# Dependency Parsing domain adaptation using transductive SVM

**Antonio Valerio Miceli-Barone**
University of Pisa, Italy /
Largo B. Pontecorvo, 3, Pisa, Italy
`miceli@di.unipi.it`

**Giuseppe Attardi**
University of Pisa, Italy /
Largo B. Pontecorvo, 3, Pisa, Italy
`attardi@di.unipi.it`

## Abstract

Dependency Parsing domain adaptation involves adapting a dependency parser, trained on an annotated corpus from a given domain (e.g., newspaper articles), to work on a different target domain (e.g., legal documents), given only an unannotated corpus from the target domain.
We present a shift/reduce dependency parser that can handle unlabeled sentences in its training set using a transductive SVM as its action selection classifier.
We illustrate the the experiments we performed with this parser on a domain adaptation task for the Italian language.

## 1 Introduction

Dependency parsing is the task of identifying syntactic relationships between words of a sentence and labeling them according to their type. Typically, the dependency relationships are not defined by an explicit grammar, rather implicitly through a human-annotated corpus which is then processed by a machine learning procedure, yielding a parser trained on that corpus.
Shift-reduce parsers (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004; Attardi, 2006) are an accurate and efficient (linear complexity) approach to this task: They scan the words of a sentence while updating an internal state by means of shift-reduce actions selected by a classifier trained on the annotated corpus.
Since the training corpora are made by human annotators, they are expensive to produce and are typically only available for few domains that don't adequately cover the whole spectrum of the language. Parsers typically lose significant accuracy when applied on text from domains not covered by their training corpus. Several techniques have been proposed to adapt a parser to a new domain, even when only unannotated samples from it are available (Attardi et al., 2007a; Sagae and Tsujii, 2007).
In this work we present a domain adaptation based on the semi-supervised training of the classifier of a shift-reduce parser. We implement the classifier as a multi-class SVM and train it with a transductive SVM algorithm that handles both labeled examples (generated from the source-domain annotated corpus) and unlabeled examples (generated from the the target-domain unannotated corpus).

## 2 Background

### 2.1 Shift-Reduce Parsing

A shift-reduce dependency parser is essentially a pushdown automaton that scans the sentence one token at a time in a fixed direction, while updating a stack of tokens and also updating a set of directed, labeled edges that is eventually returned as the dependency parse graph of the sentence.
Let $T$ be the set of input token instances of the sentence and $D$ be the set of dependency labels. The state of the parser is defined by the tuple $\langle s, q, p\rangle$, where $s \in T^*$ is the stack, $q \in T^*$ is the current token sequence and $p \in \{E | E \subseteq 2^{T\times T\times D}, E \text{ is a forest}\}$ is the current parse graph.
The parser starts in the state $\langle [], q_0, \{\}\rangle$, where $q_0$ is the input sentence, and terminates whenever it reaches a state in the form $\langle s, [], p\rangle$. At each step,

it performs one of the following actions:

$$\begin{aligned}
\texttt{shift} &: \frac{\langle s,\,[t|q],\,p\rangle}{\langle [t|s],\,q,\,p\rangle}\\
\texttt{rightreduce}_{\texttt{d}} &: \frac{\langle [u|s],\,[t|q],\,p\rangle}{\langle s,\,[t|q],\,p\cup\{(u,\,t,\,d)\}\rangle}\\
\texttt{leftreduce}_{\texttt{d}} &: \frac{\langle [u|s],\,[t|q],\,p\rangle}{\langle s,\,[u|q],\,p\cup\{(t,\,u,\,d)\}\rangle}
\end{aligned}$$

note that there are $rightreduce_d$ and $leftreduce_d$ actions for each label $d \in D$.
Action selection is done by the combination of two functions $f \circ c$ : a feature extraction function $f\, :\, States \rightarrow \mathbb{R}^n$ that computes a (typically sparse) vector of numeric features of the current state and the multi-class classifier $c : \mathbb{R} \rightarrow Actions$. Alternatively, the classifier could score each available action, allowing a search procedure such as best-first (Sagae and Tsujii, 2007) or beam search to be used.
In our experiments we used an extension of this approach that has an additional stack and additional actions to handle non-projective dependency relationships (Attardi, 2006). Training is performed by computing, for each sentence in the annotated training corpus, a sequence of states and actions that generates its correct parse, yielding, for each transition, a training example $(x,\, y) \in \mathbb{R}^n \times Actions$ for the classifier.
Various classification algorithms have been successfully used, including maximum entropy, multi-layer perceptron, averaged perceptron, SVM, etc. In our approach, the classifier is always a multi-class SVM composed of multiple (one-per-parsing-action) two-class SVMs in one-versus-all configuration.

## 2.2 Parse Graph Revision

Attardi and Ciaramita (2007b) developed a method for improving parsing accuracy using parse graph revision: the output of the parser is fed to a procedure that scans the parsed sentence in a fixed direction and, at each step, possibly revises the current node (rerouting or relabeling its unique outgoing edge) based on the classifier's output.
Training is performed by parsing the training corpus and comparing the outcome against the annotation: for each sentence, a sequence of actions necessary to transform the machine-generated parse into the reference parse is computed and it is used to train the classifier. (Usually, a lower-quality parser is used during training, assuming that it will generate more errors and hence more revision opportunities).
This method tends to produce robust parsers: errors in the first stage have the opportunity to be corrected in the revision stage, thus, even if it does not learn from unlabeled data, it nevertheless performs well in domain adaptation tasks (Attardi et al., 2007a). In our experiments we used parse graph revision both as a baseline for accuracy comparison, and in conjunction with our approach (using a transductive SVM classifier in the revision stage).

## 2.3 Transductive SVM

Transductive SVM (Vapnik, 1998) is a framework for the semi-supervised training of SVM classifiers.
Consider the inductive (completely supervised) two-class SVM training problem: given a training set $\{(x_i,\, y_i)\, |x_i \in \mathbb{R}^n,\, y_i \in \{-1,\, 1\}\}_{i=1}^{L}$, find the maximum margin separation hypersurface $w \cdot \phi\,(x) + b = 0$ by solving the following optimization problem:

$$\arg \min_{w,\, b,\, \xi} \frac{1}{2}\, \|w\|_2^2 + C \sum_{i=1}^{L} \xi_i \qquad (1)$$

$$\begin{aligned}
\forall i &: \; y_i\, w \cdot \phi\,(x) + b \geq 1 - \xi_i\\
\forall i &: \; \xi_i \geq 0\\
& \quad w \in \mathbb{R}^m,\, b \in \mathbb{R}
\end{aligned}$$

where $C \geq 0$ is a regularization parameter and $\phi(\cdot)$ is defined such that $k\,(x,\, \hat{x}) \equiv \phi(x) \cdot \phi\,(\hat{x})$ is the SVM kernel function. This is a convex quadratic programming problem that can be solved efficiently by specialized algorithms.
Including an unlabeled example set $\left\{x_j^* | x_j^* \in \mathbb{R}^n\right\}_{j=1}^{L^*}$ we obtain the transductive SVM training problem:

$$\arg \min_{w,\, b,\, \xi,\, y^*,\, xi^*} \frac{1}{2}\, \|w\|_2^2 + C \sum_{i=1}^{L} \xi_i + C^* \sum_{j=1}^{L^*} \xi_j^* \qquad (2)$$

$$\begin{aligned}
\forall i &: y_i\, w \cdot \phi\left(x_i\right) + b \geq 1 - \xi_i \\
\forall j &: y_j^*\, w \cdot \phi\left(x_j^*\right) + b \geq 1 - \xi_j^* \\
\forall i &: \xi_i \geq 0 \\
\forall j &: \xi_j^* \geq 0 \\
\forall j &: y_j^* \in \{-1,\, 1\} \\
& \quad w \in \mathbb{R}^m,\, b \in \mathbb{R}
\end{aligned}$$

This formulation essentially models the unlabeled examples the same way the labeled examples are modeled, with the key difference that the $y_j^*$ (the unknown labels of the unlabeled examples) are optimization variables rather than parameters. Optimizing over these discrete variables makes the problem non-convex and in fact NP-hard. Nevertheless, algorithms that feasibly find a local minimum that is typically good enough for practical purposes do exist. In our experiments we used the iterative transductive SVM algorithm implemented in the SvmLight library (Joachims, 1999). This algorithm tends to become impractical when the number of unlabeled examples is greater than a few thousands, hence we were forced to use only a small portion on the available target domain corpus. We also tried the concave-convex procedure (CCCP) TSVM algorithm (Collobert et al., 2006) as implemented by the the Universvm package, and the multi-switch and deterministic annealing algorithms for linear TSVM (Sindhwani and Keerthi, 2007) as implemented by the Svmlin package. These methods are considerably faster but appear to be substantially less accurate than SvmLight on our training data.

## 3 Proposed approach

We present a semi-supervised training procedure for shift/reduce SVM parsers that allows to include unannotated sentences in the training corpus.

We randomly sample a small number (approx. 100) of sentences from the unannotated corpus (the target domain corpus in a domain adaptation task). For each of these sentences, we generate a sequence of states that the parser may encounter while scanning the sentence. For each state we extract the features to generate an unlabeled training example for the SVM classifier which is included in the training set along with the labeled examples generated from the annotated corpus. There is a caveat here: the parser state at any given point during the parsing of a sentence generally depends on the actions taken before, but when we are training on an unannotated sentence, we have no way of knowing what actions the parser should have taken, and thus the state we generate can be generally incorrect. For this reason we evaluated pre-parsing the unannotated sentences with a non-transductively trained parser in order to generate plausible state transitions while still adding unlabeled examples. However, it turned out that this pre-parsing does not seem to improve accuracy. We conjecture that, because the classifier does not see actual states but only features derived from them, and many of these features are independent of previous states and actions (features such as the lemma and POS tag of the current token and its neighbors have this property), these features contain enough information to perform parsing.

The classifier is trained using the SvmLight transductive algorithm. Since SvmLight supports only two-class SVMs while our classifier is multi-class (one class for each possible parsing action), we implement it in terms of two-class classifiers. We chose the one-versus-all strategy:

We train a number of sub-classifiers equal to the number of original classes. Each labeled training example $(x,\, y)$ is converted to the example $(x,\, 1)$ for the sub-classifier number $y$ and to the example $(x,\, -1)$ for the rest of sub-classifiers. Unlabeled examples are just replicated to all sub-classifiers. During classification the input example is evaluated by all the sub-classifiers and the one returning the maximum SVM score determines the class.

Our approach has been also applied to the second stage of the revision parser, by presenting the features of the unannotated sentences to the revision classifier as unlabeled training examples.

## 4 Experiments

### 4.1 Experimental setup

We performed our experiments using the DeSR parser (Attardi, 2006) on the data sets for the Evalita 2011 dependency parsing domain adaptation task for the Italian language (Evalita, 2011). The data set consists in an annotated source-domain corpus (newspaper articles) and an unannotated target-domain corpus (legal documents),

plus a small annotated development corpus also from the target domain, which we used to evaluate the performance.
We performed a number of runs of the DeSR parser in various configurations, which differed in the number and type of features extracted, the sentence scanning direction, and whether or not parse tree revision was enabled. The SVM classifiers always used a quadratic kernel. In order to keep the running time of transductive SVM training acceptable, we limited the number of unannotated sentences to one hundred, which resulted in about 3200 unlabeled training examples fed to the classifiers. The annotated sentences were 3275.
We performed one run with 500 unannotated sentences and, at the cost of a greatly increased running time, the accuracy improvement was about 1%. We conjecture that a faster semi-supervised training algorithm could allow greater performance improvements by increasing the size of the unannotated corpus that can be processed. All the experiments were performed on a machine equipped with an quad-core Intel Xeon X3440 processor (8M Cache, 2.53 GHz) and 12 Gigabytes of RAM.

### 4.2 Discussion

As it is evidenced from the table in figure 1, our approach typically outperforms the non-transductive parser by about 1% of all the three score measures we considered. While the improvement is small, it is consistent with different configurations of the parser that don't use parse tree revision. Accuracy remained essentially equal or became slightly worse in the two configurations that use parse tree revision. This is possibly due to the fact that the first stage parser of the revision configurations uses a maximum entropy classifier during training that does not learn from the unlabeled examples.
These results suggest that unlabeled examples contain information that can exploited to improve the parser accuracy on a domain different than the labeled set domain. However, the computational cost of transductive learning algorithm we used limits the amount of unlabeled data we can exploit.
This is consistent with the results obtained by the self-training approaches, where a first parser is trained on a the labeled set, which is used to parse the unlabeled set which is then included into the training set of a second parser. (In fact, self-training is performed in the first step of the SvmLight TSVM algorithm).
Despite earlier negative results, (Sagae, 2010) showed that even naive self-training can provide accuracy benefits (about 2%) in domain adaptation, although these results are not directly comparable to ours because they refer to constituency parsing rather than dependency parsing. (McClosky et al., 2006) obtain even better results (5% f-score gain) using a more sophisticated form of self-training, involving n-best generative parsing and discriminative reranking. (Sagae and Tsujii, 2007) obtain similar gains (about 3 %) for dependency parsing domain adaptation, using self-training on a subset of the target-domain instances selected on the basis of agreement between two different parsers. (the results are not directly comparable to ours because they were obtained on a different corpus in a different language).

## 5 Conclusions and future work

We presented a semi-supervised training approach for shift/reduce SVM parsers and we illustrated an application to domain adaptation, with small but mostly consistent accuracy gains. While these gains may not be worthy enough to justify the extra computational cost of the transductive SVM algorithm (at least in the SvmLight implementation), they do point out that there exist a significant amount of information in an unannotated corpus that can be exploited for increasing parser accuracy and performing domain adaptation. We plan to further investigate this method by exploring classifier algorithms other than transductive SVM and combinations with other semi-supervised parsing approaches. We also plan to test our method on standardized English-language corpora to obtain results that are directly comparable to those in the literature.

## References

H. Yamada and Y. Matsumoto. 2003. *Statistical Dependency Analysis with Support Vector Machines.* Proceedings of the 9th International Workshop on Parsing Technologies.

J. Nivre and M. Scholz. 2004. *Deterministic Dependency Parsing of English Text.* Proceedings of COLING 2004.

G. Attardi. 2006. *Experiments with a Multilanguage*

Figure 1: Experimental results

Accuracy (-R: right-to-left, -rev: left-to-right with revision, -rev2: right-to-left with revision):

| | Transductive | | | Normal | | |
|---|---|---|---|---|---|---|
| Parser configuration | LAS | UAS | Label only | LAS | UAS | Label only |
| 6 | **74.3** | **77.0** | **87.5** | 73.1 | 75.5 | 86.7 |
| 6-R | **75.7** | **78.6** | **88.7** | 74.6 | 77.6 | 87.8 |
| 6-rev | **75.2** | **78.2** | **88.6** | 75.1 | 78.0 | 88.3 |
| 6-rev2 | 75.0 | 77.8 | 88.7 | **75.8** | **78.6** | 88.7 |
| 8 | **74.3** | **77.0** | **87.3** | 73.4 | 76.0 | 85.9 |
| 8-R | **75.7** | **78.6** | **88.7** | 75.3 | 78.3 | 88.1 |
| 2 | **74.7** | **77.4** | **87.4** | 73.1 | 75.8 | 86.5 |

Figure 2: Typical features (configuration 6).

Numbers denote offsets.

'FEATS' denotes rich morphological features (grammatical number, gender, etc).

| | |
|---|---|
| LEMMA | -2 -1 0 1 2 3 prev(0) leftChild(-1) leftChild(0) rightChild(-1) rightChild(0) |
| POSTAG | -2 -1 0 1 2 3 next(-1) leftChild(-1) leftChild(0) rightChild(-1) rightChild(0) |
| CPOSTAG | -1 0 1 |
| FEATS | -1 0 1 |
| DEPREL | leftChild(-1) leftChild(0) rightChild(-1) |

*Non-Projective Dependency Parser*. Proceedings of CoNNL-X 2006.

G. Attardi, A. Chanev, M. Ciaramita, F. Dell'Orletta and M. Simi. 2007. *Multilingual Dependency Parsing and domain adaptation using DeSR*. Proceedings the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Prague, 2007.

Kenji Sagae and Jun'ichi Tsujii. 2007. *Dependency parsing and domain adaptation with LR models and parser ensembles*. CoNLL Shared Task.

G. Attardi, M. Ciaramita. 2007. *Tree Revision Learning for Dependency Parsing*. Proc. of the Human Language Technology Conference 2007.

V. Vapnik. 1998. *Statistical Learning Theory*. Wiley.

Ronan Collobert and Fabian Sinz and Jason Weston and Lon Bottou and Thorsten Joachims. 2006. *Large Scale Transductive SVMs*. Journal of Machine Learning Research

Thorsten Joachims. 1999. *Transductive Inference for Text Classification using Support Vector Machines*. International Conference on Machine Learning (ICML), 1999.

Vikas Sindhwani and S. Sathiya Keerthi 2007. *Newton Methods for Fast Solution of Semisupervised Linear SVMs*. Large Scale Kernel Machines. MIT Press (Book Chapter), 2007

Kenji Sagae 2010. *Self-Training without Reranking for Parser Domain Adaptation and Its Impact on Semantic Role Labeling*. Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing. Uppsala, Sweden: Association for Computational Linguistics. p. 37-44

David McClosky, Eugene Charniak and Mark Johnson 2006. *Reranking and self-training for parser adaptation*. Proceeding ACL-44. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics

Evalita. 2011. *Domain Adaptation for Dependency Parsing*. .