Domain Adaptation by Active Learning

Giuseppe Attardi, Maria Simi, Andrea Zanelli

Università di Pisa, Dipartimento di Informatica, Largo B. Pontecorvo 3,
56127 Pisa, Italy
{attardi, simi, andreaz}@di.unipi.it

Abstract. We tackled the Evalita 2011 Domain Adaptation task with a strategy of active learning. The DeSR parser can be configured to provide different measures of perplexity in its own ability to parse sentences correctly. After parsing sentences in the target domain, a small number of the sentences with the highest perplexity were selected, revised manually and added to the training corpus in order to build a new parser model incorporating some knowledge from the target domain. The process is repeated a few times for building a new training resource partially adapted to the target domain. Using the new resource we trained three stacked parsers, and their combination was used to produce the final results.

Keywords: Dependency parser, domain adaptation, active learning, parser combination.

1 Introduction

Active learning (AL) was used as a strategy for domain adaptation. The active learning process aims at reducing the human annotation effort, only asking for advice when the utility of the new example is high. AL requires identifying a criterion for selecting new training examples to add to the training corpus that would maximize the learning rate, limiting the costs of annotating the additional examples for training. The primary question is therefore *query* formulation: how to choose which example (or examples) to try next. A separate issue, which influences the speed and performance of the active learning process, is the number of training instances to be added at each iteration. Adding one instance at a time slows the overall learning process down. If, on the other hand, a batch of instances is added, the learning progresses faster, but it becomes more difficult to find strategies for selecting a good batch.

In previous experiments, we applied active learning to the problem of learning how to parse questions, given a training corpus with a few instances of questions [1]. In that context we explored various possible metrics for scoring the examples to be selected and compared their efficacy with respect to random selection. A simple strategy that proved quite valuable was to use the score that the parser itself provides

as a measure of the perplexity in parsing a sentence. The DeSR parser can be configured to provide several measures of perplexity, including the minimum, maximum and overall likelihood of each sentence. In the adaptation phase, the DeSR parser was used with the MLP algorithm (Multi Layer Perceptron) for speeding up the process of parsing the large collection from the target domain, assuming that accuracy would not be an issue for the purpose of selecting the best training examples.

Among the possible measures computed by the parser, we chose to use the overall likelihood measure to rank the sentences in the target domain. For AL we chose a small number of the sentences with the highest perplexity, we revised them manually and added them to the training corpus in order to build a new parser model incorporating some knowledge from the target domain. The process could then be iterated a few times. Each time the new parser was tested on the target domain to check the improvements and the process was repeated again using the new parser for selecting new training examples.

At the end of the AL process a combination of three parsers with SVM (Support Vector Machine) as classifier was used to obtain more accurate results.

2 Description of the System

DesR is a transition-based parser [2], which uses a classifier to decide which parsing action to perform. The classifier computes a probability distribution for the possible actions to perform at each step. Given a parsed sentence, the probability of each parsing step is therefore available to compute different metrics by which to estimate the confidence of the parser in its own output. For example:

a. *Likelihood of a parse tree,* computed as the product of the probabilities of all the steps used in building the tree;
b. *Average probability* of the parsing steps in building the tree.

For the Evalita 2011 task, we selected sentences according to *Lowest likelihood* of sentence parse tree, which amounts to preferring sentences that were judged more difficult by the parser.

For the configuration of DeSR we used the configurations that gave the best results on the Pilot Subtask of Evalita 2009 Dependency Parsing Track [3] with the ISST corpus, that is the current source domain corpus. The best parser obtained with the MLP classifier was used for the whole process of domain adaptation, while more performing parser combinations based on the SVM classifier were used to monitor the progress of adaptation and for producing a more accurate final result.

2.1 Active Learning

We performed three steps of active learning. Each time we selected a small number of sentences to revise manually (between 50 and 100) and limited their length to a reasonable number of tokens (within 40 tokens). These constraints were suggested by previous experience with Active Learning and compatibility with the amount of resources available (in terms of annotator's time).

In addition to the scored parser output and the filter on maximum sentence length, manual intervention was needed at each step to discard noisy sentences. The target domain corpus in fact contains many useless sentences, such as sentences that contain only punctuation, sentences derived from lists or tables, or sentences in a language different from that of the target domain. All these sentences are of course the ones leading to lowest confidence scores but are to be excluded from the adaptation process, since they are not good representatives of target domain texts.

(*Step 0*) Before adaptation, the parser trained on the given training corpus consisting of 3275 sentences in the source domain achieved 79.96% LAS (Labeled Attachment Score) on the source domain development set and 74.82% LAS on the target domain development set.

(*Step 1*) The target domain corpus was parsed with the parser built at the Step 0 and 50 sentences, with a maximum length of 20 tokens, were selected among those with worst score. These were manually revised and added to the training corpus. A new parser was built achieving 78.94% LAS on the source domain development set and 75.26% LAS on the target domain development set.

(*Step 2*) The target domain corpus was parsed with the new parser and 60 sentences with a maximum length of 20 tokens and other 60 sentences with a maximum length of 40 tokens were selected among those with worst score. 89 of these were manually revised and added to the training corpus. The new parser scored 78.59% LAS on the source domain development set and 78.14% on the target domain development set.

(*Step 3*) Again, the target domain corpus was parsed and 50 sentences with a maximum length of 40 tokens were selected among those with worst score. These were manually revised and added to the training corpus. A new parser was build achieving 78.51% of LAS on the source domain development set and 79.48% on the target domain development set.

Table 1. Results of three steps of active learning.

Step	*Training Corpus*	*LAS of the MLP Parser*		*LAS of the Parsers Combination*	
		Source Dev Set	*Target Dev Set*	*Source Dev Set*	*Target Dev Set*
0	Source domain training set	79.96 %	74.82 %	82.09 %	76.28 %
1	+ 50 revised sentences	78.94 %	75.26 %	81.80 %	79.34 %
2	+ 89 revised sentences	78.59 %	78.14 %	81.70 %	82.36 %
3	+ 50 revised sentences	78.51 %	79.48 %	81.92 %	82.55 %

The MLP classifier, used in the AL process for parsing the target domain texts, is efficient but not the best performing parser. A combination of parsers with SVM

allows better results and will be used in the end. Table 1 reports the performance progress in the three steps of AL along with intermediate results of the parser combination while the training set is expanded to cover more cases in the target domain.

Fig. 1 compares graphically the performances (in terms of LAS score) of the best parser combination (the green line) and the MLP based parser (the blue line) during the three steps of AL. Both parsers increase their ability to parse sentences from the target domain as new examples are added, while the performance of the parser combination on the source domain (yellow line) remains almost stable. At the end, the parser combination performs slightly better on the target domain than on the source domain. We can also note that the last step of active learning, while still effective in improving the performance of the poor parser, has a limited impact on the best performing parser. This can be taken as an indication that we can stop adaptation.

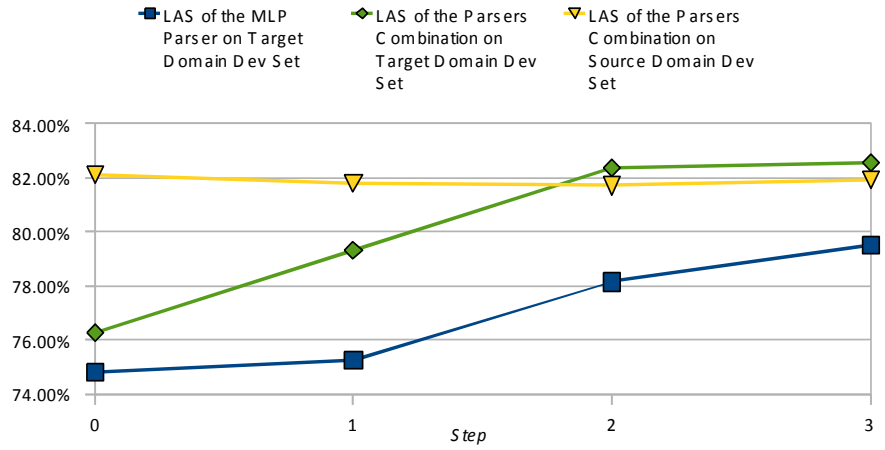

Fig. 1. The graph shows the improvement brought by the adaptation process on the target domain against a very slight deterioration of the performance on the source domain.

2.2 Parser Combination

The best results were obtained with the combination of three different configurations of the DeSR parser [4].

All three configurations were two stage Reverse Revision parsers [3], i.e. a stacked Right-to-Left parser that uses hints produced by a first pass Left-to-Right parser. The first pass uses a lower accuracy Maximum Entropy classifier, in order to produce a larger number of incorrect results that become a useful source of training for the second corrective stage, which uses a more accurate SVM classifier.

The differences among the three configurations used by the parsers are summarized in the following table.

Table 2. Configurations used by the three parsers.

Feature	**Vers. 2**	**Vers 6**	**Vers. 8**
Notice non word in children	True	False	True
Note type of entities in children	False	True	True
Keep count of previous verbs	True	True	False
MaxEntropy iterations	60	60	50

The first configuration option concerns a feature which takes into account the presence of punctuation symbols among the dependent tokens; the second options takes into account whether among children there are terms expressing time or location; the third option whether the parser should keep count of previous verbs in a sentence; finally the fourth configuration option concerns the maximum number of iterations to be performed by the maximum entropy classifier used in the first pass.

3 Results

For the open task (the first subtask) we used the same configuration used for the closed task (the combination of three SVM parsers); the only difference was the addition of the target domain development set to the final training corpus.

Table 3. Results of the two runs for Evalita 2011 Domain Adaptation

Task	*Adaptation*	*LAS on Source Domain Dev Set (Training corpus)*	*LAS on Test Set (Training corpus)*
Open task (1)	Before adaptation	82.09% (Source domain training set)	80.29% (Source domain training set + Target domain development set)
	After adaptation	82.34% (Source domain training set + Target domain development set + Revised sentences)	**81.39%** (Source domain training set + Target domain development set + Revised sentences)
Closed task (2)	Before adaptation	82.09% (Source domain training set)	75.85% (Source domain training set)
	After adaptation	81.92% (Source domain training set + Revised sentences)	**80.83%** (Source domain training set + Revised sentences)

Table 3 reports the results achieved on the test set of Evalita 2011 Domain Adaptation Task for both subtasks, before and after the process of adaptation, along with the results achieved on the source domain.

4 Discussion

In both the subtasks the adaptation led to an improvement on the target domain without affecting the performance on the source domain.

A small initial test, which, however, did not give encouraging results, was done by trying a strategy of *self-training*. We trained the parser on the source domain training set, we parsed the target domain corpus with the MLP parser emitting confidence scores. Then we extracted the 100 sentences with more confidence, with a minimum length of 10 tokens, and added them to the training corpus. A new parser was built on this training corpus and was tested on the target domain achieving 74.84 % of LAS. Without the adaptation, the parser obtained 74.82 % of LAS. The small improvement brought by the self learning (0.02 %) is really not significant, especially if compared with the result of the first step of the active learning process that is 75.26 % of LAS with the same configuration.

References

1. Atserias, J., Attardi, G., Simi, M., Zaragoza, H.: Active Learning for Building a Corpus of Questions for Parsing. Proc. of LREC 2010, Malta (2010)
2. Attardi, G.: Experiments with a Multilanguage Non-Projective Dependency Parser, In: Tenth Conference on Natural Language Learning, New York, (NY) (2006)
3. Attardi, G., Dell'Orletta, F., Simi, M., Turian, J.: Accurate Dependency Parsing with a Stacked Multilayer Perceptron. In: Workshop Evalita 2009, Reggio-Emilia, Italy ISBN 978-88-903581-1-1 (2009)
4. Attardi, G., Dell'Orletta, F.: Reverse Revision and Linear Tree Combination for Dependency Parsing. In: NAACL HLT 2009 (2009)