In Proc. 2006 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2006).
Funded by Swiss NSF grant 200020-109685 and EU FP6 project TALK.

Loss Minimization in Parse Reranking

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

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

Abstract

We propose a general method for reranker construction which targets choosing the candidate with the least expected loss, rather than the most probable candidate. Different approaches to expected loss approximation are considered, including estimating from the probabilistic model used to generate the candidates, estimating from a discriminative model trained to rerank the candidates, and learning to approximate the expected loss. The proposed methods are applied to the parse reranking task, with various baseline models, achieving significant improvement both over the probabilistic models and the discriminative rerankers. When a neural network parser is used as the probabilistic model and the Voted Perceptron algorithm with data-defined kernels as the learning algorithm, the loss minimization model achieves 90.0% labeled constituents F_1 score on the standard WSJ parsing task.

1 Introduction

The reranking approach is widely used in parsing (Collins and Koo, 2005; Koo and Collins, 2005; Henderson and Titov, 2005; Shen and Joshi, 2003) as well as in other structured classification problems. For structured classification tasks, where labels are complex and have an internal structure of interdependency, the 0-1 loss considered in classical formulation of classification algorithms is not a natural choice and different loss functions are normally employed. To tackle this problem, several approaches have been proposed to accommodate loss functions in learning algorithms (Tsochantaridis et al., 2004; Taskar et al., 2004; Henderson and Titov, 2005). A very different use of loss functions was considered in the areas of signal processing and machine translation, where direct minimization of expected loss (Minimum Bayes Risk decoding) on word sequences was considered (Kumar and Byrne, 2004; Stolcke et al., 1997). The only attempt to use Minimum Bayes Risk (MBR) decoding in parsing was made in (Goodman, 1996), where a parsing algorithm for constituent recall minimization was constructed. However, their approach is limited to binarized PCFG models and, consequently, is not applicable to state-of-the-art parsing methods (Charniak and Johnson, 2005; Henderson, 2004; Collins, 2000). In this paper we consider several approaches to loss approximation on the basis of a candidate list provided by a baseline probabilistic model.

The intuitive motivation for expected loss minimization can be seen from the following example. Consider the situation where there are a group of several very similar candidates and one very different candidate whose probability is just slightly larger than the probability of any individual candidate in the group, but much smaller than their total probability. A method which chooses the maximum probability candidate will choose this outlier candidate, which is correct if you are only interested in getting the label exactly correct (i.e. 0-1 loss), and you think the estimates are accurate. But if you are interested in a loss function where the loss is small when you choose a candidate which is similar to the correct candidate, then it is better to choose one of the candidates in the group. With this choice the loss will only be large if the outlier turns out to be correct, while if the outlier is chosen then the loss will be large if any of the group are correct. In other words, the expected loss of

choosing a member of the group will be smaller than that for the outlier.

More formally, the Bayes risk of a model $y = h(x)$ is defined as

$$R(h) = E_{x,y}\Delta(y, h(x)), \quad (1)$$

where the expectation is taken over all the possible inputs x and labels y and $\Delta(y, y')$ denotes a loss incurred by assigning x to y' when the correct label is y. We assume that the loss function possesses values within the range from 0 to 1, which is equivalent to the requirement that the loss function is bounded in (Tsochantaridis et al., 2004). It follows that an optimal reranker $h^\star$ is one which chooses the label y that minimizes the expected loss:

$$h^\star(x) = \arg\min_{y' \in G(x)} \sum_y P(y|x)\Delta(y, y'), \quad (2)$$

where $G(x)$ denotes a candidate list provided by a baseline probabilistic model for the input x. In this paper we propose different approaches to loss approximation. We apply them to the parse reranking problem where the baseline probabilistic model is a neural network parser (Henderson, 2003), and to parse reranking of candidates provided by the (Collins, 1999) model. The resulting reranking method achieves very significant improvement in the considered loss function and improvement in most other standard measures of accuracy.

In the following three sections we will discuss three approaches to learning such a classifier. The first two derive a classification criteria for use with a predefined probability model (the first generative, the second discriminative). The third defines a kernel for use with a classification method for minimizing loss. All use previously proposed learning algorithms and optimization criteria.

2 Loss Approximation with a Probabilistic Model

In this section we discuss approximating the expected loss using probability estimates given by a baseline probabilistic model. Use of probability estimates is not a serious limitation of this approach because in practice candidates are normally provided by some probabilistic model and its probability estimates are used as additional features in the reranker (Collins and Koo, 2005; Shen and Joshi, 2003; Henderson and Titov, 2005).

In order to estimate the expected loss on the basis of a candidate list, we make the assumption that the total probability of the labels not in the candidate list is sufficiently small that the difference $\delta(x, y')$ of expected loss between the labels in the candidate list and the labels not in the candidate list does not have an impact on the loss defined in (1):

$$\delta(x, y') = \frac{\sum_{y \notin G(x)} P(y|x)\Delta(y, y')}{\sum_{y \notin G(x)} P(y|x)} - \frac{\sum_{y \in G(x)} P(y|x)\Delta(y, y')}{\sum_{y \in G(x)} P(y|x)} \quad (3)$$

This gives us the following approximation to the expected loss for the label:

$$l(x, y') = \frac{\sum_{y \in G(x)} P(y|x)\Delta(y, y')}{\sum_{y \in G(x)} P(y|x)}. \quad (4)$$

For the reranking case, often the probabilistic model only estimates the joint probability $P(x, y)$. However, neither this difference nor the denominator in (4) affects the classification. Thus, replacing the true probabilities with their estimates, we can define the classifier

$$\hat{h}(x) = \arg\min_{y' \in G(x)} \sum_{y \in G(x)} P(x, y|\hat{\theta})\Delta(y, y'), \quad (5)$$

where $\hat{\theta}$ denotes the parameters of the probabilistic model learned from the training data. This approach for expected loss approximation was considered in the context of word error rate minimization in speech recognition, see for example (Stolcke et al., 1997).

3 Estimating Expected Loss with Discriminative Classifiers

In this section we propose a method to improve on the loss approximation used in (5) by constructing the probability estimates using a trained discriminative classifier. Special emphasis is placed on linear classifiers with data-defined kernels for reranking (Henderson and Titov, 2005), because they do not require any additional domain knowledge not already encoded in the probabilistic model, and they have demonstrated significant improvement over the baseline probabilistic model for the parse reranking task. This kernel construction can be motivated by the existence of a function which maps a linear function in the feature space of the kernel to probability estimates which are superior to the estimates of the original probabilistic model.

3.1 Estimation with Fisher Kernels

The Fisher kernel for structured classification is a trivial generalization of one of the best known data-defined kernels for binary classification (Jaakkola and Haussler, 1998). The Fisher score of an example input-label pair (x, y) is a vector of partial derivatives of the log-likelihood of the example with respect to the model parameters[1]:

$$\phi^{FK}_{\hat{\theta}}(x,y) = \quad (6)$$
$$(\log P(x,y|\hat{\theta}), \frac{\partial \log P(x,y|\hat{\theta})}{\partial \theta_1}, ..., \frac{\partial \log P(x,y|\hat{\theta})}{\partial \theta_l}).$$

This kernel defines a feature space which is appropriate for estimating the discriminative probability in the candidate list in the form of a normalized exponential

$$\frac{P(x,y)}{\sum_{y' \in G(x)} P(x,y')} \approx \quad (7)$$
$$\frac{\exp(w^{\star T} \phi^{FK}_{\hat{\theta}}(x,y))}{\sum_{y' \in G(x)} \exp(w^{\star T} \phi^{FK}_{\hat{\theta}}(x,y'))}$$

for some choice of the decision vector $w = w^\star$ with the first component equal to one.

It follows that it is natural to use an estimator of the discriminative probability $P(y|x)$ in exponential form and, therefore, the appropriate form of the loss minimizing classifier is the following:

$$\hat{h}_{FK}(x) = \quad (8)$$
$$\arg\min_{y' \in G(x)} \sum_{y \in G(x)} exp(A\hat{w}^T \phi^{FK}_{\hat{\theta}}(x,y')) \Delta(y,y'),$$

where $\hat{w}$ is learned during classifier training and the scalar parameter A can be tuned on the development set. From the construction of the Fisher kernel, it follows that the optimal value A is expected to be close to inverse of the first component of $\hat{w}$, $1/\hat{w}_1$.

If an SVM is used to learn the classifier, then the form (7) is the same as that proposed by (Platt, 1999), where it is proposed to use the logistic sigmoid of the SVM output as the probability estimator for binary classification problems.

[1]The first component $log P(x,y|\hat{\theta})$ is not in the strict sense part of the Fisher score, but usually added to kernel features in practice (Henderson and Titov, 2005).

3.2 Estimation with TOP Kernels for Reranking

The TOP Reranking kernel was defined in (Henderson and Titov, 2005), as a generalization of the TOP kernel (Tsuda et al., 2002) proposed for binary classification tasks. The feature extractor for the TOP reranking kernel is given by:

$$\phi^{TK}_{\hat{\theta}}(x,y) = \quad (9)$$
$$(v(x,y,\hat{\theta}), \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_1}, ..., \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_l}),$$

where

$$v(x,y,\hat{\theta}) = \log P(x,y|\hat{\theta}) - \log \sum_{y' \in G(x) - \{y\}} P(x,y'|\hat{\theta}).$$

The TOP reranking kernel has been demonstrated to perform better than the Fisher kernel for the parse reranking task (Henderson and Titov, 2005). The construction of this kernel is motivated by the minimization of the classification error of a linear classifier $w^T \phi_{\hat{\theta}}(x,y)$. This linear classifier has been shown to converge, assuming estimation of the discriminative probability in the candidate list can be in the form of the logistic sigmoid (Titov and Henderson, 2005):

$$\frac{P(x,y)}{\sum_{y' \in G(x)} P(x,y')} \approx \quad (10)$$
$$\frac{1}{1 + \exp(-w^{\star T} \phi^{TK}_{\hat{\theta}}(x,y))}$$

for some choice of the decision vector $w = w^\star$ with the first component equal to one. From this fact, the form of the loss minimizing classifier follows:

$$\hat{h}_{TK}(x) = \quad (11)$$
$$\arg\min_{y' \in G(x)} \sum_{y \in G(x)} g(A\hat{w}^T \phi^{TK}_{\hat{\theta}}(x,y')) \Delta(y,y'),$$

where g is the logistic sigmoid and the scalar parameter A should be selected on the development set. As for the Fisher kernel, the optimal value of A should be close to $1/\hat{w}_1$.

3.3 Estimates from Arbitrary Classifiers

Although in this paper we focus on approaches which do not require additional domain knowledge, the output of most classifiers can be used to estimate the discriminative probability in equation (7). As mentioned above, the form of (7)

is appropriate for the SVM learning task with arbitrary kernels, as follows from (Platt, 1999). Also, for models which combine classifiers using votes (e.g. the Voted Perceptron), the number of votes cast for each candidate can be used to define this discriminative probability. The discriminative probability of a candidate is simply the number of votes cast for that candidate normalized across candidates. Intuitively, we can think of this method as treating the votes as a sample from the discriminative distribution.

4 Expected Loss Learning

In this section, another approach to loss approximation is proposed. We consider learning a linear classifier to choose the least loss candidate, and propose two constructions of data-defined loss kernels which define different feature spaces for the classification. In addition to the kernel, this approach differs from the previous one in that the classifier is assumed to be linear, rather than the nonlinear functions in equations (8) and (11).

4.1 Loss Kernel

The Loss Kernel feature extractor is composed of the logarithm of the loss estimated by the probabilistic model and its first derivatives with respect to each model parameter:

$$\phi^{LK}_{\hat{\theta}}(x,y) = \quad (12)$$
$$(v(x,y,\hat{\theta}), \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_1}, ..., \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_l}),$$

where

$$v(x,y,\hat{\theta}) = \log(\sum_{y' \in G(x)} P(y',x|\hat{\theta})\Delta(y',y)).$$

The motivation for this kernel is very similar to that for the Fisher kernel for structured classification. The feature space of the kernel guarantees convergence of an estimator for the expected loss if the estimator is in normalized exponential form. The standard Fisher kernel for structured classification is a special case of this Loss Kernel when $\Delta(y,y')$ is 0-1 loss.

4.2 Loss Logit Kernel

As the Loss kernel was a generalization of the Fisher kernel to arbitrary loss function, so the Loss Logit Kernel is a generalization of the TOP kernel for reranking. The construction of the Loss Logit Kernel, like the TOP kernel for reranking, can be motivated by the minimization of the classification error of a linear classifier $w^T\phi^{LLK}_{\hat{\theta}}(x,y)$, where $\phi^{LLK}_{\hat{\theta}}(x,y)$ is the feature extractor of the kernel given by:

$$\phi^{LLK}_{\hat{\theta}}(x,y) = \quad (13)$$
$$(v(x,y,\hat{\theta}), \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_1}, ..., \frac{\partial v(x,y,\hat{\theta})}{\partial \theta_l}),$$

where

$$v(x,y,\hat{\theta}) = \log(\sum_{y' \in G(x)} P(y'|x,\hat{\theta})(1-\Delta(y',y))) - \log(\sum_{y' \in G(x)} P(y'|x,\hat{\theta})\Delta(y',y)).$$

5 Experimental Evaluation

To perform empirical evaluations of the proposed methods, we considered the task of parsing the Penn Treebank Wall Street Journal corpus (Marcus et al., 1993). First, we perform experiments with SVM Struct (Tsochantaridis et al., 2004) as the learner. Since SVM Struct already uses the loss function during training to rescale the margin or slack variables, this learner allows us to test the hypothesis that loss functions are useful in parsing not only to define the optimization criteria but also to define the classifier and to define the feature space. However, SVM Struct training for large scale parsing experiments is computationally expensive[2], so here we use only a small portion of the available training data to perform evaluations of the different approaches. In the other two sets of experiments, described below, we test our best model on the standard Wall Street Journal parsing benchmark (Collins, 1999) with the Voted Perceptron algorithm as the learner.

5.1 The Probabilistic Models of Parsing

To perform the experiments with data-defined kernels, we need to select a probabilistic model of parsing. Data-defined kernels can be applied to any kind of parameterized probabilistic model.

For our first set of experiments, we choose to use a publicly available neural network based probabilistic model of parsing (Henderson, 2003).

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

This parsing model is 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). This gives us a baseline which is hard to beat, and allows us to compare results of our new approaches with the results of the original data-defined kernels for reranking.

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 data-defined kernels in terms of the second level of parameterization (the network weights).

For the last set of experiments, we used the probabilistic model described in (Collins, 1999) (model 2), and the Tree Kernel (Collins and Duffy, 2002). However, in these experiments we only used the estimates from the discriminative classifier, so the details of the probabilistic model are not relevant.

5.2 Experiments with SVM Struct

Both the neural network probabilistic model and the kernel based classifiers were trained on section 0 (1,921 sentences, 40,930 words). Section 24 (1,346 sentences, 29,125 words) was used as the validation set during the neural network learning and for choosing parameters of the models. Section 23 (2,416 sentences, 54,268 words) was used for the final testing of the models.

We used a publicly available tagger (Ratnaparkhi, 1996) to provide the part-of-speech tags for each word in the sentence. For each tag, there is an unknown-word vocabulary item which is used for all those words which are not sufficiently frequent with that tag to be included individually in the vocabulary. For these experiments, we only included a specific tag-word pair in the vocabulary if it occurred at least 20 time in the training set, which (with tag-unknown-word pairs) led to the very small vocabulary of 271 tag-word pairs. The same model was used both for choosing the list of candidate parses and for the probabilistic model used for loss estimation and kernel feature extraction. For training and testing of the kernel models, we provided a candidate list consisting of the top 20 parses found by the probabilistic model. For the testing set, selecting the candidate with an oracle results in an F_1 score of 89.1%.

	R	P	F_1	CM
SSN	80.9	81.7	81.3	18.3
TRK	81.1	82.4	81.7	18.2
SSN-Estim	81.4	82.3	81.8	18.3
LLK-Learn	81.2	82.4	81.8	17.6
LK-Learn	81.5	82.2	81.8	17.8
FK-Estim	81.4	82.6	82.0	18.3
TRK-Estim	81.5	82.8	82.1	18.6

Table 1: Percentage labeled constituent recall (R), precision (P), combination of both (F_1) and percentage complete match (CM) on the testing set.

We used the SVM Struct software package (Tsochantaridis et al., 2004) to train the SVM for all the approaches based on discriminative classifier learning, with slack rescaling and linear slack penalty. The loss function is defined as $\Delta(y, y') = 1 - F_1(y, y')$, where F_1 denotes F_1 measure on bracketed constituents. This loss was used both for rescaling the slacks in the SVM and for defining our classification models and kernels.

We performed initial testing of the models on the validation set and preselected the best model for each of the approaches before testing it on the final testing set. Standard measures of parsing accuracy, plus complete match accuracy, are shown in table 1.[3] As the baselines, the table includes the results of the standard TOP reranking kernel (TRK) (Henderson and Titov, 2005) and the baseline probabilistic model (SSN) (Henderson, 2003). SSN-Estim is the model using loss estimation on the basic probabilistic model, as explained in section 2. LLK-Learn and LK-Learn are the models which define the kernel based on loss, using the Loss Logit Kernel (equation (13)) and the Loss Kernel (equation (12)), respectively. FK-Estim and TRK-Estim are the models which esti-

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

mate the loss with data-defined kernels, using the Fisher Kernel (equation (8)) and the TOP Reranking kernel (equation (11)), respectively.

All our proposed models show better F_1 accuracy than the baseline probabilistic model SSN, and all these differences are statistically significant.[4] The difference in F_1 between TRK-Estim and FK-Estim is not statistically significant, but otherwise TRK-Estim demonstrates a statistically significant improvement over all other models. It should also be noted that exact match measures for TRK-Estim and SSN-Estim are not negatively affected, even though the F_1 loss function was optimized. It is important to point out that SSN-Estim, which improves significantly over SSN, does not require the learning of a discriminative classifier, and differs from the SSN only by use of the different classification model (equation (5)), which means that it is extremely easy to apply in practice.

One surprising aspect of these results is the failure of LLK-Learn and LK-Learn to achieve improvement over SSN-Estim. This might be explained by the difficulty of learning a linear approximation to (4). Under this explanation, the performance of LLK-Learn and LK-Learn could be explained by the fact that the first component of their kernels is a monotonic function of the SSN-Estim estimation. To test this hypothesis, we did an additional experiment where we removed the first component of Loss Logit Kernel (13) from the feature vector and performed learning. Surprisingly, the model achieved virtually the same results, rather than the predicted worse performance. This result might indicate that the LLK-Learn model still can be useful for different problems where discriminative learning gives more advantage over generative approaches.

These experimental results demonstrate that the loss approximation reranking approaches proposed in this paper demonstrate significant improvement over the baseline models, achieving about the same relative error reduction as previously achieved with data-defined kernels (Henderson and Titov, 2005). This improvement is despite the fact that the loss function is already used in the definition of the training criteria for all the models except SSN. It is also interesting to note that the best result on the validation set for estimation of the loss with data-defined kernels (12) and (13) was achieved when the parameter A is close to the inverse of the first component of the learned decision vector, which confirms the motivation for these kernels.

[4]We measured significance of all the experiments in this paper with the randomized significance test (Yeh, 2000).

5.3 Experiments with Voted Perceptron and Data-Defined Kernels

The above experiments with the SVM Struct demonstrate empirically the viability of our approaches. The aim of experiments on the entire WSJ is to test whether our approaches still achieve significant improvement when more accurate generative models are used, and also to show that they generalize well to learning methods different from SVMs. We perform experiments on the standard WSJ parsing data using the standard split into training, validation and testing sets. We replicate completely the setup of experiments in (Henderson and Titov, 2005). For a detailed description of the experiment setup, we refer the reader to (Henderson and Titov, 2005). We only note here that the candidate list has 20 candidates, and, for the testing set, selecting the candidate with an oracle results in an F_1 score of 95.4%.

We selected the TRK-Estim approach for these experiments because it demonstrated the best results in the previous set of experiments (5.2). We trained the Voted Perceptron (VP) modification described in (Henderson and Titov, 2005) with the TOP Reranking kernel. VP is not a linear classifier, so we were not able to use a classifier in the form (11). Instead the normalized counts of votes given to the candidate parses were used as probability estimates, as discussed in section 3.3.

The resulting accuracies of this model are presented in table 2, together with results of the TOP Reranking kernel VP (Henderson and Titov, 2005) and the SSN probabilistic model (Henderson, 2003). Model TRK-Estim achieves significantly better results than the previously proposed models, which were evaluated in the same experimental setup. Again, the relative error reduction is about the same as that of TRK. The resulting system, consisting of the generative model and the reranker, achieves results at the state-of-the-art level. We believe that this method can be applied to most parsing models to achieve a significant improvement.

	R	P	F_1
Henderson, 2003	88.8	89.5	89.1
Henderson&Titov, 2005	89.1	90.1	89.6
TRK-Estim	89.5	90.5	90.0

Table 2: Percentage labeled constituent recall (R), precision (P), combination of both (F_1) on the testing set.

5.4 Experiments with Voted Perceptron and Tree Kernel

In this series of experiments we validate the statement in section 3.3, where we suggested that loss approximation from a discriminative classifier is not limited only to models with data-defined kernels. We apply the same method as used in the TRK-Estim model above to the Tree Kernel (Collins and Duffy, 2002), which we call the TK-Estim model.

We replicated the parse reranking experimental setup used for the evaluation of the Tree Kernel in (Collins and Duffy, 2002), where the candidate list was provided by the generative probabilistic model (Collins, 1999) (model 2). A list of on average 29 candidates was used, with an oracle F_1 score on the testing set of 95.0%. We trained VP using the same parameters for the Tree Kernel and probability feature weighting as described in (Collins and Duffy, 2002). A publicly available efficient implementation of the Tree Kernel was utilized to speed up computations (Moschitti, 2004). As in the previous section, votes of the perceptron were used to define the probability estimate used in the classifier.

The results for the MBR decoding method (TK-Estim), defined in section 3.3, along with the standard Tree Kernel VP results (Collins and Duffy, 2002) (TK) and the probabilistic baseline (Collins, 1999) (CO99) are presented in table 3. The proposed model improves in F_1 score over the standard VP results. Differences between all the models are statistically significant. The error reduction of TK-Estim is again about the same as the error reduction of TK. This improvement is achieved without adding any additional linguistic features. It is important to note that the model improves in other accuracy measures as well. We would expect even better results with MBR-decoding if larger n-best lists are used. The n-best parsing algorithm (Huang and Chiang, 2005) can be used to efficiently produce candidate lists as large as 10^6 parse trees with the model of (Collins, 1999).

	R	P	F_{1*}	CB	0C	2C
CO99	88.1	88.3	88.2	1.06	64.0	85.1
TK	88.6	88.9	88.7	0.99	66.5	86.3
TK-Estim	89.0	89.5	89.2	0.91	66.6	87.4

* F_1 for previous models may have rounding errors.

Table 3: Result on the testing set. Percentage labeled constituent recall (R), precision (P), combination of both (F_1), an average number of crossing brackets per sentence (CB), percentage of sentences with 0 and $\leq$ 2 crossing brackets (0C and 2C, respectively).

6 Conclusions

This paper considers methods for the estimation of expected loss for parse reranking tasks. The proposed methods include estimation of the loss from a probabilistic model, estimation from a discriminative classifier, and learning of the loss using a specialized kernel. An empirical comparison of these approaches on parse reranking tasks is presented. Special emphasis is given to data-defined kernels for reranking, as they do not require the introduction of any additional domain knowledge not already encoded in the probabilistic model. The best approach, estimation of the loss on the basis of a discriminative classifier, achieves very significant improvements over the baseline generative probabilistic models and the discriminative classifier itself. Though the largest improvement is demonstrated in the measure which corresponds to the considered loss functional, other measures of accuracy are also improved. The proposed method achieves 90.0% F_1 score on the standard Wall Street Journal parsing task when the SSN neural network is used as the probabilistic model and VP with a TOP Reranking kernel as the discriminative classifier.

Acknowledgments

We would like to thank Michael Collins and Terry Koo for providing us their data and useful comments on experimental setup, and Alessandro Moschitti for providing us the source code for his Tree Kernel implementation. We also thank anonymous reviewers for their constructive comments.

References

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. 43rd Meeting of Association for Computational Linguistics*, pages 173–180, Ann Arbor, MI.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures and the voted perceptron. In *Proc. 40th Meeting of Association for Computational Linguistics*, pages 263–270, Philadelphia, PA.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69.

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

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. 17th Int. Conf. on Machine Learning*, pages 175–182, Stanford, CA.

Joshua Goodman. 1996. Parsing algorithms and methods. In *Proc. 34th Meeting of the Association for Computational Linguistics*, pages 177–183, Santa Cruz, CA.

James Henderson and Ivan Titov. 2005. Data-defined kernels for parse reranking derived from probabilistic models. In *Proc. 43rd Meeting of Association for Computational Linguistics*, Ann Arbor, MI.

James Henderson. 2003. Inducing history representations for broad coverage statistical parsing. In *Proc. joint meeting of North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conf.*, pages 103–110, Edmonton, Canada.

James Henderson. 2004. Discriminative training of a neural network statistical parser. In *Proc. 42nd Meeting of Association for Computational Linguistics*, Barcelona, Spain.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proc. 9th Int. Workshop on Parsing Technologies*, Vancouver, Canada.

Tommi S. Jaakkola and David Haussler. 1998. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processes Systems 11*.

Terry Koo and Michael Collins. 2005. Hidden-variable models for discriminative reranking. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics*, Boston, MA.

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.

Alessandro Moschitti. 2004. A study on convolutional kernels for shallow semantic parsing. In *Proc. 42nd Meeting of the Association for Computational Linguistics*, Barcelona, Spain.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparision to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, pages 133–142, Univ. of Pennsylvania, PA.

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

Andreas Stolcke, Yochai Konig, and Mitchel Weintraub. 1997. Explicit word error minimization in n-best list rescoring. In *Proc. of 5th European Conference on Speech Communication and Technology*, pages 163–165, Rhodes, Greece.

Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Christopher Manning. 2004. Max-margin parsing. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Ivan Titov and James Henderson. 2005. Deriving kernels from MLP probability estimators for large categorization problems. In *International Joint Conference on Neural Networks*, Montreal, 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.

K. Tsuda, M. Kawanabe, G. Ratsch, S. Sonnenburg, and K. Muller. 2002. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414.

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