Bootstrapping POS taggers using Unlabelled Data

Stephen Clark, James R. Curran and Miles Osborne

School of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh. EH8 9LW
{stephenc,jamesc,osborne}@cogsci.ed.ac.uk

Abstract

This paper investigates booststrapping part-of-speech taggers using co-training, in which two taggers are iteratively re-trained on each other's output. Since the output of the taggers is noisy, there is a question of which newly labelled examples to add to the training set. We investigate selecting examples by directly maximising tagger agreement on unlabelled data, a method which has been theoretically and empirically motivated in the co-training literature. Our results show that agreement-based co-training can significantly improve tagging performance for small seed datasets. Further results show that this form of co-training considerably outperforms self-training. However, we find that simply re-training on all the newly labelled data can, in some cases, yield comparable results to agreement-based co-training, with only a fraction of the computational cost.

1 Introduction

Co-training (Blum and Mitchell, 1998), and several variants of co-training, have been applied to a number of NLP problems, including word sense disambiguation (Yarowsky, 1995), named entity recognition (Collins and Singer, 1999), noun phrase bracketing (Pierce and Cardie, 2001) and statistical parsing (Sarkar, 2001; Steedman et al., 2003). In each case, co-training was used successfully to bootstrap a model from only a small amount of labelled data and a much larger pool of unlabelled data. Previous co-training approaches have typically used the score assigned by the model as an indicator of the reliability of a newly labelled example. In this paper we take a different approach, based on theoretical work by Dasgupta et al. (2002) and Abney (2002), in which newly labelled training examples are selected using a greedy algorithm which explicitly maximises the POS taggers' agreement on unlabelled data.

We investigate whether co-training based upon directly maximising agreement can be successfully applied to a pair of part-of-speech (POS) taggers: the Markov model TnT tagger (Brants, 2000) and the maximum entropy C&C tagger (Curran and Clark, 2003). There has been some previous work on boostrapping POS taggers (e.g., Zavrel and Daelemans (2000) and Cucerzan and Yarowsky (2002)), but to our knowledge no previous work on co-training POS taggers.

The idea behind co-training the POS taggers is very simple: use output from the TnT tagger as additional labelled data for the maximum entropy tagger, and vice versa, in the hope that one tagger can learn useful information from the output of the other. Since the output of both taggers is noisy, there is a question of which newly labelled examples to add to the training set. The additional data should be accurate, but also useful, providing the tagger with new information. Our work differs from the Blum and Mitchell (1998) formulation of co-training by using two different learning algorithms rather than two independent feature sets (Goldman and Zhou, 2000).

Our results show that, when using very small amounts of manually labelled seed data and a much larger amount of unlabelled material, agreement-based co-training can significantly improve POS tagger accuracy. We also show that simply re-training on all of the newly labelled data is surprisingly effective, with performance depending on the amount of newly labelled data added at each iteration. For certain sizes of newly labelled data, this simple approach is just as effective as the agreement-based method. We also show that co-training can still benefit both taggers when the performance of one tagger is initially much better than the other.

We have also investigated whether co-training can improve the taggers already trained on large amounts of

manually annotated data. Using standard sections of the WSJ Penn Treebank as seed data, we have been unable to improve the performance of the taggers using self-training or co-training.

Manually tagged data for English exists in large quantities, which means that there is no need to create taggers from small amounts of labelled material. However, our experiments are relevant for languages for which there is little or no annotated data. We only perform the experiments in English for convenience. Our experiments can also be seen as a vehicle for exploring aspects of co-training.

2 Co-training

Given two (or more) "views" (as described in Blum and Mitchell (1998)) of a classification task, co-training can be informally described as follows:

- Learn separate classifiers for each view using a small amount of labelled seed data.
- Use each classifier to label some previously unlabelled data.
- For each classifier, add some subset of the newly labelled data to the training data.
- Retrain the classifiers and repeat.

The intuition behind the algorithm is that each classifier is providing extra, informative labelled data for the other classifier(s). Blum and Mitchell (1998) derive PAC-like guarantees on learning by assuming that the two views are individually sufficient for classification and the two views are conditionally independent given the class.

Collins and Singer (1999) present a variant of the Blum and Mitchell algorithm, which directly maximises an objective function that is based on the level of agreement between the classifiers on unlabelled data. Dasgupta et al. (2002) provide a theoretical basis for this approach by providing a PAC-like analysis, using the same independence assumption adopted by Blum and Mitchell. They prove that the two classifiers have low generalisation error if they agree on unlabelled data.

Abney (2002) argues that the Blum and Mitchell independence assumption is very restrictive and typically violated in the data, and so proposes a weaker independence assumption, for which the Dasgupta et al. (2002) results still hold. Abney also presents a greedy algorithm that maximises agreement on unlabelled data, which produces comparable results to Collins and Singer (1999) on their named entity classification task.

Goldman and Zhou (2000) show that, if the newly labelled examples used for re-training are selected carefully, co-training can still be successful even when the views used by the classifiers do not satisfy the independence assumption.

In remainder of the paper we present a practical method for co-training POS taggers, and investigate the extent to which example selection based on the work of Dasgupta et al. and Abney can be effective.

3 The POS taggers

The two POS taggers used in the experiments are TNT, a publicly available Markov model tagger (Brants, 2000), and a reimplementation of the maximum entropy (ME) tagger MXPOST (Ratnaparkhi, 1996). The ME tagger, which we refer to as C&C, uses the same features as MXPOST, but is much faster for training and tagging (Curran and Clark, 2003). Fast training and tagging times are important for the experiments performed here, since the bootstrapping process can require many tagging and training iterations.

The model used by TNT is a standard tagging Markov model, consisting of emission probabilities, and transition probabilities based on trigrams of tags. It also deals with unknown words using a suffix analysis of the target word (the word to be tagged). TNT is very fast for both training and tagging.

The C&C tagger differs in a number of ways from TNT. First, it uses a conditional model of a tag sequence given a string, rather than a joint model. Second, ME models are used to define the conditional probabilities of a tag given some context. The advantage of ME models over the Markov model used by TNT is that arbitrary features can easily be included in the context; so as well as considering the target word and the previous two tags (which is the information TNT uses), the ME models also consider the words either side of the target word and, for unknown and infrequent words, various properties of the string of the target word.

A disadvantage is that the training times for ME models are usually relatively slow, especially with iterative scaling methods (see Malouf (2002) for alternative methods). Here we use Generalised Iterative Scaling (Darroch and Ratcliff, 1972), but our implementation is much faster than Ratnaparkhi's publicly available tagger. The C&C tagger trains in less than 7 minutes on the 1 million words of the Penn Treebank, and tags slightly faster than TNT.

Since the taggers share many common features, one might think they are not different enough for effective co-training to be possible. In fact, both taggers are sufficiently different for co-training to be effective. Section 4 shows that both taggers can benefit significantly from the information contained in the other's output.

The performance of the taggers on section 00 of the WSJ Penn Treebank is given in Table 1, for different seed set sizes (number of sentences). The seed data is taken

Tagger	50 seed	500 seed	≈ 40,000 seed
TnT	81.3	91.0	96.5
C&C	73.2	88.3	96.8

Table 1: Tagger performance for different seed sets

from sections 2–21 of the Treebank. The table shows that the performance of TnT is significantly better than the performance of C&C when the size of the seed data is very small.

4 Experiments

The co-training framework uses labelled examples from one tagger as additional training data for the other. For the purposes of this paper, a labelled example is a tagged sentence. We chose complete sentences, rather than smaller units, because this simplifies the experiments and the publicly available version of TnT requires complete tagged sentences for training. It is possible that co-training with sub-sentential units might be more effective, but we leave this as future work.

The co-training process is given in Figure 1. At each stage in the process there is a *cache* of unlabelled sentences (selected from the total pool of unlabelled sentences) which is labelled by each tagger. The cache size could be increased at each iteration, which is a common practice in the co-training literature. A subset of those sentences labelled by TnT is then added to the training data for C&C, and vice versa. Blum and Mitchell (1998) use the combined set of newly labelled examples for training each view, but we follow Goldman and Zhou (2000) in using separate labelled sets. In the remainder of this section we consider two possible methods for selecting a subset. The cache is cleared after each iteration.

There are various ways to select the labelled examples for each tagger. A typical approach is to select those examples assigned a high score by the relevant classifier, under the assumption that these examples will be the most reliable. A score-based selection method is difficult to apply in our experiments, however, since TnT does not provide scores for tagged sentences.

We therefore tried two alternative selection methods. The first is to simply add all of the cache labelled by one tagger to the training data of the other. We refer to this method as *naive co-training*. The second, more sophisticated, method is to select that subset of the labelled cache which maximises the agreement of the two taggers on unlabelled data. We call this method *agreement-based co-training*. For a large cache the number of possible subsets makes exhaustive search intractable, and so we randomly sample the subsets.

S is a seed set of labelled sentences
L_T is labelled training data for TnT
L_C is labelled training data for C&C
U is a large set of unlabelled sentences
C is a cache holding a small subset of U
initialise:
  $L_T \leftarrow L_C \leftarrow S$
  Train TnT and C&C on S
loop:
  Partition U into the disjoint sets C and U'.
  Label C with TnT and C&C
  Select sentences labelled by TnT and add to L_C
  Train C&C on L_C
  Select sentences labelled by C&C and add to L_T
  Train TnT on L_T
  $U = U'$.
Until U **is empty**

Figure 1: The general co-training process

C is a cache of sentences labelled by the *other* tagger
U is a set of sentences, used for measuring agreement
initialise:
  $c_{max} \leftarrow \emptyset$; $A_{max} \leftarrow 0$
Repeat n **times:**
  Randomly sample $c \subseteq C$
  Retrain *current* tagger using c as additional data
  if new agreement rate, A, on $U > A_{max}$
    $A_{max} \leftarrow A$; $c_{max} \leftarrow c$
return c_{max}

Figure 2: Agreement-based example selection

The pseudo-code for the agreement-based selection method is given in Figure 2. The *current* tagger is the one being retrained, while the *other* tagger is kept static. The co-training process uses the selection method for selecting sentences from the cache (which has been labelled by one of the taggers). Note that during the selection process, we repeatedly sample from *all* possible subsets of the cache; this is done by first randomly choosing the size of the subset and then randomly choosing sentences based on the size. The number of subsets we consider is determined by the number of times the loop is traversed in Figure 2.

If TnT is being trained on the output of C&C, then the most recent version of C&C is used to measure agreement (and vice versa); so we first attempt to improve one tagger, then the other, rather than both at the same time. The agreement rate of the taggers on unlabelled sentences is the per-token agreement rate; that is, the number of times each word in the unlabelled set of sentences is assigned the same tag by both taggers.

For the small seed set experiments, the seed data was an arbitrarily chosen subset of sections 10–19 of the WSJ Penn Treebank; the unlabelled training data was taken from 50,000 sentences of the 1994 WSJ section of the North American News Corpus (NANC); and the unlabelled data used to measure agreement was around 10,000 sentences from sections 1–5 of the Treebank. Section 00 of the Treebank was used to measure the accuracy of the taggers. The cache size was 500 sentences.

4.1 Self-Training and Agreement-based Co-training Results

Figure 3 shows the results for self-training, in which each tagger is simply retrained on its *own* labelled cache at each round. (By *round* we mean the re-training of a single tagger, so there are two rounds per co-training iteration.) TNT does improve using self-training, from 81.4% to 82.2%, but C&C is unaffected. Re-running these experiments using a range of unlabelled training sets, from a variety of sources, showed similar behaviour.

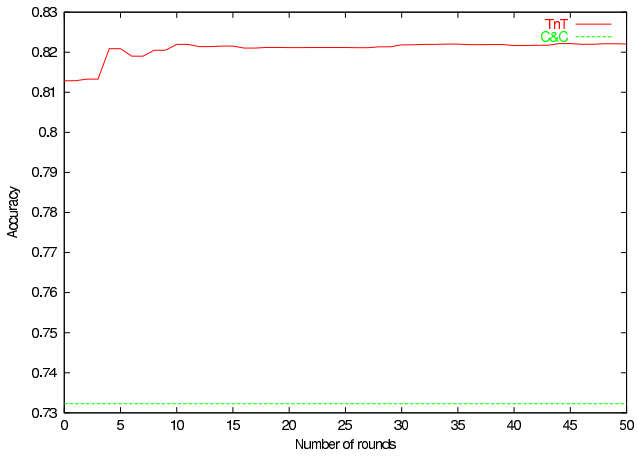

Figure 3: Self-training TNT and C&C (50 seed sentences). The upper curve is for TNT; the lower curve is for C&C.

Figure 4 gives the results for the greedy agreement co-training, using a cache size of 500 and searching through 100 subsets of the labelled cache to find the one that maximises agreement. Co-training improves the performance of *both* taggers: TNT improves from 81.4% to 84.9%, and C&C improves from 73.2% to 84.3% (an error reduction of over 40%).

Figures 5 and 6 show the self-training results and agreement-based results when a larger seed set, of 500 sentences, is used for each tagger. In this case, self-training harms TNT and C&C is again unaffected. Co-training continues to be beneficial.

Figure 7 shows how the size of the labelled data set (the number of sentences) grows for each tagger per round.

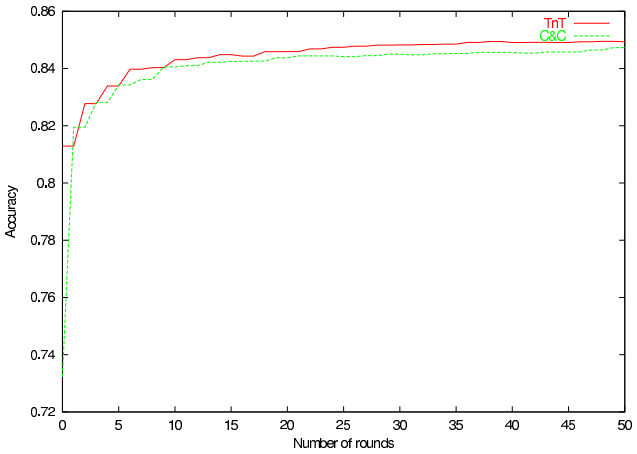

Figure 4: Agreement-based co-training between TNT and C&C (50 seed sentences). The curve that starts at a higher value is for TNT.

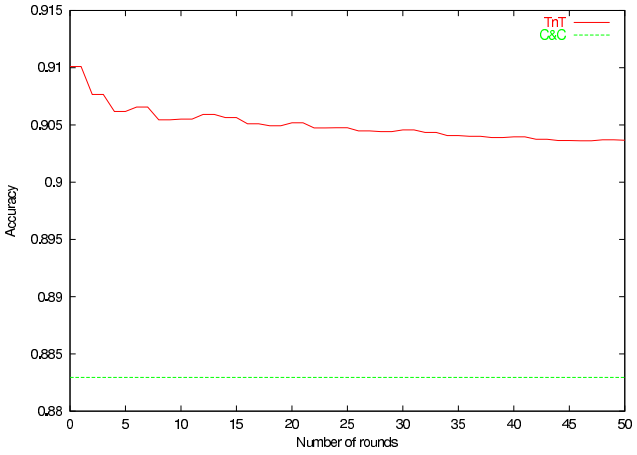

Figure 5: Self-training TNT and C&C (500 seed sentences). The upper curve is for TNT; the lower curve is for C&C.

Towards the end of the co-training run, more material is being selected for C&C than TNT. The experiments using a seed set size of 50 showed a similar trend, but the difference between the two taggers was less marked. By examining the subsets chosen from the labelled cache at each round, we also observed that a large proportion of the cache was being selected for both taggers.

4.2 Naive Co-training Results

Agreement-based co-training for POS taggers is effective but computationally demanding. The previous two agreement maximisation experiments involved retraining each tagger 2,500 times. Given this, and the observation that maximisation generally has a preference for selecting a large proportion of the labelled cache, we looked at *naive co-training*: simply retraining upon *all* available mate-

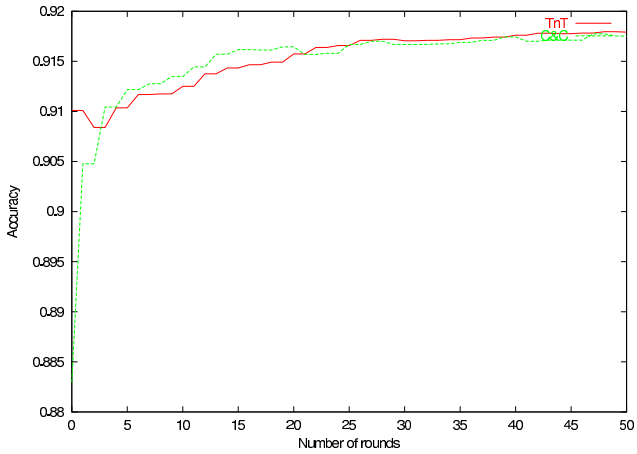

Figure 6: Agreement-based co-training between TNT and C&C (500 seed sentences). The curve that starts at a higher value is for TNT.

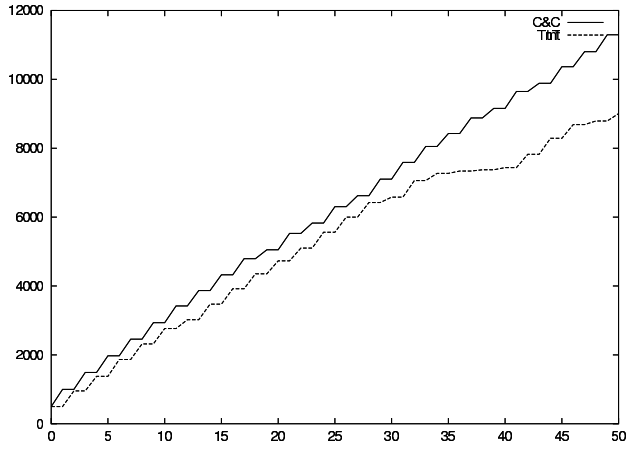

Figure 7: Growth in training-set sizes for co-training TNT and C&C (500 seed sentences). The upper curve is for C&C.

rial (i.e. the whole cache) at each round. Table 2 shows the naive co-training results after 50 rounds of co-training when varying the size of the cache. 50 manually labelled sentences were used as the seed material. Table 3 shows results for the same experiment, but this time with a seed set of 500 manually labelled sentences.

We see that naive co-training improves as the cache size increases. For a large cache, the performance levels for naive co-training are very similar to those produced by our agreement-based co-training method. After 50 rounds of co-training using 50 seed sentences, the agreement rates for naive and agreement-based co-training were very similar: from an initial value of 73% to 97% agreement.

Naive co-training is more efficient than agreement-based co-training. For the parameter settings used in the previous experiments, agreement-based co-training required the taggers to be re-trained 10 to 100 times more often then naive co-training. There are advantages to agreement-based co-training, however. First, the agreement-based method dynamically selects the best sample at each stage, which may not be the whole cache. In particular, when the agreement rate cannot be improved upon, the selected sample can be rejected. For naive co-training, new samples will always be added, and so there is a possibility that the noise accumulated at later stages will start to degrade performance (see Pierce and Cardie (2001)). Second, for naive co-training, the optimal amount of data to be added at each round (i.e. the cache size) is a parameter that needs to be determined on held out data, whereas the agreement-based method determines this automatically.

Amount added	TNT	C&C
0	81.3	73.2
50	82.9	82.7
100	83.5	83.3
150	84.4	84.3
300	85.0	84.9
500	85.3	85.1

Table 2: Naive co-training accuracy results when varying the amount added after each round (50 seed sentences)

Amount added	TNT	C&C
0	91.0	88.3
100	92.0	91.9
300	92.0	91.9
500	92.1	92.0
1000	92.0	91.9

Table 3: Naive co-training accuracy results when varying the amount added after each round (500 seed sentences)

4.3 Larger-Scale Experiments

We also performed a number of experiments using much more unlabelled training material than before. Instead of using 50,000 sentences from the 1994 WSJ section of the North American News Corpus, we used 417,000 sentences (from the same section) and ran the experiments until the unlabelled data had been exhausted.

One experiment used naive co-training, with 50 seed sentences and a cache of size 500. This led to an agreement rate of 99%, with performance levels of 85.4% and 85.4% for TNT and C&C respectively. 230,000 sentences ($\approx$ 5 million words) had been processed and were used as training material by the taggers. The other experiment used our agreement-based co-training approach (50 seed sentences, cache size of 1,000 sentences, explor-

ing at most 10 subsets in the maximisation process per round). The agreement rate was 98%, with performance levels of 86.0% and 85.9% for both taggers. 124,000 sentences had been processed, of which 30,000 labelled sentences were selected for training TnT and 44,000 labelled sentences were selected for training C&C.

Co-training using this much larger amount of unlabelled material did improve our previously mentioned results, but not by a large margin.

4.4 Co-training using Imbalanced Views

It is interesting to consider what happens when one view is initially much more accurate than the other view. We trained one of the taggers on much more labelled seed data than the other, to see how this affects the co-training process. Both taggers were initialised with either 500 or 50 seed sentences, and agreement-based co-training was applied, using a cache size of 500 sentences. The results are shown in Table 4.

Seed material		Initial Perf		Final Perf	
TnT	C&C	TnT	C&C	TnT	C&C
50	500	81.3	88.3	90.0	89.4
500	50	91.0	73.2	91.3	91.3

Table 4: Co-training Results for Imbalanced Views

Co-training continues to be effective, even when the two taggers are imbalanced. Also, the final performance of the taggers is around the same value, irrespective of the direction of the imbalance.

4.5 Large Seed Experiments

Although bootstrapping from unlabelled data is particularly valuable when only small amounts of training material are available, it is also interesting to see if self-training or co-training can improve state of the art POS taggers.

For these experiments, both C&C and TnT were initially trained on sections 00–18 of the WSJ Penn Treebank, and sections 19–21 and 22–24 were used as the development and test sets. The 1994–1996 WSJ text from the NANC was used as unlabelled material to fill the cache.

The cache size started out at 8000 sentences and increased by 10% in each round to match the increasing labelled training data. In each round of self-training or naive co-training 10% of the cache was randomly selected and added to the labelled training data. The experiments ran for 40 rounds.

The performance of the different training regimes is listed in Table 5. These results show no significant improvement using either self-training or co-training with very large seed datasets. Self-training shows only a slight improvement for C&C[1] while naive co-training performance is always worse.

Method	WSJ19–21		WSJ22–24	
	C&C	TnT	C&C	TnT
Initial	96.71	96.50	96.78	96.46
Self-train	96.77	96.45	96.87	96.42
Naive co-train	96.74	96.48	96.76	96.46

Table 5: Performance with large seed sets

5 Conclusion

We have shown that co-training is an effective technique for bootstrapping POS taggers trained on small amounts of labelled data. Using unlabelled data, we are able to improve TnT from 81.3% to 86.0%, whilst C&C shows a much more dramatic improvement of 73.2% to 85.9%.

Our agreement-based co-training results support the theoretical arguments of Abney (2002) and Dasgupta et al. (2002), that directly maximising the agreement rates between the two taggers reduces generalisation error. Examination of the selected subsets showed a preference for a large proportion of the cache. This led us to propose a naive co-training approach, which significantly reduced the computational cost without a significant performance penalty.

We also showed that naive co-training was unable to improve the performance of the taggers when they had already been trained on large amounts of manually annotated data. It is possible that agreement-based co-training, using more careful selection, would result in an improvement. We leave these experiments to future work, but note that there is a large computational cost associated with such experiments.

The performance of the bootstrapped taggers is still a long way behind a tagger trained on a large amount of *manually* annotated data. This finding is in accord with earlier work on bootstrapping taggers using EM (Elworthy, 1994; Merialdo, 1994). An interesting question would be to determine the minimum number of manually labelled examples that need to be used to seed the system before we can achieve comparable results as using all available manually labelled sentences.

For our experiments, co-training never led to a decrease in performance, regardless of the number of iterations. The opposite behaviour has been observed in other applications of co-training (Pierce and Cardie, 2001). Whether this robustness is a property of the tagging problem or our approach is left for future work.

[1] This is probably by chance selection of better subsets.

Acknowledgements

This work has grown out of many fruitful discussions with the 2002 JHU Summer Workshop team that worked on weakly supervised bootstrapping of statistical parsers. The first author was supported by EPSRC grant GR/M96889, and the second author by a Commonwealth scholarship and a Sydney University Travelling scholarship. We would like to thank the anonymous reviewers for their helpful comments, and also Iain Rae for computer support.

References

Steven Abney. 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367, Philadelphia, PA.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madisson, WI.

Thorsten Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Empirical Methods in NLP Conference*, pages 100–110, University of Maryland, MD.

Silviu Cucerzan and David Yarowsky. 2002. Bootstrapping a multilingual part-of-speech tagger in one person-day. In *Proceedings of the 6th Workshop on Computational Language Learning*, Taipei, Taiwan.

James R. Curran and Stephen Clark. 2003. Investigating GIS and Smoothing for Maximum Entropy Taggers. In *Proceedings of the* 11^{th} *Annual Meeting of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. (to appear).

J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

Sanjoy Dasgupta, Michael Littman, and David McAllester. 2002. PAC generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 375–382, Cambridge, MA. MIT Press.

D. Elworthy. 1994. Does Baum-Welch re-estimation help taggers? In *Proceedings of the* 4^{th} *Conference on Applied Natural Language Processing*, pages 53–58, Stuttgart, Germany.

Sally Goldman and Yan Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Workshop on Natural Language Learning*, pages 49–55, Taipei, Taiwan.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proceedings of the Empirical Methods in NLP Conference*, Pittsburgh, PA.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the EMNLP Conference*, pages 133–142, Philadelphia, PA.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of the 2nd Annual Meeting of the NAACL*, pages 95–102, Pittsburgh, PA.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. (to appear).

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA.

Jakub Zavrel and Walter Daelemans. 2000. Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 17–20, Athens, Greece.