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

Manu Agarwal
UTEID: ma53767

April 5, 2016

Introduction

Statistical Parsing is a group of parsing methods that use a probabilistic model of syntax to assign probabilities to different grammatical productions. This gives us the relative frequency of different grammatical productions which in turn can be used to deduce the probability of a complete parse tree for a sentence. However, a large amount of annotated data is required to train such parsers. An alternative is to train on a different domain for which the training data is available in plenty and then test on the target domain. The performance of such parsers, however, declines when training and testing on different domains.

The task of domain adaptation or transfer learning aims to transfer knowledge learnt from a source domain to a target domain. In this assignment, we train a statistical parser on the source domain for which huge amounts of annotated training data is available and use this trained parser to annotate data in the target domain. This newly annotated data is then used to retrain the parser. For the purposes of this experiment, we just use one iteration of retraining. Ideally, we would like to redo this process until convergence.

Experiments

We make use of the Stanford parser for the purposes of this assignment. We first train the parser on the Wall Street Journal (WSJ) corpus and test it on the the WSJ corpus itself. Then, we test this trained parser on Brown corpus to see how much the performance drops. Further, we do domain adaptation on the Brown corpus and see if the performance of testing on Brown corpus improves. We also invert the source and the target domain and see if it has any impact on the performance of the parser.

We use MemoryTreebanks for implementation purposes. The Brown set is split into two - the first set contains the first 90% sentences from each genre and the second set contains the remaining 10% sentences from each genre. The sections 2-22 of WSJ are used for training (if training on WSJ) and the section 23 of WSJ is used for testing (if testing on WSJ). Arguments are passed during runtime specifying which corpus to train/self-train/test on and which portion of that corpus to use.

We see that the parser performs reasonably well when testing on the domain on which it is trained. The performance drops significantly when we test on a different domain (drops by 3 percentage points in the case of WSJ and almost 7 percentage points in the case of Brown). Self-training on the target domain improves the performance quite a bit but is still not able to beat in domain testing performance which is understandable. Other observations are described in the subsequent

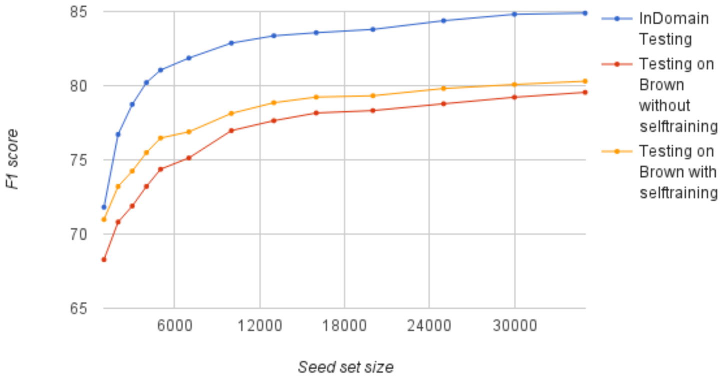

Figure 1: Plot of the F1 score for different values of seed set size. The source corpus is WSJ

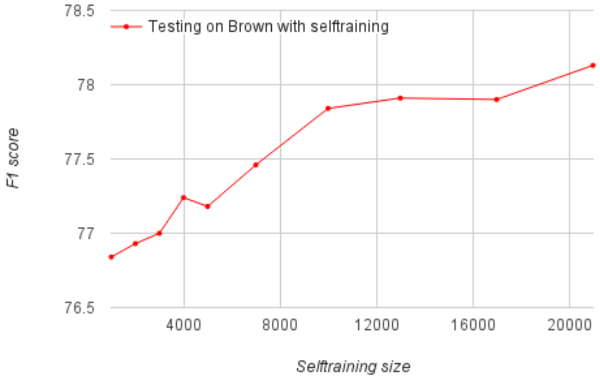

Figure 2: Plot of the F1 score for different values of self-training set size. The source corpus is WSJ

sections. The complete statistics of the results are available in the form of a spreadsheet here.

How much does performance drop from in-domain testing to out-of-domain testing?

Figure 1 shows the performance of the parser in terms of F1 score when training the parser initially on sections 2-22 of the WSJ corpus. We see that there is a drop of almost 3 percentage points in the F1 score when going from in-domain testing to out-of-domain testing(on Brown) if the seed set is 1000. This difference becomes more prominent as we increase the size of the seed set. The difference becomes almost 6 percentage points when the size of the seed set is 7000. If we increase the size of the training set further, the difference between the in-domain F1 score and the out-of-domain F1 score decreases and becomes almost 5 percentage points when the size of the seed set is 35000. This is understandable as initially the parser adapts to the domain much faster and hence the difference between in-domain and out-of-domain F1 score keeps on increasing. This performance drop is attributable to the fact that WSJ and Brown are two pretty different corpora. Still, our parser is able to achieve a decent accuracy (almost 80%) when the size of the seed set is 35000.

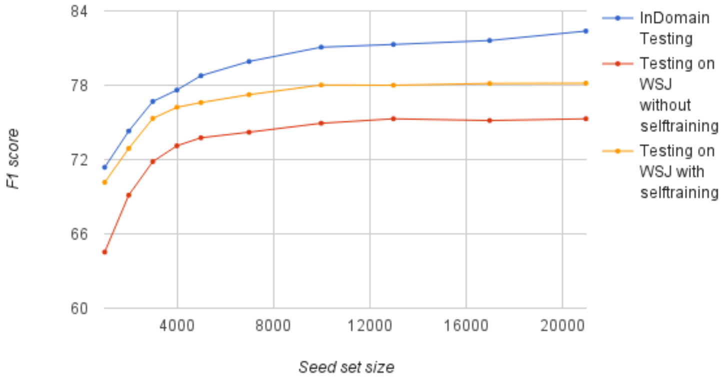

Figure 3: Plot of the F1 score for different values of seed set size. The source corpus is Brown

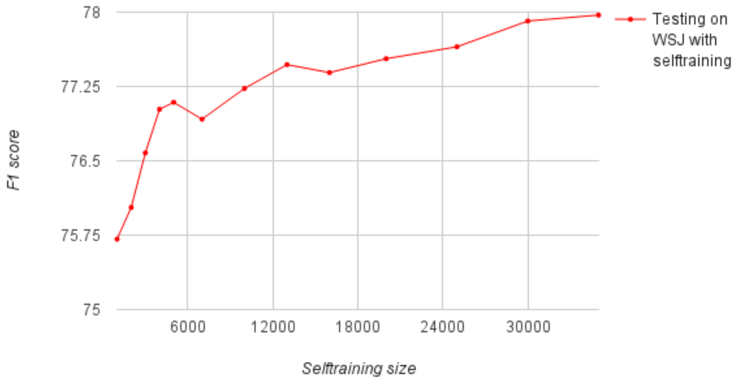

Figure 4: Plot of the F1 score for different values of self-training set size. The source corpus is Brown

How does unsupervised domain adaptation impact performance on out-of-domain testing?

Figure 1 shows that unsupervised domain adaptation does improve the performance of the parser on out-of-domain testing. There is an improvement of almost 2.5% points over out-of-domain testing without self-training when the size of the seed set is 1000. Increasing the size of the seed set improves the overall accuracy of the parser but the difference between out-of-domain testing with self-training and without-self-training becomes less prominent. The difference becomes almost 1.3% points when the size of the seed set is 35000. Initially, when the seed set was pretty small, the size of the self-training set was significant and hence the improvement was more. As the seed set becomes much larger than the self-training set, the difference between out-of-domain testing with self-training and without-self-training drops.

How does increasing the size of seed and self-supervised training sets affect the relative performance?

Figure 1 and Figure 2 show that the performance increases as we increase the size of the seed set and the self-training set. Figure 1 tells us that the performance increases sharply (by almost 10 percentage points within a span of increasing seed set from 1000 to 5000) in the initial stages for all the three cases (in-domain testing, out-of-domain testing without self-training and out-of-domain testing with self-training) and then slows down. However, Figure 2 tells us that there is an almost linear increase in performance till the size of the self-training set reaches 10000 after which the F1 score increases slowly. However, it is interesting to observe that the curve in Figure 2 has not converged as in the case of Figure 1 and that there is still scope of improvement given more data for self-training. This shows that increasing the size of the seed set has more impact in the initial stages (when the size of the seed set was small) but increasing the size of the self-training set has more impact in the later stages (when the parser has been trained on a sufficiently large seed set).

How does inverting the ”source” and ”target” impact your results and why?

Figure 3 and Figure 4 show the gradation in F1 score when we trained the parser initially on the Brown corpus. Comparing these plots with Figure 1 and Figure 2 tells us that self-training helps much more (jump of almost 6 percentage points from testing without self-training on WSJ to testing with self-training on WSJ) when we are training our parser using Brown as the source set and WSJ as the target set as compared to using WSJ as the source and Brown as the target. The drop in performance from in-domain testing to out-of-domain testing is also more significant in the case of using Brown as the source set and WSJ as the target set(almost 7 percentage point drop in the F1 score when the seed set is 1000). This seems legitimate as Brown is a pretty diverse corpus with data from different genres. However, WSJ is a pretty skewed set of articles with focus on news from the financial domain. Thus, when moving from Brown to WSJ, the performance drops much more. Figure 2 and Figure 4 are pretty similar and tell us that increasing the size of the self-training set impacts both the cases (WSJ as source and Brown as target; Brown as source and WSJ as target) in quite a similar fashion.

How do your results compare to the results described in the Reichart and Rappoport paper for the OI setting?

Comparing our results to those in the Reichart and Rappoport paper, we see that our curves are pretty similar in shape to the plots for F1 in the paper, the difference being they vary the seed set size from 10 to 2000 while we vary it from 1000 to 35000 when using WSJ as the seed and 1000 to 21000 when using Brown as the seed. Comparing the performance gain when moving from out-of-domain testing without self-training to out-of-domain testing with self-training, we see that it is almost 3.5 percentage points when the size of the seed set is 2000 for both our results and the results in the paper. This also demonstrates that both parsers - Collins and Stanford parsers work well for these experiments.

Conclusion

The results of the experiments demonstrate that self-training is a good way to achieve good performance when we do not have much labeled data for the target domain. In fact, if we do not have any labeled data for the target domain, unsupervised domain adaptation also provides a nice work around where the parser is trained on a source corpus and then self-trained on the target corpus. Recently, a generalization of this technique called fine-tuning has been employed to a plethora of deep learning techniques and has provided significant improvement in results when the labeled data for the target task is small.