In Proc. Tenth Conf. on Computational Natural Language Learning (CoNLL 2006).
Funded by Swiss NSF grant 200020-109685 and EU FP6 project TALK.

Porting Statistical Parsers with Data-Defined Kernels

Ivan Titov
University of Geneva
24, rue Général Dufour
CH-1211 Genève 4, Switzerland
ivan.titov@cui.unige.ch

James Henderson
University of Edinburgh
2 Buccleuch Place
Edinburgh EH8 9LW, United Kingdom
james.henderson@ed.ac.uk

Abstract

Previous results have shown disappointing performance when porting a parser trained on one domain to another domain where only a small amount of data is available. We propose the use of data-defined kernels as a way to exploit statistics from a source domain while still specializing a parser to a target domain. A probabilistic model trained on the source domain (and possibly also the target domain) is used to define a kernel, which is then used in a large margin classifier trained only on the target domain. With a SVM classifier and a neural network probabilistic model, this method achieves improved performance over the probabilistic model alone.

1 Introduction

In recent years, significant progress has been made in the area of natural language parsing. This research has focused mostly on the development of statistical parsers trained on large annotated corpora, in particular the Penn Treebank WSJ corpus (Marcus et al., 1993). The best statistical parsers have shown good results on this benchmark, but these statistical parsers demonstrate far worse results when they are applied to data from a different domain (Roark and Bacchiani, 2003; Gildea, 2001; Ratnaparkhi, 1999). This is an important problem because we cannot expect to have large annotated corpora available for most domains. While identifying this problem, previous work has not proposed parsing methods which are specifically designed for porting parsers. Instead they propose methods for training a standard parser with a large amount of out-of-domain data and a small amount of in-domain data.

In this paper, we propose using data-defined kernels and large margin methods to specifically address porting a parser to a new domain. Data-defined kernels are used to construct a new parser which exploits information from a parser trained on a large out-of-domain corpus. Large margin methods are used to train this parser to optimize performance on a small in-domain corpus.

Large margin methods have demonstrated substantial success in applications to many machine learning problems, because they optimize a measure which is directly related to the expected testing performance. They achieve especially good performance compared to other classifiers when only a small amount of training data is available. Most of the large margin methods need the definition of a kernel. Work on kernels for natural language parsing has been mostly focused on the definition of kernels over parse trees (e.g. (Collins and Duffy, 2002)), which are chosen on the basis of domain knowledge. In (Henderson and Titov, 2005) it was proposed to apply a class of kernels derived from probabilistic models to the natural language parsing problem.

In (Henderson and Titov, 2005), the kernel is constructed using the parameters of a trained probabilistic model. This type of kernel is called a data-defined kernel, because the kernel incorporates information from the data used to train the probabilistic model. We propose to exploit this property to transfer information from a large corpus to a statistical parser for a different domain. Specifically, we propose to train a statistical parser on data including the large corpus, and to derive the kernel from this trained model. Then this derived kernel is used in a large margin classifier trained on the small amount of training data available for the target domain.

In our experiments, we consider two different scenarios for porting parsers. The first scenario is the pure porting case, which we call "transferring". Here we only require a probabilistic model trained on the large corpus. This model is then reparameterized so as to extend the vocabulary to better suit the target domain. The kernel is derived from this reparameterized model. The second scenario is a mixture of parser training and porting, which we call "focusing". Here we train a probabilistic model on both the large corpus and the target corpus. The kernel is derived from this trained model. In both scenarios, the kernel is used in a SVM classifier (Tsochantaridis et al., 2004) trained on a small amount of data from the target domain. This classifier is trained to rerank the candidate parses selected by the associated probabilistic model. We use the Penn Treebank Wall Street Journal corpus as the large corpus and individual sections of the Brown corpus as the target corpora (Marcus et al., 1993). The probabilistic model is a neural network statistical parser (Henderson, 2003), and the data-defined kernel is a TOP reranking kernel (Henderson and Titov, 2005).

With both scenarios, the resulting parser demonstrates improved accuracy on the target domain over the probabilistic model alone. In additional experiments, we evaluate the hypothesis that the primary issue for porting parsers between domains is differences in the distributions of words in structures, and not in the distributions of the structures themselves. We partition the parameters of the probability model into those which define the distributions of words and those that only involve structural decisions, and derive separate kernels for these two subsets of parameters. The former model achieves virtually identical accuracy to the full model, but the later model does worse, confirming the hypothesis.

2 Data-Defined Kernels for Parsing

Previous work has shown how data-defined kernels can be applied to the parsing task (Henderson and Titov, 2005). Given the trained parameters of a probabilistic model of parsing, the method defines a kernel over sentence-tree pairs, which is then used to rerank a list of candidate parses.

In this paper, we focus on the TOP reranking kernel defined in (Henderson and Titov, 2005), which are closely related to Fisher kernels. The reranking task is defined as selecting a parse tree from the list of candidate trees $(y_1, \ldots, y_s)$ suggested by a probabilistic model $P(x, y|\hat{\theta})$, where $\hat{\theta}$ is a vector of model parameters learned during training the probabilistic model. The motivation for the TOP reranking kernel is given in (Henderson and Titov, 2005), but for completeness we note that the its feature extractor is given by:

$$\begin{aligned}\phi_{\hat{\theta}}(x, y_k) = \\ (v(x, y_k, \hat{\theta}), \tfrac{\partial v(x, y_k, \hat{\theta})}{\partial \theta_1}, \ldots, \tfrac{\partial v(x, y_k, \hat{\theta})}{\partial \theta_l}),\end{aligned} \quad (1)$$

where $v(x, y_k, \hat{\theta}) = \log P(x, y_k|\hat{\theta}) - \log \sum_{t \neq k} P(x, y_t|\hat{\theta})$. The first feature reflects the score given to (x, y_k) by the probabilistic model (relative to the other candidates for x), and the remaining features reflect how changing the parameters of the probabilistic model would change this score for (x, y_k).

The parameters $\hat{\theta}$ used in this feature extractor do not have to be exactly the same as the parameters trained in the probabilistic model. In general, we can first reparameterize the probabilistic model, producing a new model which defines exactly the same probability distribution as the old model, but with a different set of adjustable parameters. For example, we may want to freeze the values of some parameters (thereby removing them from $\hat{\theta}$), or split some parameters into multiple cases (thereby duplicating their values in $\hat{\theta}$). This flexibility allows the features used in the kernel method to be different from those used in training the probabilistic model. This can be useful for computational reasons, or when the kernel method is not solving exactly the same problem as the probabilistic model was trained for.

3 Porting with Data-Defined Kernels

In this paper, we consider porting a parser trained on a large amount of annotated data to a different domain where only a small amount of annotated data is available. We validate our method in two different scenarios, transferring and focusing. Also we verify the hypothesis that addressing differences between the vocabularies of domains is more important than addressing differences between their syntactic structures.

3.1 Transferring to a Different Domain

In the transferring scenario, we are given just a probabilistic model which has been trained on a large corpus from a source domain. The large corpus is not available during porting, and the small corpus for the target domain is not available during training of the probabilistic model. This is the case of pure parser porting, because it only requires the source domain parser, not the source domain corpus. Besides this theoretical significance, this scenario has the advantage that we only need to train a single probabilistic parser, thereby saving on training time and removing the need for access to the large corpus once this training is done. Then any number of parsers for new domains can be trained, using only the small amount of annotated data available for the new domain.

Our proposed porting method first constructs a data-defined kernel using the parameters of the trained probabilistic model. A large margin classifier with this kernel is then trained to rerank the top candidate parses produced by the probabilistic model. Only the small target corpus is used during training of this classifier. The resulting parser consists of the original parser plus a very computationally cheap procedure to rerank its best parses.

Whereas training of standard large margin methods, like SVMs, isn't feasible on a large corpus, it is quite tractable to train them on a small target corpus.[1] Also, the choice of the large margin classifier is motivated by their good generalization properties on small datasets, on which accurate probabilistic models are usually difficult to learn.

We hypothesize that differences in vocabulary across domains is one of the main difficulties with parser portability. To address this problem, we propose constructing the kernel from a probabilistic model which has been reparameterized to better suit the target domain vocabulary. As in other lexicalized statistical parsers, the probabilistic model we use treats words which are not frequent enough in the training set as 'unknown' words (Henderson, 2003). Thus there are no parameters in this model which are specifically for these words. When we consider a different target domain, a substantial proportion of the words in the target domain are treated as unknown words, which makes the parser only weakly lexicalized for this domain.

To address this problem, we reparameterize the probability model so as to add specific parameters for the words which have high enough frequency in the target domain training set but are treated as unknown words by the original probabilistic model. These new parameters all have the same values as their associated unknown words, so the probability distribution specified by the model does not change. However, when a kernel is defined with this reparameterized model, the kernel's feature extractor includes features specific to these words, so the training of a large margin classifier can exploit differences between these words in the target domain. Expanding the vocabulary in this way is also justified for computational reasons; the speed of the probabilistic model we use is greatly effected by vocabulary size, but the large-margin method is not.

3.2 Focusing on a Subdomain

In the focusing scenario, we are given the large corpus from the source domain. We may also be given a parsing model, but as with other approaches to this problem we simply throw this parsing model away and train a new one on the combination of the source and target domain data. Previous work (Roark and Bacchiani, 2003) has shown that better accuracy can be achieved by finding the optimal re-weighting between these two datasets, but this issue is orthogonal to our method, so we only consider equal weighting. After this training phase, we still want to optimize the parser for only the target domain.

Once we have the trained parsing model, our proposed porting method proceeds the same way in this scenario as in transferring. However, because the original training set already includes the vocabulary from the target domain, the reparameterization approach defined in the preceding section is not necessary so we do not perform it. This reparameter-

[1] In (Shen and Joshi, 2003) it was proposed to use an ensemble of SVMs trained the Wall Street Journal corpus, but we believe that the generalization performance of the resulting classifier is compromised in this approach.

ization could be applied here, thereby allowing us to use a statistical parser with a smaller vocabulary, which can be more computationally efficient both during training and testing. However, we would expect better accuracy of the combined system if the same large vocabulary is used both by the probabilistic parser and the kernel method.

3.3 Vocabulary versus Structure

It is commonly believed that differences in vocabulary distributions between domains effects the ported parser performance more significantly than the differences in syntactic structure distributions. We would like to test this hypothesis in our framework. The probabilistic model (Henderson, 2003) allows us to distinguish between those parameters responsible for the distributions of individual vocabulary items, and those parameters responsible for the distributions of structural decisions, as described in more details in section 4.2. We train two additional models, one which uses a kernel defined in terms of only vocabulary parameters, and one which uses a kernel defined in terms of only structure parameters. By comparing the performance of these models and the model with the combined kernel, we can draw conclusion on the relative importance of vocabulary and syntactic structures for parser portability.

4 An Application to a Neural Network Statistical Parser

Data-defined kernels can be applied to any kind of parameterized probabilistic model, but they are particularly interesting for latent variable models. Without latent variables (e.g. for PCFG models), the features of the data-defined kernel (except for the first feature) are a function of the counts used to estimate the model. For a PCFG, each such feature is a function of one rule's counts, where the counts from different candidates are weighted using the probability estimates from the model. With latent variables, the meaning of the variable (not just its value) is learned from the data, and the associated features of the data-defined kernel capture this induced meaning. There has been much recent work on latent variable models (e.g. (Matsuzaki et al., 2005; Koo and Collins, 2005)). We choose to use an earlier neural network based probabilistic model of parsing (Henderson, 2003), whose hidden units can be viewed as approximations to latent variables. This parsing model is also a good candidate for our experiments because it achieves state-of-the-art results on the standard Wall Street Journal (WSJ) parsing problem (Henderson, 2003), and data-defined kernels derived from this parsing model have recently been used with the Voted Perceptron algorithm on the WSJ parsing task, achieving a significant improvement in accuracy over the neural network parser alone (Henderson and Titov, 2005).

4.1 The Probabilistic Model of Parsing

The probabilistic model of parsing in (Henderson, 2003) has two levels of parameterization. The first level of parameterization is in terms of a history-based generative probability model. These parameters are estimated using a neural network, the weights of which form the second level of parameterization. This approach allows the probability model to have an infinite number of parameters; the neural network only estimates the bounded number of parameters which are relevant to a given partial parse. We define our kernels in terms of the second level of parameterization (the network weights).

A history-based model of parsing first defines a one-to-one mapping from parse trees to sequences of parser decisions, $d_1,..., d_m$ (i.e. derivations). Henderson (2003) uses a form of left-corner parsing strategy, and the decisions include generating the words of the sentence (i.e. it is generative). The probability of a sequence $P(d_1,..., d_m)$ is then decomposed into the multiplication of the probabilities of each parser decision conditioned on its history of previous decisions $\Pi_i P(d_i|d_1,..., d_{i-1})$.

4.2 Deriving the Kernel

The complete set of neural network weights isn't used to define the kernel, but instead reparameterization is applied to define a third level of parameterization which only includes the network's output layer weights. As suggested in (Henderson and Titov, 2005) use of the complete set of weights doesn't lead to any improvement of the resulting reranker and makes the reranker training more computationally expensive.

Furthermore, to assess the contribution of vocabulary and syntactic structure differences (see section 3.3), we divide the set of the parameters into vocabulary parameters and structural parameters. We consider the parameters used in the estimation of the probability of the next word given the history representation as vocabulary parameters, and the parameters used in the estimation of structural decision probabilities as structural parameters. We define the kernel with structural features as using only structural parameters, and the kernel with vocabulary features as using only vocabulary parameters.

5 Experimental Results

We used the Penn Treebank WSJ corpus and the Brown corpus to evaluate our approach. We used the standard division of the WSJ corpus into training, validation, and testing sets. In the Brown corpus we ran separate experiments for sections F (informative prose: popular lore), K (imaginative prose: general fiction), N (imaginative prose: adventure and western fiction), and P (imaginative prose: romance and love story). These sections were selected because they are sufficiently large, and because they appeared to be maximally different from each other and from WSJ text. In each Brown corpus section, we selected every third sentence for testing. From the remaining sentences, we used 1 sentence out of 20 for the validation set, and the remainder for training. The resulting datasets sizes are presented in table 1.

For the large margin classifier, we used the SVM-Struct (Tsochantaridis et al., 2004) implementation of SVM, which rescales the margin with F_1 measure of bracketed constituents (see (Tsochantaridis et al., 2004) for details). Linear slack penalty was employed.[2]

5.1 Experiments on Transferring across Domains

To evaluate the pure porting scenario (transferring), described in section 3.1, we trained the SSN parsing model on the WSJ corpus. For each tag, there is an unknown-word vocabulary item which is used for all those words not sufficiently frequent with that tag to be included individually in the vocabulary. In the vocabulary of the parser, we included the unknown-word items and the words which occurred in the training set at least 20 times. This led to the vocabulary of 4,215 tag-word pairs.

	testing	training	validation
WSJ	2,416 (54,268)	39,832 (910,196)	1,346 (31,507)
Brown F	1,054 (23,722)	2,005 (44,928)	105 (2,300)
Brown K	1,293 (21,215)	2,459 (39,823)	129 (1,971)
Brown N	1,471 (22,142)	2,797 (42,071)	137 (2,025)
Brown P	1,314 (21,763)	2,503 (41,112)	125 (1,943)

Table 1: Number of sentences (words) for each dataset.

We derived the kernel from the trained model for each target section (F, K, N, P) using reparameterization discussed in section 3.1: we included in the vocabulary all the words which occurred at least twice in the training set of the corresponding section. This approach led to a smaller vocabulary than that of the initial parser but specifically tied to the target domain (3,613, 2,789, 2,820 and 2,553 tag-word pairs for sections F, K, N and P respectively). There is no sense in including the words from the WSJ which do not appear in the Brown section training set because the classifier won't be able to learn the corresponding components of its decision vector. The results for the original probabilistic model (SSN-WSJ) and for the kernel method (TOP-Transfer) on the testing set of each section are presented in table 2.[3]

To evaluate the relative contribution of our porting technique versus the use of the TOP kernel alone, we also used this TOP kernel to train an SVM on the WSJ corpus. We trained the SVM on data from the development set and section 0, so that the size of this dataset (3,267 sentences) was about the same as for each Brown section.[4] This gave us a "TOP-WSJ"

[2] Training of the SVM takes about 3 hours on a standard desktop PC. Running the SVM is very fast, once the probabilistic model has finished computing the probabilities needed to select the candidate parses.

[3] All our results are computed with the evalb program following the standard criteria in (Collins, 1999).

[4] We think that using an equivalently sized dataset provides a fair test of the contribution of the TOP kernel alone. It would also not be computationally tractable to train an SVM on the full WSJ dataset without using different training techniques, which would then compromise the comparison.

model, which we tested on each of the four Brown sections. In each case, the TOP-WSJ model did worse than the original SSN-WSJ model, as shown in table 2. This makes it clear that we are getting no improvement from simply using a TOP kernel alone or simply using more data, and all our improvement is from the proposed porting method.

5.2 Experiments on Focusing on a Subdomain

To perform the experiments on the approach suggested in section 3.2 (focusing), we trained the SSN parser on the WSJ training set joined with the training set of the corresponding section. We included in the vocabulary only words which appeared in the joint training set at least 20 times. Resulting vocabularies comprised 4,386, 4,365, 4,367 and 4,348 for sections F, K, N and P, respectively.[5] Experiments were done in the same way as for the parser transferring approach, but reparameterization was not performed. Standard measures of accuracy for the original probabilistic model (SSN-WSJ+Br) and the kernel method (TOP-Focus) are also shown in table 2.

For the sake of comparison, we also trained the SSN parser on only training data from one of the Brown corpus sections (section P), producing a "SSN-Brown" model. This model achieved an F_1 measure of only 81.0% for the P section testing set, which is worse than all the other models and is 3% lower than our best results on this testing set (TOP-Focus). This result underlines the need to port parsers from domains in which there are large annotated datasets.

5.3 Experiments Comparing Vocabulary to Structure

We conducted the same set of experiments with the kernel with vocabulary features (TOP-Voc-Transfer and TOP-Voc-Focus) and with the kernel with the structural features (TOP-Str-Transfer and TOP-Str-Focus). Average results for classifiers with these kernels, as well as for the original kernel and the baseline, are presented in table 3.

	section	LR	LP	$F_{\beta=1}$
TOP-WSJ	F	83.9	84.9	84.4
SSN-WSJ	F	84.4	85.2	84.8
TOP-Transfer	F	84.5	85.6	85.0
SSN-WSJ+Br	F	84.2	85.2	84.7
TOP-Focus	F	84.6	86.0	85.3
TOP-WSJ	K	81.8	82.3	82.1
SSN-WSJ	K	82.2	82.6	82.4
TOP-Transfer	K	82.4	83.5	83.0
SSN-WSJ+Br	K	83.1	84.2	83.6
TOP-Focus	K	83.6	85.0	84.3
TOP-WSJ	N	83.3	84.5	83.9
SSN-WSJ	N	83.5	84.6	84.1
TOP-Transfer	N	84.3	85.7	85.0
SSN-WSJ+Br	N	85.0	86.5	85.7
TOP-Focus	N	85.0	86.7	85.8
TOP-WSJ	P	81.3	82.1	81.7
SSN-WSJ	P	82.3	83.0	82.6
TOP-Transfer	P	82.7	83.8	83.2
SSN-WSJ+Br	P	83.1	84.3	83.7
TOP-Focus	P	83.3	84.8	84.0

Table 2: Percentage labeled constituent recall (LR), precision (LP), and a combination of both ($F_{\beta=1}$) on the individual test sets.

5.4 Discussion of Results

For the experiments which directly test the usefulness of our proposed porting technique (SSN-WSJ versus TOP-Transfer), our technique demonstrated improvement for each of the Brown sections (table 2), and this improvement was significant for three out of four of the sections (K, N, and P).[6] This demonstrates that data-defined kernels are an effective way to port parsers to a new domain.

For the experiments which combine training a new probability model with our porting technique (SSN-WSJ+Br versus TOP-Focus), our technique still demonstrated improvement over training alone. There was improvement for each of the Brown sections, and this improvement was significant for two

[5] We would expect some improvement if we used a smaller threshold on the target domain, but preliminary results suggest that this improvement would be small.

[6] We measured significance in F_1 measure at the 5% level with the randomized significance test of (Yeh, 2000). We think that the reason the improvement on section F was only significant at the 10% level was that the baseline model (SSN-WSJ) was particularly lucky, as indicated by the fact that it did even better than the model trained on the combination of datasets (SSN-WSJ+Br).

	LR	LP	$F_{\beta=1}$
SSN-WSJ	83.1	83.8	83.5
TOP-Transfer	83.5	84.7	84.1
TOP-Voc-Transfer	83.5	84.7	84.1
TOP-Str-Transfer	83.1	84.3	83.7
SSN-WSJ+Br	83.8	85.0	84.4
TOP-Focus	84.1	85.6	84.9
TOP-Voc-Focus	84.1	85.6	84.8
TOP-Str-Focus	83.9	85.4	84.7

Table 3: Average accuracy of the models on chapters F, K, N and P of the Brown corpus.

out of four of the sections (F and K). This demonstrates that, even when the probability model is well suited to the target domain, there is still room for improvement from using data-defined kernels to optimize the parser specifically to the target domain without losing information about the source domain.

One potential criticism of these conclusions is that the improvement could be the result of reranking with the TOP kernel, and have nothing to do with porting. The lack of an improvement in the TOP-WSJ results discussed in section 5.1 clearly shows that this cannot be the explanation. The opposite criticism is that the improvement could be the result of optimizing to the target domain alone. The poor performance of the SSN-Brown model discussed in section 5.2 makes it clear that this also cannot be the explanation. Therefore reranking with data defined kernels must be both effective at preserving information about the source domain and effective at specializing to the target domain.

The experiments which test the hypothesis that differences in vocabulary distributions are more important than difference in syntactic structure distributions confirm this belief. Results for the classifier which uses the kernel with only vocabulary features are better than those for structural features in each of the four sections with both the Transfer and Focus scenarios. In addition, comparing the results of TOP-Transfer with TOP-Voc-Transfer and TOP-Focus with TOP-Voc-Focus, we can see that adding structural features in TOP-Focus and TOP-Transfer leads to virtually no improvement. This suggest that differences in vocabulary distributions are the only issue we need to address, although this result could possibly also be an indication that our method did not sufficiently exploit structural differences.

In this paper we concentrate on the situation where a parser is needed for a restricted target domain, for which only a small amount of data is available. We believe that this is the task which is of greatest practical interest. For this reason we do not run experiments on the task considered in (Gildea, 2001) and (Roark and Bacchiani, 2003), where they are porting from the restricted domain of the WSJ corpus to the more varied domain of the Brown corpus as a whole. However, to help emphasize the success of our proposed porting method, it is relevant to show that even our baseline models are performing better than this previous work on parser portability. We trained and tested the SSN parser in their "de-focusing" scenario using the same datasets as (Roark and Bacchiani, 2003). When trained only on the WSJ data (analogously to the SSN-WSJ baseline for TOP-Transfer) it achieves results of 82.9%/83.4% LR/LP and 83.2% F_1, and when trained on data from both domains (analogously to the SSN-WSJ+Br baselines for TOP-Focus) it achieves results of 86.3%/87.6% LR/LP and 87.0% F_1. These results represent a 2.2% and 1.3% increase in F_1 over the best previous results, respectively (see the discussion of (Roark and Bacchiani, 2003) below).

6 Related Work

Most research in the field of parsing has focused on the Wall Street Journal corpus. Several researchers have addressed the portability of these WSJ parsers to other domains, but mostly without addressing the issue of how a parser can be designed specifically for porting to another domain. Unfortunately, no direct empirical comparison is possible between our results and results with other parsers, because there is no standard portability benchmark to date where a small amount of data from a target domain is used.

(Ratnaparkhi, 1999) performed portability experiments with a Maximum Entropy parser and demonstrated that the parser trained on WSJ achieves far worse results on the Brown corpus sections. Adding a small amount of data from the target domain improves the results, but accuracy is still much lower than the results on the WSJ. They reported results when their parser was trained on the WSJ training set plus a portion of 2,000 sentences from a Brown corpus section. They achieved 80.9%/80.3% recall/precision for section K, and 80.6%/81.3% for section N.[7] Our analogous method (TOP-Focus) achieved much better accuracy (3.7% and 4.9% better F_1, respectively).

In addition to portability experiments with the parsing model of (Collins, 1997), (Gildea, 2001) provided a comprehensive analysis of parser portability. On the basis of this analysis, a technique for parameter pruning was proposed leading to a significant reduction in the model size without a large decrease of accuracy. Gildea (2001) only reports results on sentences of 40 or less words on all the Brown corpus sections combined, for which he reports 80.3%/81.0% recall/precision when training only on data from the WSJ corpus, and 83.9%/84.8% when training on data from the WSJ corpus and all sections of the Brown corpus.

(Roark and Bacchiani, 2003) performed experiments on supervised and unsupervised PCFG adaptation to the target domain. They propose to use the statistics from a source domain to define priors over weights. However, in their experiments they used only trivial sub-cases of this approach, namely, count merging and model interpolation. They achieved very good improvement over their baseline and over (Gildea, 2001), but the absolute accuracies were still relatively low (as discussed above). They report results with combined Brown data (on sentences of 100 words or less), achieving 81.3%/80.9% when training only on the WSJ corpus and 85.4%/85.9% with their best method using the data from both domains.

7 Conclusions

This paper proposes a novel technique for improving parser portability, applying parse reranking with data-defined kernels. First a probabilistic model of parsing is trained on all the available data, including a large set of data from the source domain. This model is used to define a kernel over parse trees. Then this kernel is used in a large margin classifier trained on a small set of data only from the target domain. This classifier is used to rerank the top parses produced by the probabilistic model on the target domain. Experiments with a neural network statistical parser demonstrate that this approach leads to improved parser accuracy on the target domain, without any significant increase in computational cost.

[7] The sizes of Brown sections reported in (Ratnaparkhi, 1999) do not match the sizes of sections distributed in the Penn Treebank 3.0 package, so we couldn't replicate their split. We suspect that a preliminary version of the corpus was used for their experiments.

References

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures and the voted perceptron. In *Proc. ACL 2002* , pages 263–270, Philadelphia, PA.

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proc. ACL/EACL 1997* , pages 16–23, Somerset, New Jersey.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proc. EMNLP 2001* , Pittsburgh, PA.

James Henderson and Ivan Titov. 2005. Data-defined kernels for parse reranking derived from probabilistic models. In *Proc. ACL 2005* , Ann Arbor, MI.

James Henderson. 2003. Inducing history representations for broad coverage statistical parsing. In *Proc. NAACL/HLT 2003* , pages 103–110, Edmonton, Canada.

Terry Koo and Michael Collins. 2005. Hidden-variable models for discriminative reranking. In *Proc. EMNLP 2005* , Vancouver, B.C., Canada.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proc. ACL 2005* , Ann Arbor, MI.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–175.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsuperised PCFG adaptation to novel domains. In *Proc. HLT/ACL 2003* , Edmionton, Canada.

Libin Shen and Aravind K. Joshi. 2003. An SVM based voting algorithm with application to parse reranking. In *Proc. 7th Conf. on Computational Natural Language Learning*, pages 9–16, Edmonton, Canada.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proc. 21st Int. Conf. on Machine Learning*, pages 823–830, Banff, Alberta, Canada.

Alexander Yeh. 2000. More accurate tests for the statistical significance of the result differences. In *Proc. 17th Int. Conf. on Computational Linguistics*, pages 947–953, Saarbruken, Germany.