrate the significant mismatch on style and word use between the newswire and BN genres and the importance of using a high quality in-domain seed corpus for semi-supervised training. However, this tagging performance is still unsatisfactory.

Table 4 shows that both self-training and co-training significantly improve the performance of the two taggers over directly training on CTB plus the seed corpus, with co-training strongly outperforming self-training, even for naive co-training. Note for self-training and co-training carried out in these experiments, we used cache size as 10K sentences. Among the four example selection approaches, the agreement-based approach yields the best accuracy from resulting taggers. Between agreement-based co-training and naive co-training, consistent with the findings from Clark et al. [1], agreement-based co-training is superior to naive co-training, since at each iteration this approach dynamically selects the examples that can improve the agreement rate and rejects ones that cannot fulfill the goal. In contrast, naive co-training adds all new examples in the cache which might accumulate noise during learning. On the other hand, the number of iterations of retraining that the agreement-based approach requires is generally an order of magnitude larger than that of naive co-training. Interestingly, the *max-t-min-s* approach proposed in this work produces comparable performance to the agreement-based method. Considering this approach is much more computationally efficient than the agreement-based approach, it might be promising to explore in other co-training tasks. Also, Table 4 demonstrates that *max-t-min-s* approach outperforms *max-score*. This shows that although *max-t-min-s* might let in many examples with errorful labels, the training utility of these examples for the *student* outweights the cost of errors introduced by these examples into the training data pool of the *student*. This observation of importance of training utility is consistent with the finding in active learning.

By applying co-training, we have achieved 5% to 7% relative improvement and 4.5% to 6% absolute improvement on POS tagging accuracy on Mandarin BN data by employing a quite small seed corpus of labeled data and a large amount of unlabeled data. Co-training also reduces the discrepancy between the two taggers and the best resulting POS tagging accuracy on the Mandarin BN POS evaluation test set is 94.1%, comparable to the 94.3% POS tagging accuracy we achieved on the newswire based CTB corpus using the HMM tagger[4]. We also found that we never obtained performance degradation from co-training, regardless of the number of iterations conducted. This observation is also consistent with the findings from Clark et al. [1] on the English newswire domain.

In conclusion, we have shown that co-training can be effectively applied to bootstrap POS taggers for tagging transcribed speech by combining labeled and unlabeled data. The agreement-based example selection approach outperforms naive co-training while a more computationally efficient approach proposed in this paper, which incorporates the idea of maximizing training utility from sample selection, performs comparably to the agreement-based method. In future work, we will carry out further investigations on example selection approaches, relations between the size of the manually labeled seed corpus and performance of different co-training setups, and effective combination of co-training and active learning. We will also apply co-training for POS tagging (and parsing) on more difficult genres like spontaneous speech.

[4]We achieved 94.8% POS tagging accuracy when applying co-training for the two taggers on CTB.

Table 3. Comparison of the tagging accuracy (%) of the HMM tagger and ME tagger when trained on the entire CTB corpus and the additional Mandarin BN seed corpus and tested on the Mandarin BN POS-eval test set. Known word, unknown word, and overall accuracies are included.

Tagger		Known	Unknown	Overall
HMM	CTB	80.0	69.2	79.0
	CTB+seed	90.5	75.1	89.6
ME	CTB	79.2	66.8	78.5
	CTB+seed	89.2	74.0	88.1

Table 4. Overall POS tagging accuracy (%) on the Mandarin BN POS-eval test set after applying self-training and co-training.

Training Condition		Tagger	
		HMM	ME
Initial (i.e., CTB+seed)		89.6	88.1
self-training		90.8	90.2
co-training	naive	91.9	91.8
	agreement-based	94.1	94.1
	max-score	93.2	93.1
	max-t-min-s	94.1	93.9

5. ACKNOWLEDGEMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. The authors thank Heng Ji for her work on manual POS annotation and Kristin Precoda for useful discussions regarding its content.

6. REFERENCES

[1] S. Clark, J. Curran, and M. Osborne, "Bootstrapping POS taggers using unlabelled data," in *Proceedings of CoNLL*, Edmonton, Canada, 2003, pp. 49–55.

[2] M. Mieskes and M. Strube, "Part-of-speech tagging of transcribed speech," in *Proceedings of LREC*, Genoa, Italy, 2006.

[3] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of COLT*, 1998.

[4] S. Abney, "Bootstrapping," in *Proceedings of ACL*, 2002.

[5] D. Pierce and C. Cardie, "Limitations of co-training for natural language learning from language datasets," in *Proceedings of EMNLP*, 2001.

[6] S. Dasgupta, M. Littman, and D. McAllester, "PAC generalization bounds for co-training," *In T. G. Dietterich and S. Becker and Z. Ghahramani, editors, Advances in Neural Information Processing Systems, MIT Press*, vol. 14, pp. 375–382, 2002.

[7] R. Hwa, "Sample selection for statistical grammar induction," in *Proceedings of Joing SIGDAT Conference on EMNLP and VLC*, Hongkong, China, 2000, pp. 45–52.

[8] K. Nigram and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of CIKM*, 2000.

[9] S. M. Thede and M. P. Harper, "A second-order Hidden Markov Model for part-of-speech tagging," in *Proceedings of ACL*, 1999, pp. 175–182.

[10] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133–142, 1996.

[11] M.-Y. Hwang, X. Lei, W. Wang, and T. Shinozaki, "Investigation on mandarin broadcast news speech recognition," in *Proceedings of ICSLP*, Pittsburgh, 2006, pp. 1233–1236.