Investigation of Co-training Views and Variations for Semantic Role Labeling

Rasoul Samad Zadeh Kaljahi
Department of AI, FCSIT
University Malaya, Malaysia
research@rasulsk.info

Mohd Sapiyan Baba
Department of AI, FCSIT
University Malaya, Malaysia
pian@um.edu.my

Abstract

Co-training, as a semi-supervised learning method, has been recently applied to semantic role labeling to reduce the need for costly annotated data using unannotated data. A main concern in co-training is how to split the problem into multiple views to derive learning features, so that they can effectively train each other. We investigate various feature splits based on two SRL views, constituency and dependency, with different variations of the algorithm. Balancing the feature split in terms of the performance of the underlying classifiers showed to be useful. Also, co-training with a common training set performed better than when separate training sets are used for co-trained classifiers.

1 Introduction

Semantic role labeling (SRL) parses a natural language sentence into its event structure. This information has been shown useful for several NLP tasks such as information extraction, question answering, summarization, and machine translation (Surdeanu et al., 2003; Gimenez and Marquez, 2008).

After its introduction by Gildea and Jurafsky (2002), a considerable body of NLP research has been devoted to SRL. CoNLL 2004 and 2005 (Carreras and Marquez, 2004; 2005) followed that seminal work by using similar input resources mainly built upon *constituent-based syntax* and achieved state-of-the-art results (Koomen et al., 2005). Subsequent CoNLL shared tasks (Surdeanu et al., 2008) put forth the use of another framework based on *dependency syntax*. This framework also led to well-performed systems (Johansson and Nugues, 2008).

Almost all of the SRL research has been based on supervised machine learning methods exploiting manually annotated corpora like FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). FrameNet annotates some example sentences for each *semantic frame*, which questions its representativeness of the language, necessary for statistical learning. Propbank, on the other hand, annotates all the sentences from WSJ corpus and remedies that problem to some extent, but unlike FrameNet, its coverage is limited to the newswire text of WSJ.

This domain dependence affects the performance of the systems using PropBank on any different domain of text (Carreras and Marquez, 2005). Considering the cost and difficulty of creating such resources with all of these shortcomings, it seems infeasible to build a comprehensive hand-crafted corpus of natural language for training robust SRL systems.

Such issues in statistical learning have motivated researchers to devise *semi-supervised* learning methods. These methods aim at utilizing a large amount of unannotated data along with small amount of annotated data. The existence of raw natural text in huge amounts is a promising point of using such methods for SRL.

Co-training is a semi-supervised algorithm in which two or more classifiers iteratively provide each other with the training examples by labeling unannotated data. Each classifier is based on the learning features derived from *conditionally independent* and *redundant* views of the underlying problem. These two assumptions are stated as the requirements of the algorithm in its original work by Blum and Mitchell (1998).

Constituency and dependency provide attractive views of SRL problem to be exploited in a co-training setup. The major motivation is the

Robust Unsupervised and Semi-Supervised Methods in Natural Language Processing, pages 41–49,
Hissar, Bulgaria, 15 September 2011.

promising results of their use in SRL, which satisfies the first assumption. There is a set of rules to convert constituency to dependency (Johansson and Nugues, 2007), which may question the second assumption. However, these rules are one-way, and moreover, Abney (2002) argues that this assumption can be loosened.

While several parameters are involved in co-training of SRL systems, the most important one is the split of the feature views. This work investigates the effects of feature split by comparing the co-training progress when using various splits. It also examines several variations of the algorithm. The algorithm is applied to the SRL problem when only a small amount of labeled data is available.

2 Related Work

Co-training was originally proposed by Blum and Mitchell (1998) for the problem of web page classification. They used hyper links pointing to the sample web page as one view and the content of the web page as another view to derive learning features. They could reduce the error rate of the base supervised classifier by co-training with unlabeled web pages.

Motivated by these results, the algorithm was applied to other NLP domains, ranging from binary classification problems like text classification (Nigam and Ghani, 2000) and reference resolution (Ng and Cardie, 2003) to more complex problems like parsing (Sarkar, 2001) and POS tagging (Clark et al., 2003). Some compared co-training with other semi-supervised algorithms like self-training and some studied variations of the algorithm for adapting it to the underlying problem. Whereas some of them reported successful results (Sarkar, 2001), some others preferred other algorithms over it (Ng and Cardie, 2003) or suggested further needs for studying the algorithm due to the large scale of the target problem (Pierce and Cardie, 2001).

Besides few other approaches to semi-supervised learning of SRL (Furstenau and Lapata, 2009), two works investigated the co-training algorithm for SRL.

He and Gildea (2006) addressed the problem of unseen FrameNet frames by using co-training (and self-training). They used syntactic and lexical views of the problem as two co-training views. They used only *tree path* as the syntactic and *head word form* as lexical features. To reduce the complexity of the task, they generalized argument roles to 15 thematic roles. The big performance gap between the two classifiers, unbalanced class distribution over examples, and the complexity of the task were argued as the reasons of the poor results.

Lee et al. (2007) investigated the utility of unlabeled data in amplifying the performance of SRL system. They trained Maximum Entropy classifiers on PropBank data as the base classifiers and used co-training to utilize a huge amount of unlabeled data (7 times more than labeled seed). The feature split they employed were the same as previous work, except they used more features for each view and also some features common between the views.

Unlike He and Gildea (2006) that used separate training sets for each classifier, they used a common training set. They only addressed core arguments to manage the complexity. Again, the performance gap between two views were high (~19 F1 points), but it is not clear why they reported the co-training results with the performance of all features instead of that of each view. They attributed the little gain to the low performance of the base classifiers and inadequacy of unlabeled data.

3 The SRL System

In order to be able to employ constituency and dependency features for two co-training views, we developed a two-platform SRL system: constituent-based and dependency-based.

One important issue in co-training of these two different platforms is that sample granularity in constituent-based system is a Penn tree constituent and in the dependency-based system is a dependency relation or a word token. Converting these to each other is necessary for co-training. Previous work (Hacioglu, 2004) shows that this conversion is not straightforward and negatively affect the performance.

To treat this issue we base our sample generation on constituency and then derive one dependency-based sample from every constituent-based sample. This sample is a word token (called *argument word* here), selected from among all word tokens inside the constituent using the heuristic used for preparing CoNLL 2008 shared task data (Surdeanu et al. 2008). This one-to-one relation is recorded in the system and helps avoid the conversion flaw. The system is described here.

Architecture: A three-stage pipeline architecture is used, where in the first stage less-probable argument candidates in the constituency parse tree are *pruned* using Xue and Palmer (2004) algorithm. In the next stage, final arguments are *identified* and *assigned* a semantic role jointly to decrease the complexity of task. In the final stage, a simple *global optimization* is performed using two constraints: a core argument role cannot be *repeated* for a predicate and arguments of a predicate cannot *overlap*. In addition, a preprocessing stage identifies the verb predicates of unlabeled sentences based on the parser's POS tags.

Features: Appendix A lists the learning features. Three types of features are used: constituent-based (C), dependency-based (D), and general (G) features which are not dependent on constituency or dependency. Columns 1 to 4 determine the feature sets and features present in each set, which will be described in the experiments section. We have tried to avoid features like named entity tags to depend less on extra annotation.

Classifier: *Maximum Entropy* is chosen as the base classifier for both views, because of its efficiency in training time and also its built-in multi-classification capability. Furthermore, it assigns a probability score for its predictions, which is useful in training data selection process in co-training. The *Maxent Toolkit*[1] is interfaced with the system for this purpose.

4 Co-training

Since the introduction of the original co-training algorithm, several variations of it have been used. These variants have usually been motivated by the characteristics of the underlying application. Figure 1 shows a generalized version of the algorithm with highlighted variables which constitute different versions of it. Some of the parameters addressed in this work are described here.

One important factor involved in bootstrapping is the performance of the base classifier (**C1** and **C2**). In co-training, another interesting parameter is the relative performance of the classifiers. We are interested in this parameter and investigate it by varying the feature split.

There are various stop criteria (**S**) used in literature, such as a pre-determined number of iterations, finishing all of the unlabeled data, or convergence of the process in terms of improvement. We use the second option for all experiments here, but we also look at convergence so that some data does not cause infinite loop.

In each iteration, one can label all of the unlabeled data or select and load a number of unlaleled examples (**p**) into a *pool* (**P**) and label only them. To study the effect of all parameters in a step by step approach, we do not use pool in this work and leave it for the future.

```
1- Add the seed example set L to currently
   empty training sets T1 and T2.
2- Train the base classifiers C1 and C2 with
   training sets T1 and T2 respectively.
3- Iterate the following steps until the stop cri-
   terion S is met.
   a- Select p examples from U into pool P.
   b- Label pool P with classifiers C1 and C2
   c- Select n labeled examples whose score
      meets a certain threshold t from P and
      add to training sets T1 and T2.
   d- Retrain the classifiers C1 and C2 with
      new training sets.
```

Figure 1: Generalized Co-training Algorithm

Selecting the newly labeled data to be **added** to the training set is the crucial point of co-training. First, it should be determined that both views use the *common* or *separate* training set during co-training. In the former case, ***T1*** and ***T2*** are identical. Then, it should be decided how the classifiers collaborate with each other.

With a common training set, selection can be done based on the prediction of both classifiers together. In one approach, only samples with the same predicted labels by both classifiers are selected (*agreement-based* selection). Another way is to select the most confidently labeled samples. Some select the most confident labelings from each view (Blum and Mitchell, 1998). In this method, a sample may be selected by both views, so this conflict needs to be resolved. We select the label for a sample with the highest confidence among both views (*confidenece-based* selection) to avoid conflict. Both approaches are investigated here.

With a separate trainings set, selection is done among samples labeled by each classifier individually (usually confidence-based). In this case, selected samples of one view are added to the training set of the other for collaboration. We are interetsed in the comparison of common and separate training sets, especially because from the two previous SRL co-training works, one was based on common (Lee et al., 2007) and the other on separate training sets (He and Gildea, 2006).

The next step is to chose the selection criteria. One can select all of the labeled examples, or one can only select a number of them (**n**), known as *growth size*, often based on a quality measure

[1] http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

F.S.	Synt. Input	**All Labeled Training Data**						**Seed Training Data**					
		WSJ Test			Brown Test			WSJ Test			Brown Test		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	cha	79.0	67.6	**72.9**	70.4	56.6	**62.7**	73.9	62.9	**68.0**	66.6	52.4	**58.6**
	cha.re*	79.3	73.4	**76.2**	68.6	60.8	**64.4**	75.6	68.8	**72.0**	65.1	56.1	**60.2**
2	malt	74.4	55.1	**63.3**	67.3	46.4	**55.0**	69.6	51.7	**59.4**	63.1	44.1	**51.9**
	conv*	75.5	60.8	**67.4**	69.7	52.9	**60.1**	73.6	56.9	**64.2**	66.0	47.7	**55.4**
3	cha	70.4	63.0	**66.5**	62.1	52.2	**56.8**	64.0	59.4	**61.6**	57.5	49.5	**53.2**
	cha.re*	71.2	68.8	**70.0**	68.6	60.8	**64.4**	70.4	64.3	**67.2**	60.7	53.3	**56.7**
4	malt	75.3	58.3	**65.7**	68.3	49.6	**57.5**	71.9	54.5	**62.0**	65.4	46.4	**54.2**
	conv*	76.6	64.5	**70.0**	69.7	52.9	**60.1**	76.3	59.5	**66.9**	69.0	49.8	**57.9**

Table 1: Performance of the Base Classifiers with Various Syntactic Inputs and Feature Sets

such as labeling confidence. To prevent poor labelings diminishing the quality of the training set, a threshold (**t**) is also set on this measure. We select all labeled samples here.

Finally, when adding the selected samples into the training set, a copy of them can be kept in the unlabeled data set and labeled again in the successive iterations, or all can be removed so that each sample is labeled only once. The former is called *delibility* and the latter *indelibility* (Abney 2008). We use the second method here.

5 Experiments and Results

This work uses co-training to address the SRL training problem when the amount of available annotated data is small.

The data and evaluation settings used are similar to the CoNLL 2005 and 2008 shared tasks. For evaluation, the same script used for 2005 shared task is used here and the measures are *precision*, *recall*, and their harmonic mean, *F1*. However, the data is changed in some ways to fulfill the objectives of this research, which is explained in the next section.

5.1 The Data

All the training data including labeled and unlabeled are selected from training sections of the shared tasks which consist of 39,832 PropBank sentences. The development data is WSJ section 24 of the PropBank, and the test data is WSJ section 23. Also, the Brown test data is used to evaluate the generalization ability of the system.

As syntactic input for the constituent-based system, training and test sentences were reparsed with the reranking parser of Charniak and Johnson (2005) instead of using the original parses of the shared task. The reason was a significant improvement of the SRL performance using the new parses in the preliminary experiments. These results are given in the next section for comparison.

For dependency-based system, the dependency syntax was prepared by converting the above constituent-based parses to dependency parses using the LTH converter (Johansson and Nugues, 2007). It should be noted that the data were also parsed using MaltParser (Nivre et al. 2007) at the same time, but the converter-based system outperformed it. These results are given in the next section for comparison.

As labeled seed data, 4,000 sentence of the training sentences are selected randomly. These sentences contain 70,345 argument samples covering 38 semantic roles out of 52 roles present in the total training set. Unlike previous work, we address all core and adjunctive roles.

As unlabeled training data, we use the remaining portion of the training data which contains 35,832 sentences, including 672,672 argument samples. We only address verb predicates and automatically identify them for unlabeled sentences instead of using the original predicate annotation of the data.

5.2 The Base Classifiers

Table 1 shows the performance of the base classifiers with different feature sets presented in section 3, and different syntactic input for each feature set. The first column lists the feature set numbers. In the second column, *cha* stands for the original Charniak parses of the data, and *cha.re* stands for the reranking parser used in this work. Also, *conv* stands for the converter-based dependency syntax and *malt* for dependency syntax produced by MaltParser. Those marked with * will be used here. Precision and recall are shown by *P* and *R* respectively.

To compare the performance of the classifiers with previous work, the results with all labeled data (39,832 sentences) are given on the left; to the right are the results with seed data only (4000 labeled sentences).

5.3 Feature Splits

We experimented with three kinds of feature splits. The first feature split (UBUS) uses feature sets 1 and 4. It is neither balanced nor separated: there is 5.2 and 2.4 points F_1 gap between their classifiers on WSJ and Brown test sets respectively (see Table 1, Seed Training Data, rows 2 and 8 of the result values), and they have 4 general features in common (See Appendix A). The idea behind this feature split is to understand the impact of feature separation and balancing.

The second one (UBS) consists of feature sets 1 and 2. According to Table 1 (Seed Training Data, rows 2 and 4 of the result values), there is a bigger F_1 gap between two classifiers (~8 and ~5 points on WSJ and Brown respectively) than previous split. Thus the classifiers are still *unbalanced*. However, it is a *separated* split, since there is no features common between feature sets.

The last split (BS) is also a *separated* split but has been *balanced* by moving all general features except predicate's POS tag into the dependency-based feature set. It consists of feature sets 3 and 4. According to Table 1 (Seed Training Data, rows 6 and 8 of the result values), the balance is only on F1 and gaps exist between precision and recall in opposite directions, which roughly compensate each other.

These three feature splits are used with three variations of the co-training algorithms described in section 4. In all settings, no pool is used and all unlabeled data are labeled in each iteration. Any sample which meets the selection criteria is selected and moved to training set (indelibility), i.e., no growth size and probability threshold is specified. The results are presented and discussed in the following sections.

5.4 Co-training by Common Training Set

Two variations of the algorithm, when using a common training set, are used and described here.

Agreement-based Selection: In each iteration, any sample for which the same label is assigned by both classifiers is selected and moved into training set. Figures 2 to 7 show the results with this setting. The left and right side figures are the results on WSJ Brown test sets respectively. Precision, recall, and F_1 are plotted for the classifier of each feature set as co-training progresses. The F_1 of the base classifiers and best co-trained classifier (in case of improvement) are marked on the graphs. Horizontal axis is based on co-training iterations, but the labels are the amounts of training samples used in each iteration.

It is also apparent that the dependency-based classifier is benefitting more from co-training. The reason may be twofold. First, with all splits, it has a higher precision than the other, which helps reduce noise propagation into the subsequent iterations. Next, with unbalanced splits (1 and 3) its performance is much lower and there is more room for improvement.

All the figures show an improvement on Brown test set. Seemingly, since this test set suffers from unseen events more than the other test set, new data is more useful for it.

Most of the unlabeled data (~90%) is added in the first iteration, showing a high level of agreement between classifiers.

Figure 2 shows that there is no improvement by co-training with feature set UBUS on WSJ test set over the baseline, though the dependency-based classifier improves. The feature split UBS in Figure 4, which fully separates the two feature sets, also could not gain any benefit. It seems that separating feature sets is not effective with the presence of a large gap between classifiers. This is further confirmed by observing the results for feature split BS in Figure 6, where the gap has been decreased to 0.4 F_1 points, and co-training could improve the baseline by 0.7 points.

Although these improvements are slight, but more runs of the experiments with different random selections of seed and unlabeled data showed a consistent behavior.

Confidence-based Selection: Due to the nature of this kind of selection and since there is no growth size and probability criteria, all samples are added to the training set at once, with a label that its classifier is more confident than the other's. Therefore, instead in a chart, the results could be presented in a table (Table 2). The first column lists the feature splits. In the second column, 0 stands for the base classifier and 1 is for classifier of the first (and the only) iteration.

Using all data at once leads to an overall final classifiers performance, unlike the previous setting in which remaining data for the following iterations degraded the progress.

Considering the high level of agreement between classifiers (%90), a similar behavior to agreement-based method is observed with this method as expected. The trend of precision and recall, more improvement of dependency-based classifier, and better results on Brown test set are consistent with agreement-based co-training.

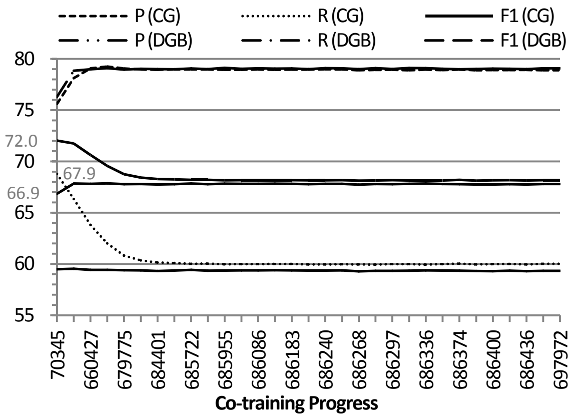

Figure 2: Agreement-based Co-training with Feature Split UBUS (WSJ Test Set)

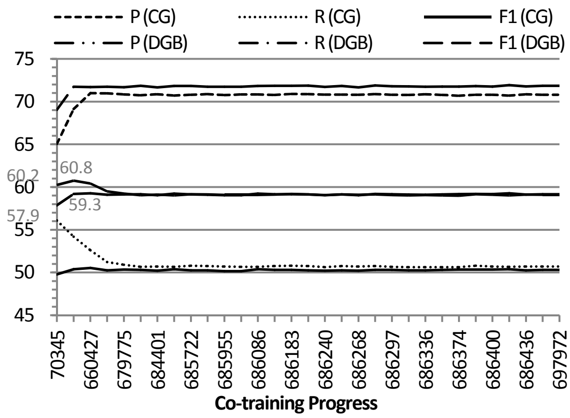

Figure 3: Agreement-based Co-training with Feature Split UBUS (Brown Test Set)

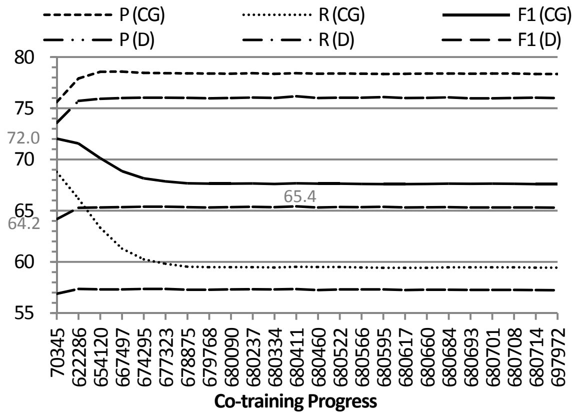

Figure 4: Agreement-based Co-training with Feature Split UBS (WSJ Test Set)

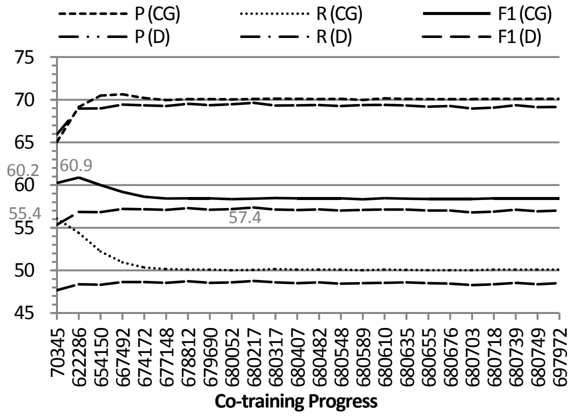

Figure 5: Agreement-based Co-training with Feature Split UBS (Brown Test Set)

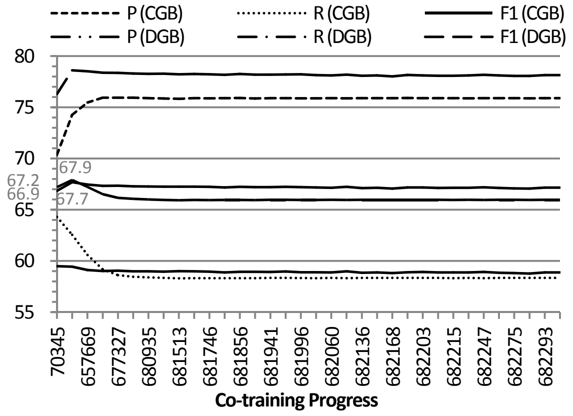

Figure 6: Agreement-based Co-training with Feature Split BS (WSJ Test Set)

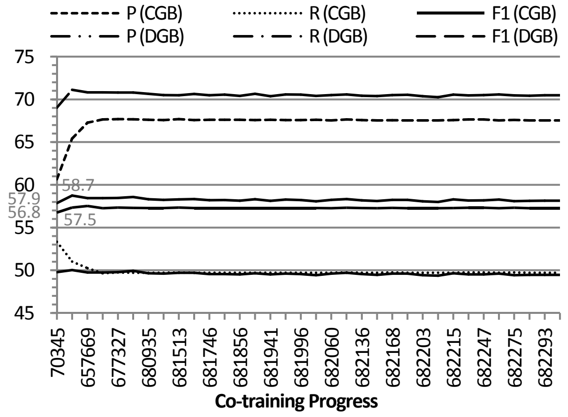

Figure 7: Agreement-based Co-training with Feature Split BS (Brown Test Set)

However, the separation of feature sets has even degraded the results over UBUS (71.2 vs. 71.8 and 59.8 vs. 60.5 F_1 points), but balancing has been again useful and improved the baselines by 0.4 and 0.9 F_1 points on WSJ and Brown test sets respectively. Comparing these values correspondingly to 0.7 and 0.9 point gains by agreement-based co-training with feature split BS shows that the latter has been slightly more promising.

5.5 Co-training by Separate Training Sets

As with confidence-based selection, with this variation of the algorithm, all samples are added to the training set at once. Table 3 shows the performance of the algorithm.

FS	It.	WSJ Test Set						Brown Test Set					
		Constituent-based			Dependency-based			Constituent-based			Dependency-based		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
UBUS	0	75.6	68.8	72.0	76.3	59.5	66.8	65.1	56.1	60.2	69.0	49.8	57.9
	1	79.0	65.8	**71.8**	77.5	59.8	**67.5**	70.5	53.0	**60.5**	70.6	50.6	**59.0**
UBS	0	75.6	68.8	72.0	73.6	56.9	64.2	65.1	56.1	60.2	66.0	47.7	55.4
	1	78.3	65.4	**71.2**	74.9	58.0	**65.4**	69.4	52.5	**59.8**	69.1	49.8	**57.9**
BS	0	70.4	64.3	67.2	76.3	59.5	66.9	60.7	53.3	56.8	69.0	49.8	57.9
	1	76.1	60.9	**67.6**	78.0	59.3	**67.4**	67.4	50.3	**57.6**	70.5	50.4	**58.8**

Table 2: Co-training Performance with Confidence-based Selection

FS	It.	WSJ Test Set						Brown Test Set					
		Constituent-based			Dependency-based			Constituent-based			Dependency-based		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
UBUS	0	75.6	68.8	72.0	76.3	59.5	66.9	65.1	56.1	60.2	69.0	49.8	57.9
	1	79.0	59.8	**68.0**	75.5	59.5	**66.6**	70.2	49.3	**57.9**	67.8	50.8	**58.1**
UBS	0	75.6	68.8	72.0	73.6	56.9	64.2	65.1	56.1	60.2	66.0	47.7	55.4
	1	76.7	58.2	**66.2**	73.7	58.1	**65.0**	69.3	49.5	**57.7**	67.2	50.0	**57.3**
BS	0	70.4	64.3	67.2	76.3	59.5	66.9	60.7	53.3	56.8	69.0	49.8	57.9
	1	76.2	57.9	**65.8**	75.1	58.4	**65.7**	67.5	48.8	**56.7**	67.5	49.9	**57.4**

Table 3: Co-training Performance with Separate Training Sets

The constituent-based classifier has been degraded with all feature splits. This even includes balanced and separated feature split (BS), which improved in previous settings.

The dependency-based system, which has always improved before, now degrades when using feature split BS, even on the Brown test set which has been previously benefited with all settings. On the other hand, feature split UBS improves on both test sets, possibly for the same reasons described before. However, the improvement of the dependency-based system with unbalanced feature split is not useful, because the performance of the constituent-based system is much higher, and it does not seem that the dependency-based classifier can reach to (or improve over it) even with more unlabeled data.

It can be seen that this variation of the algorithm performs worse compared to co-training with the common training set. Since in that case, in addition to training on the results of each other, the decision on selecting labeled data is made by both classifiers, this additional cooperation may be the possible reason of this observation.

6 Conclusion and Future Work

This work explores co-training with two views of SRL, namely constituency and dependency. Inspired by the two co-training assumptions, we investigate the performance of the algorithm with three kinds of feature splits: an unbalanced split with some general features in common between feature sets, an unbalanced but fully separated split, and a balanced and fully separated split.

In addition, three variations of the algorithms were examined with all feature splits: agreement-based and confidence-based selection for co-training with common training set, and co-training with separate training sets.

Results showed that the balanced feature split, in which the performances of the classifiers were roughly the same, is more useful for co-training. Moreover, balancing the feature split to reduce performance gap between associated classifiers, is more important than separating feature sets by removing common features.

Also, a common training set proved useful for co-training, unlike separate training sets. However, more experiments are needed to compare agreement- and confidence-based selections.

Due to significant difference between the current work and previous work on SRL co-training described in section 2 comparison is difficult. Nevertheless, unlike He and Gildea (2006), co-training showed to be useful for SRL here, though with slight improvements. In addition, the statistics reported by Lee et al. (2007) are unclear to compare for the reason mentioned in that section. However, as they concluded, more unlabeled data is needed for co-training to be practically useful.

As mentioned, we did not involve parameters like pool, growth size and probability threshold for a step-by-step study. A future work can be to investigate the effect of these parameters. Another direction of future work is to adapt the SRL architecture to better match with the co-training.

References

Abney, S. 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 360-367.

Abney, S. 2008. *Semisupervised Learning for Computational Linguistics.* Chapman and Hall, London.

Baker, F., Fillmore, C. and Lowe, J. 1998. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*, pages 86-90.

Blum, A. and Mitchell, T. 1998. Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98)*, pages 92-100.

Charniak, E. and Johnson, M. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 173-180.

Carreras, X. and Marquez, L. 2004. 'Introduction to the CoNLL-2004 Shared Task: Semantic role labeling', In *Proceedings of the 8th Conference on Computational Natural Language Learning*, pages 89-97.

Carreras, X. and Marquez, L. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the 9th Conference on Natural Language Learning*, pages 152-164.

Clark S., Curran, R. J. and Osborne M. 2003. Bootstrapping POS taggers using Unlabeled Data. In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL,* pages 49-55.

Furstenau, H. and Lapata, M. 2009. Graph Alignment for Semi-Supervised Semantic Role Labeling. In *Proceedings of the 2009 Conference on EMNLP*, pages 11-20.

Gildea, D. and Jurafsky, D. 2002. Automatic labeling of semantic roles. *CL*, 28(3): 245-288.

Gimenez, J. and Marquez, L. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation,* ACL, pages 195-198.

Hacioglu, K. 2004. Semantic Role Labeling Using Dependency Trees. In *Proceedings of 20th international Conference on Computational Linguistics*.

He, S. and Gildea, H. 2006. Self-training and Co-training for Semantic Role Labeling: Primary Report. TR 891, University of Colorado at Boulder.

Johansson, R. and Nugues, P. 2008. Dependency-based syntactic-semantic analysis with PropBank and NomBank. In *Proceedings of the 12th Conference on Computational Natural Language Learning*, pages 183-187.

Johansson, R. and Nugues, P. 2007. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA 2007*, pages 105-112.

Lee, J., Song, Y. and Rim, H. 2007. Investigation of Weakly Supervised Learning for Semantic Role Labeling. In *Proceedings of the Sixth international Conference on Advanced Language Processing and Web information Technology*, pages 165-170.

Ng, V. and Cardie, C. 2003. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the HLT-NAACL,* pages 94-101.

Nigam, K. and Ghani, R. 2000. Analyzing the Effectiveness and Applicability of Co-training. In *Proceedings of the 9th conference on Information and knowledge management,* pages 86-93.

Nivre, J. Hall, J. Nilsson, J. Chanev, A. Eryigit, G. Kubler, S. Marinov, S. and Marsi, E. 2007. MaltParser: A Language-independent System for Data-driven Dependency Parsing. *Natural Language Engineering*, 13(2): 95–135.

Palmer, M., Gildea, D. and Kingsbury, P. 2005, The Proposition Bank: A Corpus Annotated with Semantic Roles, *Computational Linguistics*, 31(1).

Pierce, D. and Cardie, C. 2001. Limitations of Co-Training for Natural Language Learning from Large Datasets. In *Proceedings of the 2001 Conference on EMNLP*, pages 1-9.

Koomen, P., Punyakanok, V., Roth, D. and Yi, W. 2005. Generalized inference with multiple semantic role labeling systems. In *Proceedings of the 9th Conference on Natural Language Learning*, pages 181-184.

Sarkar, A. 2001. Applying Co-Training Methods to Statistical Parsing. In *Proceedings of the 2001 Meeting of the North American chapter of the Association for Computational Linguistics*, pages 175-182.

Surdeanu, M., Harabagiu, S., Williams, J. and Aarseth, P. 2003. Using predicate argument structures for information extraction. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 8-15.

Surdeanu, M., Johansson, R., Meyers, A., Marquez, L. and Nivre, J. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Natural Language Learning*, pages 159-177.

Xue, N. and Palmer, M. 2004. Calibrating Features for Semantic Role Labeling. In *Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing.*

Appendix A. Learning Features

Feature Name	Type	1	2	3	4
Phrase Type	C	√		√	
Path	C	√		√	
Content Word Lemma	C	√		√	
Head Word POS	C	√		√	
Content Word POS	C	√		√	
Governing Category	C	√		√	
Predicate Subcategorization	C	√		√	
Constituent Subcategorization	C	√		√	
Clause+VP+NP Count in Path	C	√		√	
Constituent and Predicate Distance	C	√		√	
Head Word Location in Constituent	C	√		√	
Dependency Relation of Argument Word with Its Head	D		√		√
Dependency Relation of Predicate with Its Head	D		√		√
Lemma of Dependency Head of Argument Word	D		√		√
POS Tag of Dependency Head of Argument Word	D		√		√
Relation Pattern of Predicate’s Children	D		√		√
Relation Pattern of Argument Word Children	D		√		√
POS Pattern of Predicate’s Children	D		√		√
POS Pattern of Argument Word’s Children	D		√		√
Relation Path from Argument Word to Predicate	D		√		√
POS Path from Argument Word to Predicate	D		√		√
Family Relationship between Argument Word and Predicate	D		√		√
POS Tag of Least Common Ancestor of Argument Word and Predicate	D		√		√
POS Path from Argument Word to Least Common Ancestor	D		√		√
Dependency Path Length from Argument Word to Predicate	D		√		√
Whether Argument Word Starts with Capital Letter?	D		√		√
Whether Argument Word is WH word?	D		√		√
Head or Argument Word Lemma	G	√			√
Compound Verb Identifier	G	√			√
Position+Predicate Voice	G	√			√
Predicate Lemma	G	√			√
Predicate POS	G	√		√	