In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*

Limitations of Co-Training for Natural Language Learning from Large Datasets

David Pierce and Claire Cardie
Department of Computer Science
Cornell University
Ithaca NY 14853
{pierce, cardie}@cs.cornell.edu

Abstract

Co-Training is a weakly supervised learning paradigm in which the redundancy of the learning task is captured by training two classifiers using separate views of the same data. This enables bootstrapping from a small set of labeled training data via a large set of unlabeled data. This study examines the learning behavior of co-training on natural language processing tasks that typically require large numbers of training instances to achieve usable performance levels. Using base noun phrase bracketing as a case study, we find that co-training reduces by 36% the difference in error between co-trained classifiers and fully supervised classifiers trained on a labeled version of all available data. However, degradation in the quality of the bootstrapped data arises as an obstacle to further improvement. To address this, we propose a moderately supervised variant of co-training in which a human corrects the mistakes made during automatic labeling. Our analysis suggests that corrected co-training and similar moderately supervised methods may help co-training scale to large natural language learning tasks.

1 Introduction

Co-Training (Blum and Mitchell, 1998) is a weakly supervised paradigm for learning a classification task from a small set of labeled data and a large set of unlabeled data, using separate, but redundant, views of the data. While previous research (summarized in Section 2) has investigated the theoretical basis of co-training, this study is motivated by practical concerns. We seek to apply the co-training paradigm to problems in natural language learning, with the goal of reducing the amount of human-annotated data required for developing natural language processing components. In particular, many natural language learning tasks contrast sharply with the classification tasks previously studied in conjunction with co-training in that they require hundreds of thousands, rather than hundreds, of training examples. Consequently, our focus on natural language learning raises the question of how co-training scales when a large number of training examples are required to achieve usable performance levels.

This case study of co-training for natural language learning addresses the scalability question using the task of base noun phrase identification. For this task, co-training reduces by 36% the difference in error between classifiers trained on 500 labeled examples and classifiers trained on 211,000 labeled examples. While this result is satisfying, further investigation reveals that deterioration in the quality of the labeled data accumulated by co-training hinders further improvement. We address this problem with a *moderately supervised* variant, *corrected co-training*, that employs a human annotator to correct the errors made during bootstrapping. Corrected co-training proves to be quite successful, bridging the remaining gap in accuracy. Analysis of corrected co-training illuminates an interesting tension within weakly supervised learning, between the need to bootstrap accurate labeled data, and the need to cover the desired task. We evaluate one approach, using corrected co-training, to resolving this tension; and as another approach, we suggest combining weakly supervised learning with active learning (Cohn et al., 1994).

The next section of this paper introduces issues and concerns surrounding co-training. Sections 3 and 4 describe the base noun phrase bracketing task, and the application of co-training to the task, respectively. Section 5 contains an evaluation of co-training for base noun identification.

2 Theoretical and Practical Considerations for Co-Training

The co-training paradigm applies when accurate classification hypotheses for a task can be learned from either of two sets of features of the data, each called a *view*. For example, Blum and Mitchell (1998) describe a web page classification task, in which the goal is to determine whether or not a given web page is a university faculty member's home page. For this task, they suggest the following two views: (1) the words contained in the text of the page; for example, *research interests* or *publications*; (2) the words contained in links pointing to the page; for example, *my advisor*.

The intuition behind Blum and Mitchell's co-training algorithm CT[1] (Figure 1) is that two views of the data can be used to train two classifiers that can help each other. Each classifier is trained using one view of the labeled data. Then it predicts labels for instances of the unlabeled data. By selecting its most confident predictions and adding the corresponding instances with their predicted labels to the labeled data, each classifier can add to the other's available training data. Continuing the above example, web pages pointed to by *my advisor* links can be used to train the page classifier, while web pages about *research interests* and *publications* can be used to train the link classifier.

Initial studies of co-training focused on the applicability of the co-training paradigm, and in particular, on clarifying the assumptions needed to ensure the effectiveness of the CT algorithm. Blum and Mitchell (1998) presented a PAC-style analysis of co-training, introducing the concept of compatibility between the target function and the unlabeled data: that is, the target function should assign the same label to an instance regardless of which view it sees. They made two additional important points: first, that each view of the data should itself be sufficient for learning the classification task; and second, that the views should be conditionally independent of each other in order to be useful. They proved that under these assumptions, a task that is learnable with random classification noise is learnable with co-training. In experiments with the CT algorithm, they noticed that it is important to preserve the distribution of class labels in the growing body of labeled data. Finally, they demonstrated the effectiveness of co-training on a web page classification task similar to that described above.

```
repeat until done
    train classifier $h_1$ on view $V_1$ of $L$
    train classifier $h_2$ on view $V_2$ of $L$
    allow $h_1$ to posit labels for examples in $U$
    allow $h_2$ to posit labels for examples in $U$
    add $h_1$'s most confidently labeled examples to $L$
    add $h_2$'s most confidently labeled examples to $L$
```

Figure 1: An abstract schema of Blum and Mitchell's CT algorithm for co-training using a small set of labeled data (L), a large set of unlabeled data (U), and two views of the data (V_1, V_2).

Collins and Singer (1999) were concerned that the CT algorithm does not strongly enforce the requirement that hypothesis functions should be compatible with the unlabeled data. They introduced an algorithm, CoBoost, that directly minimizes mismatch between views of the unlabeled data, using a combination of ideas from co-training and AdaBoost (Freund and Shapire, 1997).

Nigam and Ghani (2000) performed the most thorough empirical investigation of the desideratum of conditional independence of views underlying co-training. Their experiments suggested that view independence does indeed affect the performance of co-training; but that CT, when compared to other algorithms that use labeled and unlabeled data, such as EM (Dempster et al., 1977; Nigam et al., 2000), may still prove effective even when an explicit feature split is unknown, provided that there is enough implicit redundancy in the data.

In contrast to previous investigations of the theoretical basis of co-training, this study is motivated by practical concerns about the applica-

[1] We refer to Blum and Mitchell's co-training algorithm as CT, to distinguish it from alternative algorithms that exploit the co-training paradigm, i.e. by using labeled and unlabeled data partitioned into distinct views. CoBoost, mentioned below, is one such alternative algorithm.

(a) In [happier news], [South Korea], in establishing [diplomatic ties] with [Poland] [yesterday], announced [$450 million] in [loans] to [the financially strapped Warsaw government].

(b) In$_O$ [happier$_I$ news$_I$] ,$_O$ [South$_I$ Korea$_I$] ,$_O$ in$_O$ establishing$_O$ [diplomatic$_I$ ties$_I$] with$_O$ [Poland$_I$] [yesterday$_B$] ,$_O$ announced$_O$ [\$$_I$ 450$_I$ million$_I$] in$_O$ [loans$_I$] to$_O$ [the$_I$ financially$_I$ strapped$_I$ Warsaw$_I$ government$_I$] .$_O$

(c)

Left Context		Focus Word	Right Context		Label
/	In/IN	happier/JJR	news/NN	,/,	I
,/,	in/IN	establishing/VBG	diplomatic/JJ	ties/NNS	O
with/IN	Poland/NNP	yesterday/NN	,/,	announced/VBD	B

Figure 2: The base NPs in (a) are indicated by square brackets, and in (b) by IOB tags. The training instances in (c) consist of the focus word with part-of-speech tag, its context words with part-of-speech tags, and its IOB label. The part-of-speech tags are from the Penn Treebank tag set. Asterisks indicate missing features at the beginning or end of the sentence.

tion of weakly supervised learning to problems in natural language learning (NLL). Many NLL tasks contrast in two ways with the web page classification task studied in previous work on co-training. First, the web page task factors naturally into page and link views, while other NLL tasks may not have such natural views. Second, many NLL problems require hundreds of thousands of training examples, while the web page task can be learned using hundreds of examples.

Consequently, our focus on natural language learning introduces new questions about the scalability of the co-training paradigm. First, *can co-training be applied to learning problems without natural factorizations into views?* Nigam and Ghani's study suggests a qualified affirmative answer to this question, for a text classification task designed to contain redundant information; however, it is desirable to continue investigation of the issue for large-scale NLL tasks. Second, *how does co-training scale when a large number of training examples are required to achieve usable performance levels?* It is plausible to expect that the CT algorithm will not scale well, due to mistakes made by the view classifiers. To elaborate, the view classifiers may occasionally add incorrectly labeled instances to the labeled data. If many iterations of CT are required for learning the task, degradation in the quality of the labeled data may become a problem, in turn affecting the quality of subsequent view classifiers. For large-scale learning tasks, the effectiveness of co-training may be dulled over time.

Finally, we note that the accuracy of automatically accumulated training data is an important issue for many bootstrapping learning methods (e.g. Yarowsky (1995), Riloff and Jones (1999)), suggesting that the rewards of understanding and dealing with this issue may be significant.

3 Base Noun Phrase Identification

Base noun phrases (base NPs) are traditionally defined as nonrecursive noun phrases, i.e. NPs that do not contain NPs. (Figure 2a illustrates base NPs with a short example.) Base noun phrase identification is the task of locating the base NPs in a sentence from the words of the sentence and their part-of-speech tags. Base noun phrase identification is a crucial component of systems that employ partial syntactic analysis, including information retrieval (e.g. Mitra et al. (1997)) and question answering (e.g. Cardie et al. (2000)) systems. Many corpus-based methods have been applied to the task, including statistical methods (e.g. Church (1988)), transformation-based learning (e.g. Ramshaw and Marcus (1998)), rote sequence learning (e.g. Cardie and Pierce (1998)), memory-based sequence learning (e.g. Argamon et al. (1999)), and memory-based learning (e.g. Sang and Veenstra (1999)), among others.

Our case study employs a well-known bracket representation, introduced by Ramshaw and Marcus, wherein each word of a sentence is tagged with one of the following tags: **I**, meaning the word is within a bracket (*inside*); **O**, meaning the word is not within a bracket (*outside*); or **B**, meaning the word is within a bracket, but not the same bracket as the preceding word, i.e. the word *begins* a new bracket. Thus, the bracketing task is transformed into a word tagging task. Figure 2b repeats the example sentence, showing the IOB tag representation. Training examples for IOB tagging have the form

$$\langle w_{-k}/t_{-k}, \ldots, w_0/t_0, \ldots, w_k/t_k \; : \; l \rangle$$

where w_0 is the focus word (i.e. the word whose tag is to be learned) and t_0 is its syntactic category (i.e. part-of-speech) tag. Words to the left and right of the focus word are included for context. Finally, l is the IOB tag of w_0. Figure 2c illustrates a few instances taken from the example sentence.

We chose naive bayes classifiers for the study, first, because they are convenient to use and, indeed, have been used in previous co-training studies; and second, because they are particularly well-suited to co-training by virtue of calculating probabilities for each prediction. For an instance x, the classifier determines the maximum a posteriori label as follows.

$$\begin{aligned} l_{map} &= \arg \max_{l \in \{\text{I,O,B}\}} P(l|x) \\ &= \arg\max P(l)P(x|l) \\ &= \arg\max P(l) \prod_{i=-k}^{k} P(w_i/t_i \,|\, l) \end{aligned}$$

In experiments with these naive bayes IOB classifiers, we found that very little accuracy was sacrificed when the word information (i.e. w_i) was ignored by the classifier.[2] We therefore substitute the simpler term $P(t_i|l)$ for $P(w_i/t_i \,|\, l)$ above.

The probabilities $P(t_i|l)$ are estimated from the training data by determining the fraction of the instances labeled l that have syntactic category t_i (on word w_i), with m-estimation.

$$P(t_i|l) = \frac{N(t_i, l) + 1}{N(l) + 45}$$

Here $N(x)$ denotes the frequency of event x in the training data. This estimate smoothes the training probability by including virtual (unseen) samples for each part-of-speech tag (of which there are 45).

4 Co-Training for IOB Classifiers

To apply co-training, the base NP classification task must first be factored into views. For the IOB instances (vectors of part-of-speech tags indexed from $-k$ to k) a view corresponds to a subset of the set of indices $\{-k, \ldots, k\}$. The most natural views are perhaps $\{-k, \ldots, 0\}$ and $\{0, \ldots, k\}$, indicating that one classifier looks at the focus tag and the tags to its left, while the other looks at the focus tag and the tags to its right. Note that these views certainly violate the desideratum of conditional independence between view features since both include the focus tag. Other views, such as left/right views omitting the focus tag, for example, may be more theoretically attractive, but we found that the left/right views including focus proved most effectual in practice.

The IOB tagging task requires some minor modifications to the CT algorithm. First, it is impractical for the co-training classifiers to predict labels for each instance from the enormous set of unlabeled data. Instead, a smaller data pool is maintained, fed with randomly selected instances from the larger set.[3] Second, the IOB tagging task is a ternary, rather than a binary, classification. Furthermore, the distribution of labels in the training data is more unbalanced than the distribution of positive and negative examples in the web page task: namely, 53.9% of examples are labeled **I**, 44.0% **O**, and 2.1% **B**. Since it is impractical to add, say, 27 **I**, 22 **O**, and 1 **B**, to the labeled data at each step of co-training, instead, instances are selected by first choosing a label l at random according to the label distribution, then adding the instance

[2] This contrasts with other results, such as Ramshaw and Marcus' (1998), indicating that word information is important for base NP identification. We speculate that the naive bayes classifiers used here are simply not sophisticated enough to take advantage of word information.

[3] This standard modification was introduced by Blum and Mitchell (1998) in an effort to cover the underlying distribution of unlabeled instances; however, Nigam and Ghani (2000) found it to be unnecessary in their experiments.

```
repeat until done
    train classifier $h_1$ on view $V_1$ of $L$
    train classifier $h_2$ on view $V_2$ of $L$
    transfer randomly selected examples
        from $U$ to $U'$ until $|U'| = u$
    for $h \in \{h_1, h_2\}$
        allow $h$ to posit labels for all examples in $U'$
        repeat $g$ times
            select label $l$ at random according to $D_L$
            transfer most confidently labeled
                $l$ example from $U'$ to $L$
```

Figure 3: The modified co-training algorithm maintains a data pool U' of size u, and labels g instances per iteration selected according to the distribution of labels D_L. As in the original algorithm, L is the labeled data, U the unlabeled data, and V_1, V_2 the views.

most confidently labeled l to the labeled data. This procedure preserves the distribution of labels in the labeled data as instances are labeled and added. The modified CT algorithm is presented in Figure 3.

5 Evaluation

We evaluate co-training for IOB classification using a standard data set assembled by Ramshaw and Marcus from sections 15–18 (training data, 211727 instances) and 20 (test data, 47377 instances) of the Penn Treebank Wall Street Journal corpus (Marcus et al., 1993). Training instances consist of part-of-speech tag and IOB label for a focus word, along with contexts of two part-of-speech tags to the left and right of the focus. Our goal accuracy of 95.17% is the performance of a supervised IOB classifier trained on the correctly labeled version of the full training data. (In our experiments the goal classifier uses the left view of the data, which actually outperforms the combined left/right view.) For initial labeled data, the first L instances of the training data are given their correct labels. We determined the best setting for the parameters of the CT algorithm by testing multiple values: L (initial amount of labeled data) varied from 10 to 5000, then u (pool size) from 200 to 5000, then g (growth size) from 1 to 50. The best setting, in terms of effectiveness of co-training in improving the accuracy of the classifier, was $L = 500, u = 1000, g = 5$. These values are used throughout the evaluation unless noted otherwise.

Co-Training. We observe the progress of the co-training process by determining, at each iteration, the accuracy of the co-training classifiers over the test data. We also record the accuracy of the growing body of labeled data. These measurements can be plotted to depict a learning curve, indicating the progress of co-training as the classifier accuracy changes. Figure 4 presents two representative curves, one for the left context classifier and one for the labeled data. (The right context classifier behaves similarly to the left, but its performance is slightly worse.) As shown, co-training results in improvement in test accuracy over the initial classifier after about 160 iterations, reducing by 36% the difference in error between the co-training classifier and the goal classifier.

Unfortunately, the improvement in test accuracy does not continue as co-training progresses; rather, performance peaks, then declines somewhat before stabilizing at around 92.5%. We hypothesize that this decline is due to degradation in the quality of the labeled data. This hypothesis is supported by Figure 4b, indicating that labeled data accuracy decreases steadily before stabilizing at around 94%. Note that the accuracy of the classifier stabilizes at a point a bit lower than the stable accuracy of the labeled data, as would be expected if labeled data quality hinders further improvement from co-training.

Furthermore, co-training for base NP identification seems to be quite sensitive to the CT parameter settings. For example, with $L = 200$ the co-training classifiers appear not to be accurate enough to sustain co-training, while with $L = 1000$, they are too accurate, in the sense that co-training contributes very little accuracy before the labeled data deteriorates (Figure 5).

In the next sections, we address the problems of data degradation and parameter sensitivity for co-training.

Corrected Co-Training. As shown above, the degradation of the labeled data introduces a scalability problem for co-training because successive view classifiers use successively poorer quality data for training. A straightforward solution to this problem is to have a human an-

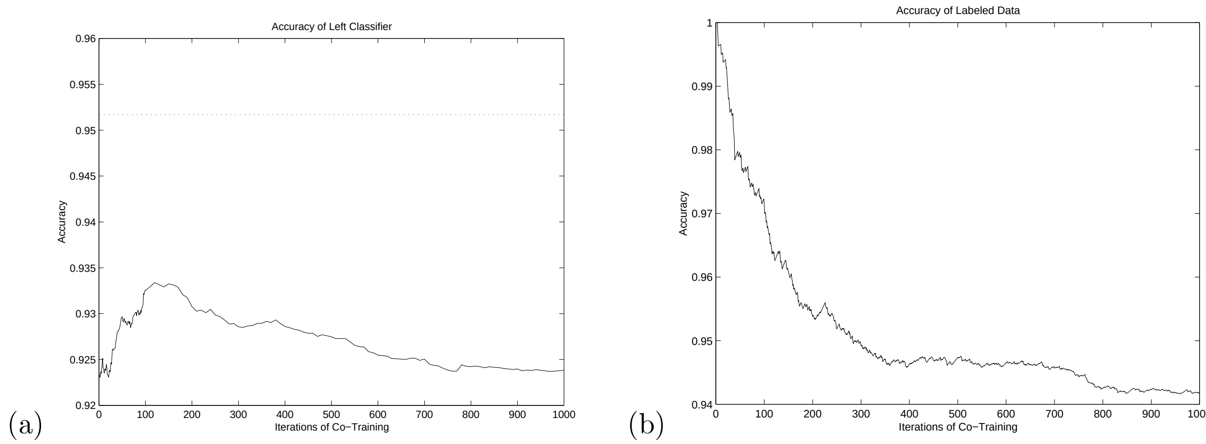

Figure 4: Learning curves for co-training. The solid curve in (a) indicates the accuracy of the left context classifier, while the dotted line shows the goal performance of the same classifier trained on a labeled version of the complete training data. Graph (b) is the accuracy of the labeled data over the course of co-training. Both curves are averages over five runs.

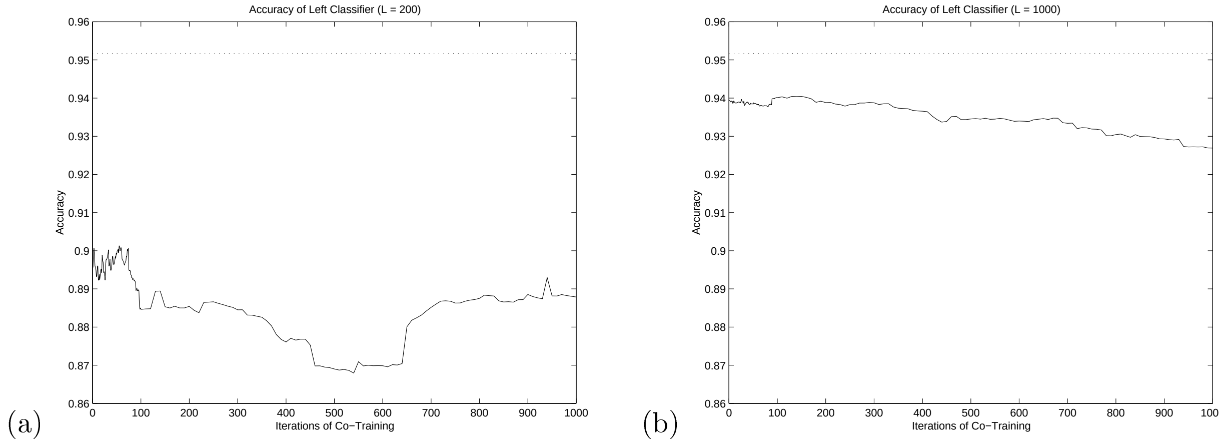

Figure 5: Learning curves for co-training with varying amounts of initial labeled data: (a) $L = 200$, (b) $L = 1000$.

notator intervene by reviewing and correcting instances labeled by the view classifiers.[4] By arresting the deterioration of the labeled data, we hope to prevent the ultimate decline in accuracy for co-training.

To evaluate this possibility, we simulate a human annotator by automatically correcting each newly labeled instance as it is added to the labeled data. Figure 6a presents the results of this experiment. As hypothesized, the classifier accuracy no longer suffers a decrease, instead increasing steadily to within about 0.5% of the goal accuracy after 800 iterations. For a real human annotator, the effort to achieve this improvement would have included reviewing 10 instances per iteration, or 8000 instances, but only correcting about 450 that were incorrectly labeled. Thus corrected co-training, a “moderately supervised” method, maintains the quality of the labeled data, yet the effort on the part of the human annotator remains small in proportion to the amount of data ultimately labeled and used for training.

[4] The human could also simply discard incorrectly labeled instances to reduce the effort; but we do not evaluate this alternative.

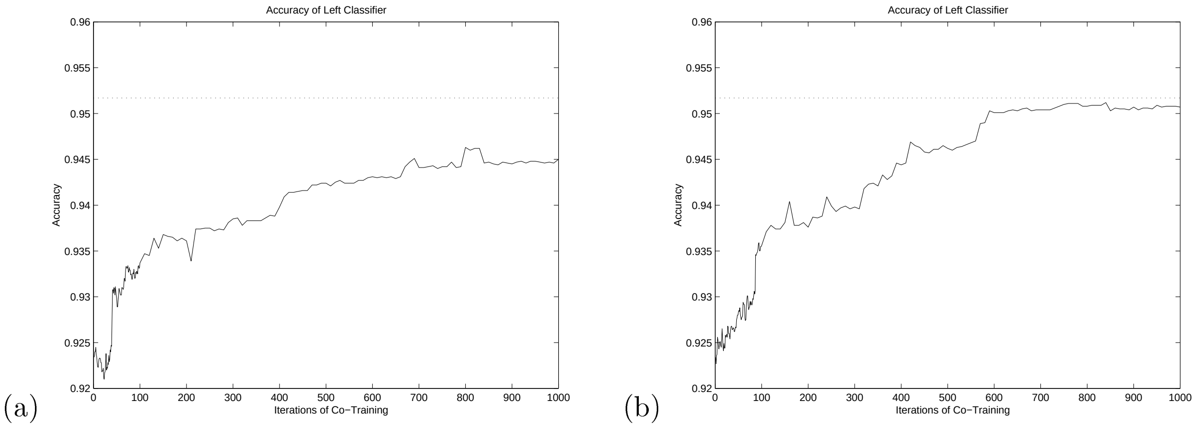

Figure 6: Corrected co-training (a) eliminates degradation of the labeled data by correcting labeling errors. With u set to 100 (b), corrected co-training achieves essentially the accuracy of the fully supervised classifier.

Furthermore, corrected co-training alleviates another problem with the CT algorithm, namely its sensitivity to the amount of initial labeled data. When too little labeled data is provided, the initial classifiers cannot sustain co-training improvement. With supervised correction, however, the risk of starting with too little labeled data is completely eliminated, so that co-training can be initiated with arbitrarily small amounts of labeled data.[5]

Note that corrected co-training is subtly different from co-testing (Muslea et al., 2000). One might say that co-testing is an active learning process that uses co-training, whereas corrected co-training is a co-training process that uses active learning. To be more precise, co-testing applies the compatibility concept from co-training to identify examples for a human annotator to label: examples whose classifications by the view classifiers differ (the contention set) are considered prime candidates for active learning queries. Corrected co-training, on the other hand, uses the view classifiers to select and label examples before presenting them to the human reviewer. Both methods have the goal of alleviating human annotation effort: co-testing by reducing the total number of instances annotated; corrected co-training by automatically labeling many of the unlabeled instances.

Task Coverage. Interestingly, there remains a slight gap between the accuracy of corrected co-training and the goal accuracy. To explain this, we hypothesize that co-training does not necessarily find the most useful unlabeled examples. More precisely, since unlabeled examples are selected and labeled based on the learners' confidence in their predictions, these examples are most likely representative of the part of the task space familiar to the classifiers, rather than helpful for learning new aspects of the task. In other words, with too little labeled data, co-training classifiers only learn part of the task.

By happy coincidence, the data pool parameter u of the CT algorithm allows us to indirectly explore this issue. Since the view classifiers must select instances to label from the data pool U', we can indirectly force them to select less certain, albeit potentially more useful, instances to label by setting a small pool size. By limiting the number of examples available to label, we prevent the view classifiers from selecting exclusively from familiar-looking examples. (Note that this is a terrible strategy for the original formulation of co-training, but with corrected co-training the labeled data remains flawless even if the view classifiers, hobbled by the small size of the pool of available examples, make many mistakes.)

Figure 6b presents the results of a corrected co-training experiment in which u is set to 100 (instead of 1000). The desired effect is real-

[5] There is an additional concern that with too little initial data, co-training may not be able to achieve adequate coverage of the task. This issue is addressed below.

ized, as co-training achieves 95.03% accuracy, just 0.14% away from the goal, after 600 iterations (and reaches 95.12% after 800 iterations). Additionally, the human annotator reviews 6000 examples and corrects only 358. Thus, by limiting the number of unlabeled examples under consideration—with the hope of forcing broader task coverage—we achieve essentially the goal accuracy in fewer iterations and with fewer corrections! Surprisingly, the error rate of the view classifiers per iteration remains essentially unchanged despite the reduction of the pool of unlabeled examples to choose from.

We believe the preceding experiment illuminates a fundamental tension in weakly supervised learning, between automatically obtaining reliable training data (usually requiring familiar examples), and adequately covering the learning task (usually requiring unfamiliar examples). This tension suggests that combining weakly supervised learning methods with active learning methods might be a fruitful endeavor. On one hand, the goal of weakly supervised learning is to bootstrap a classifier from small amounts of labeled data and large amounts of unlabeled data, often by automatically labeling some of the unlabeled data. On the other hand, the goal of active learning is to process (unlabeled) training examples in the order in which they are most useful or informative to the classifier (Cohn et al., 1994). Usefulness is commonly quantified as the learner's uncertainty about the class of an example (Lewis and Catlett, 1994). This neatly dovetails with the criterion for selecting instances to label in CT. We envision a learner that would alternate between selecting its most certain unlabeled examples to label and present to the human for acknowledgment, and selecting its most uncertain examples to present to the human for annotation. Ideally, efficient automatic bootstrapping would be complemented by good coverage of the task. We leave evaluation of this possibility to future work.

6 Conclusions

This case study explored issues involved with applying co-training to the natural language processing task of identifying base noun phrases, particularly, the scalability of co-training for large-scale problems. Our experiments indicate that co-training is an effective method for learning bracketers from small amounts of labeled data. Naturally, the resulting classifier does not perform as well as a fully supervised classifier trained on hundreds of times as much labeled data, but if the difference in accuracy is less important than the effort required to produce the labeled training data, co-training is especially attractive.

Furthermore, our experiments support the hypothesis that labeled data quality is a crucial issue for co-training. Our moderately supervised variant, corrected co-training, maintains labeled data quality without unduly increasing the burden on the human annotator. Corrected co-training bridges the gap in accuracy between weak initial classifiers and fully supervised classifiers.

Finally, as an approach to resolving the tension in weakly supervised learning between accumulating accurate training data and covering the desired task, we suggest combining weakly supervised methods such as co-training or self-training with active learning.

Acknowledgments

Thanks to three anonymous reviewers for their comments and suggestions. This work was supported in part by DARPA TIDES contract N66001-00-C-8009, and NSF Grants 9454149, 0081334, and 0074896.

References

S. Argamon, I. Dagan, and Y. Krymolowski. 1999. A memory-based approach to learning shallow natural language patterns. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(3). Available as cmp-lg/9806011.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*.

C. Cardie and D. Pierce. 1998. Error-driven pruning of treebank grammars for base noun phrase identification. In *Proceedings of the 36th Annual Meeting of the ACL and COLING-98*, pages 218–224. Available as cmp-lg/9808015.

C. Cardie, V. Ng, D. Pierce, and C. Buckley. 2000. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question answering system. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-NAACL 2000)*, pages 180–187.

K. Church. 1988. A stochastic parts programs and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.

D. Cohn, L. Atlas, and R. Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Y. Freund and R. Shapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

D. Lewis and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156.

M. Marcus, M. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

M. Mitra, C. Buckley, A. Singhal, and C. Cardie. 1997. An analysis of statistical and syntactic phrases. In *5TH RIAO Conference, Computer-Assisted Information Searching On the Internet*, pages 200–214.

I. Muslea, S. Minton, and C. Knoblock. 2000. Selective sampling with redundant views. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 621–626.

K. Nigam and R. Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Ninth International Conference on Information and Knowledge Management (CIKM-2000)*.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

L. Ramshaw and M. Marcus. 1998. Text chunking using transformation-based learning. In *Natural Language Processing Using Very Large Corpora*. Kluwer. Originally appeared in WVLC95.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 474–479.

E. Tjong Kim Sang and J. Veenstra. 1999. Representing text chunks. In *Proceedings of EACL'99*. Available as cs.CL/9907006.

D. 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.