CS 388 Natural Language Processing
Homework 3: Neural Dependency Parsing with "Unsupervised" Domain Adaptation

Victoria Anugrah Lestari (val565)

April 5, 2017

1 Introduction

The purpose of this assignment is to roughly replicate the experiment performed by Roi Reichart and Ari Rapoport in their paper "Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets". They performed self-training with small labeled datasets for two reasons. The first is to improve the quality of a parser. The second is to adapt the parser to a different domain.

In this homework, we are instructed to conduct in-domain experiment and out-of-domain experiment. In-domain experiment means that we train the parser on one part of the corpus and test it on another part of the same corpus. On the other hand, out-of-domain experiment means that we train the parser on a corpus (seed) and test it on another corpus with a different topic. This parser will not perform well; thus, we use semi-supervised learning (also called self-training) by training the parser on the seed data and testing it on the other corpus to produce annotated results. The annotated results are concatenated with the seed, and then the parser is retrained.

2 Datasets

We use the Wall Street Journal (WSJ) corpus and the Brown corpus. We perform two sets of experiments. In the first set, the WSJ corpus is used as the source (training set), and the Brown corpus is used as the target (testing set). In the second set, the Brown corpus is used as the source and the WSJ corpus as the target. The size of the Brown test set is 2514 sentences, which is roughly 10% of all sentences in the Brown corpus. The size of the WSJ test set is 1981 sentences from folder 23.

3 Implementation

3.1 Preprocessing

The instruction specifies that we use training sets with various sizes. For the WSJ corpus, we have to use 1,000, 2,000, 3,000, 4,000, 5,000, 7,000, 10,000, 12,000, and 14,000 sentences from sections 2-10. For the Brown corpus, we have to use 1,000 to 2,000, 3,000, 4,000, 5,000, 7,000, 10,000, 13,000, 17,000, and 21,000 sentences. Since the Brown corpus is grouped by topics, I ensure that the topics are distributed evenly in the training sets. To generate the training sets and testing sets, I created a Python script `create_seed.py`.

3.2 Scenarios

Each set of experiment has four scenarios. For example, with the WSJ corpus as source and the Brown corpus as target, the scenarios look like the following:

- Training on WSJ sections 2-10, testing on WSJ section 23.
- Training on WSJ sections 2-10, testing on 10% Brown.
- Training on WSJ sections 2-10, self-training on 90% Brown, testing on 10% Brown.

- Training on WSJ sections 2-10 (10,000 sentences only), self-training on Brown (increase the self-training set from 1,000 to 21,000), and testing on 10% Brown.

The scenarios in the second set of the experiment (with Brown as source and WSJ as target) are similar to those above, with 90% Brown as the seed set, WSJ 2-10 as the self-training set, and 10% Brown as the testing set.

I used Stanford NLP Dependency Parser to train the parser for 200 iterations. For the self-training scenario, I added a pre-model to improve the performance of the parser. A pre-model is the model generated when we train the parser on the seed and test it on the self-training set. I modified `DependencyParserAPIUsage.java` to allow the user to select one of these scenarios when running the Java program. I ran the experiments on my laptop, on the CS server, and on Condor.

Further clarification about how to run the program is written in the README file.

4 Discussion

Figure 1 shows the performance comparison between in-domain experiment and two out-of-domain experiments using WSJ as source and Brown as target.

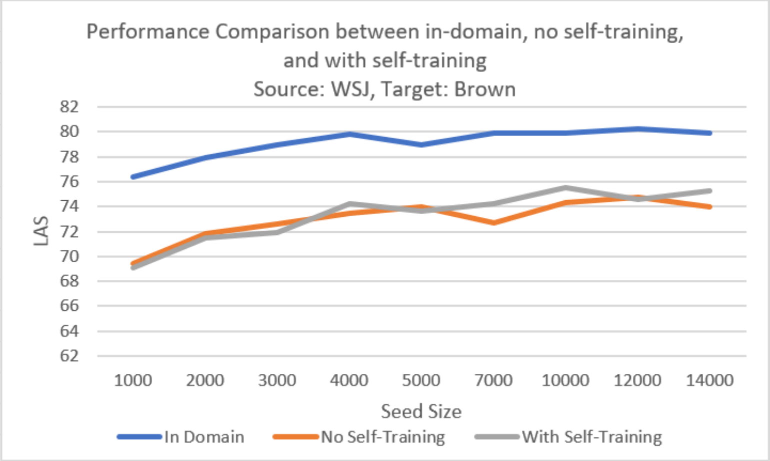

Figure 1: Performance comparison for WSJ as source and Brown as target

Figure 2 shows the performance comparison between in-domain experiment and two out-of-domain experiments using Brown as source and WSJ as target.

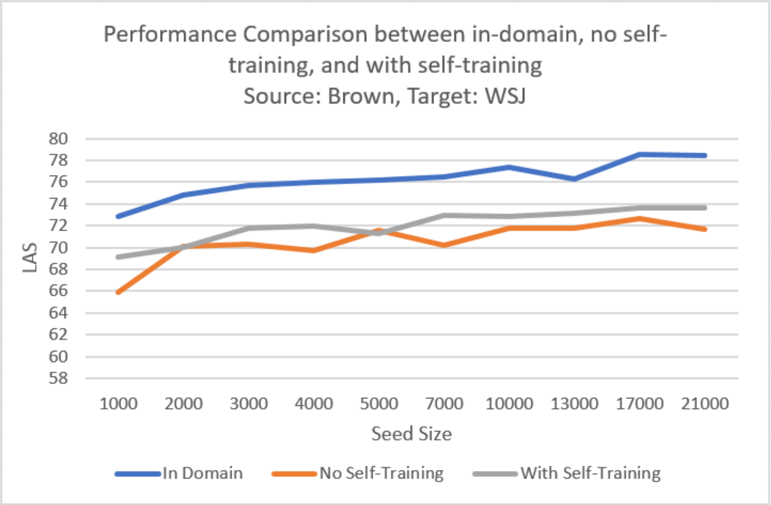

Figure 2: Performance comparison for Brown as source and WSJ as target

4.1 Comparison between In-Domain and Out-of-Domain Testing

For any of the scenarios, we observe that the labeled attachment scores (LAS) for in-domain testing are always higher than the scores for out-of-domain testing. Table 1 shows the performance of in-domain testing, out-of-domain without self-training, and out-of-domain with self-training for multiple seed sizes (from 1000 to 14000) using WSJ as source and Brown as target. The performance drops from in-domain testing to out-of-domain testing, ranging from 5% to 7%.

Table 1: Labeled Attachment Scores (LAS) using WSJ as source and Brown as target

Seed size	In-domain	No self-training	With self-training
1000	76.4141	69.4037	69.1071
2000	77.924	71.8649	71.4679
3000	78.9594	72.5753	71.8817
4000	79.7766	73.4244	74.2209
5000	78.9713	73.9578	73.6349
7000	79.9132	72.714	74.1922
10000	79.8845	74.362	75.4909
12000	80.2248	74.7088	74.5749
14000	79.9444	73.9578	75.29

Table 2 shows the performance of in-domain testing, out-of-domain without self-training, and out-of-domain with self-training for multiple seed sizes (from 1000 to 21000) using Brown as source and WSJ as target. Like in the previous set of experiments using WSJ as source and Brown as target, the performance drops from in-domain testing to out-of-domain testing, ranging from 3% to 5%.

That the performance drops from in-domain testing to out-of-domain testing is expected because the topics of the WSJ corpus are different from the topics of the Brown corpus. Consequently, there are differences in the vocabulary and sentence structures. To obtain a parser with high accuracy, the parser must be trained in various topics. Training a parser in a certain domain and then testing it on another domain will not yield high accuracy.

Table 2: Labeled Attachment Scores (LAS) using Brown as source and WSJ as target

Seed size	In-domain	No self-training	With self-training
1000	72.8623	65.847	69.1473
2000	74.8427	70.0724	69.9621
3000	75.6942	70.2713	71.7716
4000	75.9741	69.7464	72.0089
5000	76.1343	71.5847	71.2587
7000	76.4668	70.1946	72.9316
10000	77.3637	71.7621	72.831
13000	76.2946	71.7549	73.1833
17000	78.507	72.6992	73.6267
21000	78.4855	71.7237	73.6267

4.2 Unsupervised Domain Adaptation's Impact on Performance

Unsupervised domain adaptation improves the performance on out-of-domain testing. We can see that figures 1 and 2 show improvements from experiments without self-training to experiments with self-training. There are some occasions, though, where no self-training is better than with self-training.

Tables 1 and 2 shows the detailed results of all experiments. The experiments with WSJ as source and Brown as target show slight improvements, raning from 0% - 1.3%. The experiments with Brown as source and WSJ as target has the higher improvements, ranging from 1% to 4%.

4.3 Increasing the Seed Size and Self-Training Sets

Increasing the seed size improves the LAS for any scenario. We see in figures 1 and 2 that the learning curves go upward. The reason is that the larger the seed size is, the more various sentences the parser learns. Therefore, it knows a lot more sentences and performs better in testing.

On the other hand, increasing the self-training sets reduces the LAS using either corpus as sources and targets. The reason is simple: at first, we train the parser on the seed, which contains 10000 sentences. When we test the parser on 1000 sentences for self-training, the accuracy is better than when we test it on more sentences, say 2000 or 3000. As the self-training set size increases, the accuracy drops. Figures 3 and 4 show the performance of the parser trained on increasing self-training sets.

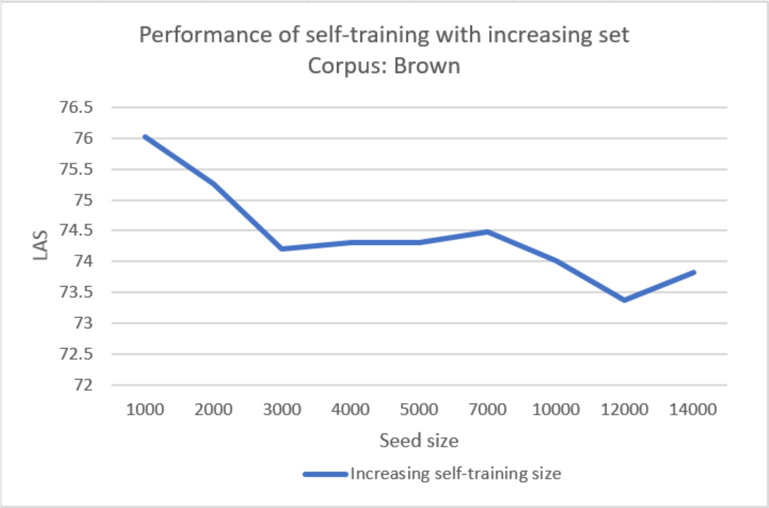

Figure 3: Increasing self-training set size for Brown corpus

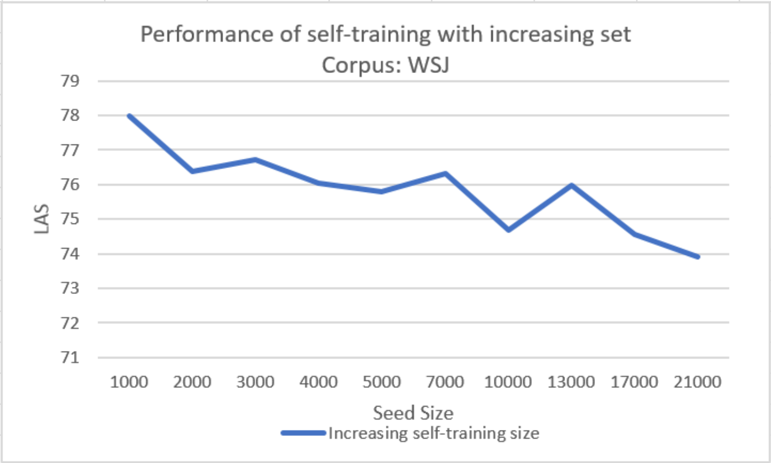

Figure 4: Increasing self-training set size for WSJ corpus

4.4 Inverting Source and Target

Inverting the source and target does not change the way the experiments behave. In other words, in-domain testing always produces the best results, self-training improves the out-of-domain testing, and increasing self-training sets lowers the performance of the parser. The difference between the first set of experiments (using WSJ as source and Brown as target) and the second set of experiments (using Brown as source and WSJ as target) is the amount of increase or decrease in LAS.

Table 3: Comparison to Reichart and Rapoport's Experiments

	No self-training		With self-training	
Seed size	Mine	R & R	Mine	R & R
1000	69.4%	<70%	69.1%	∼73%
2000	71.8%	>70%	71.4%	∼75%

For the in-domain experiments, training and testing on WSJ produces higher LAS than training and testing on Brown from 1.5% to 3.5%. I think that this is because WSJ has similar topics, only covering business, economics, and politics. Meanwhile, Brown has wider topics, covering sports, government, science, medicine, etc.

For the out-of-domain experiments, training and testing on WSJ also produces higher LAS than training and testing on Brown. However, the improvement from no self-training experiments to self-training experiments in WSJ is very slight, as seen in Figure 1. This could be due to randomization in the Stanford Dependency Parser algorithm, though, since I only ran the experiments once for each scenario.

4.5 Results Compared to Reichart and Rapoport's Experiments

This assignment is comparable to Reichart and Rapoport's OI (outside-inside) experiment. Their OI experiment is so called because the seed, WSJ, is ”outside” the testing set domain, which is Brown, while the self-training set, Brown, is ”inside” the testing set domain.

Their OI scenario was training the parser on WSJ sections 2-21, self-training on Brown training section, and testing on Brown testing section. The Brown training section consisted of 90% of sentences in the Brown corpus across all topics, and the Brown testing section consisted of the remaining 10%. They mentioned that they also performed their experiments with Brown and WSJ trading places with similar results, which were not discussed in their paper. Since Reichart and Rapoport did not use dependency parser, they measured their performance using precision/recall and F-score.

Figure 3 in their paper shows the performance of their OI experiments. As in the previous comparison, I only compare my experiments with 1000 and 2000 sentences as the seed sizes. Their baseline experiment (no self-training) with 1000 sentences yielded below 70% F-score. My experiment yields 69.4% LAS. For 2000-sentence baseline experiment, Reichart and Rapoport recorded above 70% F-score, whereas I achieved 71.8% LAS. For the self-training experiments, Reichart and Rapoport reached around 73% F-score, while I got 69.1% LAS for 1000 sentences. For 2000 sentences, they got 75% F-score, while I got 71.4% LAS.

Table 3 shows the summary of comparison between my experiments and their experiments. My experiment yielded lower LAS than their F-scores. I suspect this is due to a difference in the evaluation measurement, since LAS measurement is not exactly comparable to F-score measurement. In addition, the seed data they used is slightly different from the seed data I used. They used random sentences from WSJ sections 2-21, while I used consecutive sentences from WSJ sections 2-10. Lastly, we used different parsers. They used Collins parser, while I used Stanford parser.