

Discriminative Reranking for Machine Translation

Libin Shen, UPenn

Anoop Sarkar, Simon Fraser U.

Franz Josef Och, USC/ISI

May 4, 2004

Outline

- Discriminative Training
 - in Machine Translation
 - in other NLP Problems
- Discriminative Reranking for MT
- Experiments
- Conclusions and Future Work

Machine Translation

- Source-channel model
 - (Brown et al., 1990, 1993)
- Discriminative training
 - Maximum entropy
 - (Papineni et al., 1997), (Och and Ney, 2002)
 - Maximum BLEU training
 - (Och, 2003), (SMT Team, 2003)

Discriminative Training in NLP

- The need for global features in generative problems
- Difficulty of dynamic programming
- Collins' Perceptron algorithm in EMNLP 02
- Reranking algorithms as approximation
 - Boosting (Collins, 2000)
 - Perceptron (Collins and Duffy, 2002)
 - SVMs (Shen and Joshi, 2003)

Collins EMNLP 02

- Input: Training samples (s_i, t_i)
- For $round = 1 \dots T, i = 1 \dots n$
 1. Calculate $x_i = \arg \max_{x \in \mathbf{GEN}(s_i)} \Phi(s_i, x) \cdot \bar{w}$
 2. If $(x_i \neq t_i)$, $\bar{w} = \bar{w} + \Phi(s_i, t_i) - \Phi(s_i, x_i)$
- Output: Parameter vector: \bar{w}
- To compute x_i is time consuming in some cases
- If x_i is always in a small set of candidates at each iteration, use **reranking** algorithms.

Reranking Algorithms

- Boosting

- $\text{BoostLoss}(w) = \sum_{i,j} e^{-w \cdot (\tilde{\mathbf{x}}_i - \mathbf{x}_{i,j})}$

- Perceptron

- $w = w + (\tilde{\mathbf{x}}_i - \mathbf{x}_{i,j})$

- SVMs

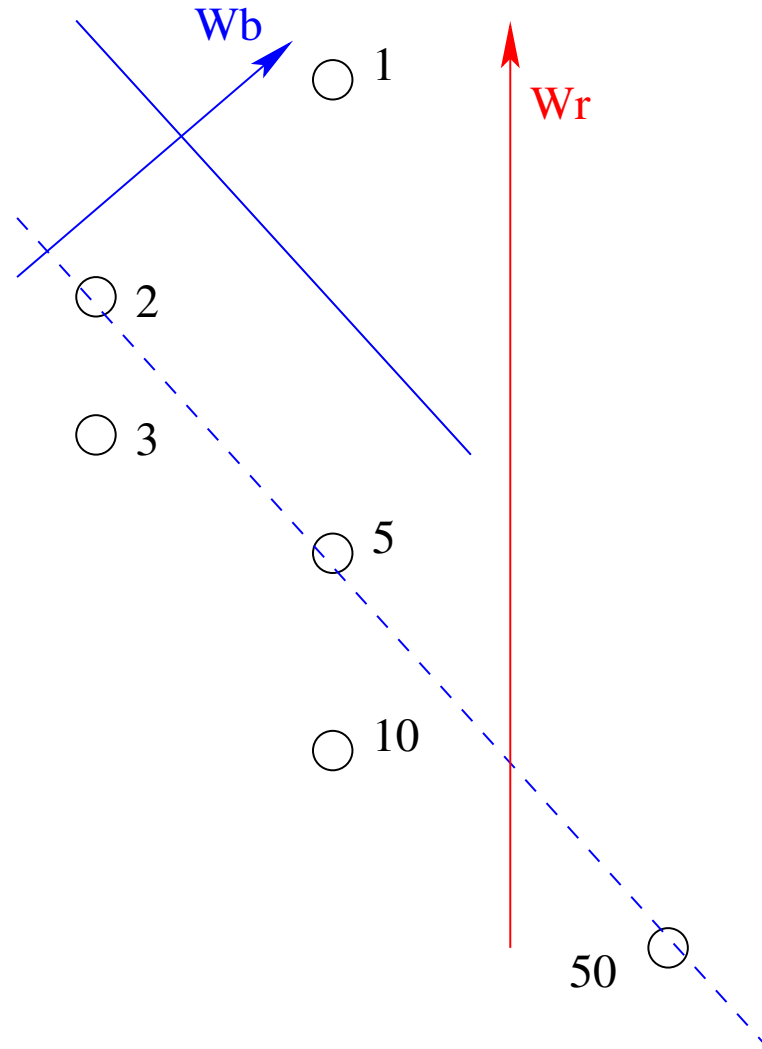
- $\mathbf{u}_{i,j}^+ \equiv \tilde{\mathbf{x}}_i - \mathbf{x}_{i,j}, \quad \mathbf{u}_{i,j}^- \equiv \mathbf{x}_{i,j} - \tilde{\mathbf{x}}_i$

- In these algorithms, we only distinguish the best candidate $\tilde{\mathbf{x}}_i$ from the rest for each sample s_i .

Explain Reranking with Margin

- Large margin classification
 - large margin = lower expected errors (Vapnik, 1998)
- For reranking and Collins' Perceptron
 - margin = distance between #1 and the rest
 - *Is this enough for MT?*
- But, for Machine Translation reranking
 - multiple references, no single best translation.
 - different best translation w.r.t. different score metric.

the Best vs. the Rest



Algorithm 1 : Splitting

- For example, for each source sentence with 1000 translations
 - best 300 as good samples
 - worst 300 as bad samples
 - discard middle 400 for margin
- Using pairwise samples
 - positive: $\mathbf{r}_{\text{good}} - \mathbf{r}_{\text{bad}}$
 - negative: $\mathbf{r}_{\text{bad}} - \mathbf{r}_{\text{good}}$

Algorithm 1: Splitting

- Problem: computational complexity
 - Space complexity: $300*300*2 = 180K$ samples each source sentence, for example.
- Dynamic pairing
 - Decrease space complexity at the expense of time complexity.
- Further improvement to decrease the time complexity
 - Compute $\mathbf{w} \cdot \mathbf{r}_{i,j}$ only once in each iteration, instead of, for example, 600 times.

Algorithm 1: Splitting

repeat

for (sentence $i = 1, \dots, m$) **do**

compute $\mathbf{w}^t \cdot \mathbf{r}_{i,j}$ and $u_j \leftarrow 0$ for all j 's;

for ($j \leq r < n-k \leq l$) **do**

if ($\mathbf{w}^t \cdot \mathbf{r}_{i,j} < \mathbf{w}^t \cdot \mathbf{r}_{i,l} + \tau$) **then**

$u_j \leftarrow u_j + 1; u_l \leftarrow u_l - 1;$

end if

end for

$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \sum_j u_j \mathbf{r}_{i,j}; t \leftarrow t + 1;$

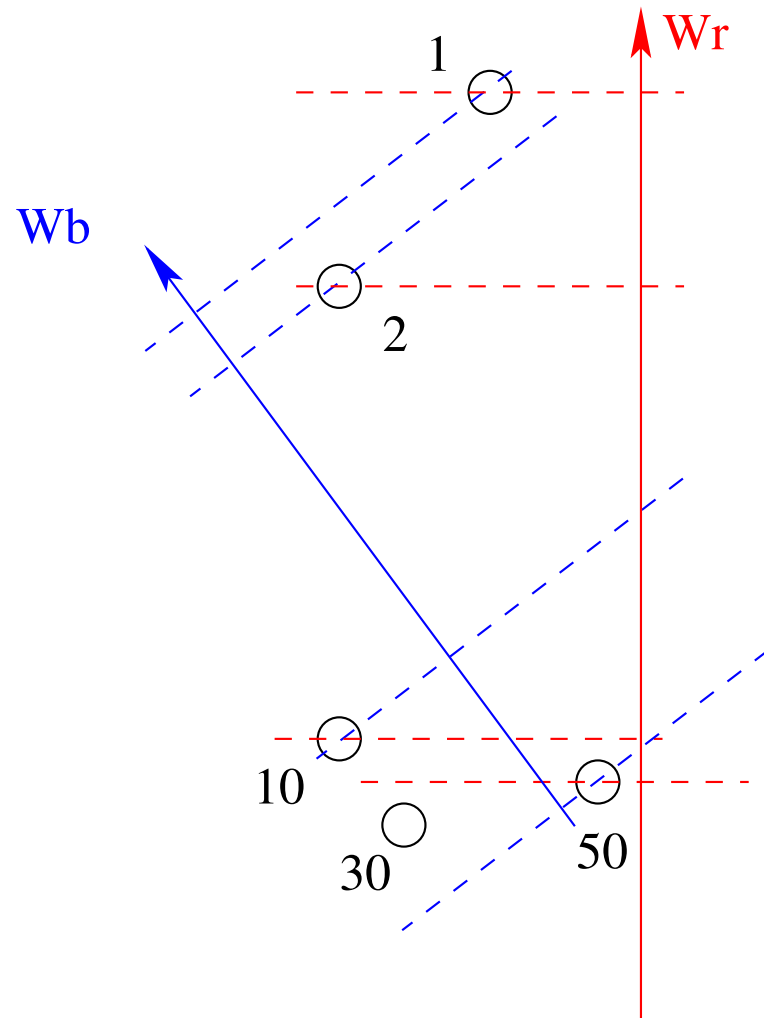
end for

until no updates made in the outer **for** loop

Full Pairwise Samples

- Splitting does not distinguish \mathbf{r}_1 and \mathbf{r}_{200} in training, for example.
 - \mathbf{r}_i be the i^{th} best translation
- Full pairwise is more desirable
 - $\mathbf{w} \cdot \mathbf{r}_1 > \mathbf{w} \cdot \mathbf{r}_{200} > \mathbf{w} \cdot \mathbf{r}_{600}$
 - $\mathbf{w} \cdot \mathbf{r}_i > \mathbf{w} \cdot \mathbf{r}_j$, if $i > j$
- Reranking as Ordinal Regression ?

Ordinal Regression



Uneven Margins

- Binary classification with uneven margin (Li et al., 2002)
 - Much more negative samples
 - Bigger loss with errors on positive samples
 - Larger margin for positive samples
- Distribution of top candidates is more important
 - Top candidates will compete for the first position
 - We are penalized if $\mathbf{w} \cdot \mathbf{r}_{11} > \mathbf{w} \cdot \mathbf{r}_1$
 - but NOT if $\mathbf{w} \cdot \mathbf{r}_{11} > \mathbf{w} \cdot \mathbf{r}_{21}$
- Larger margin for top candidates
 - $\text{margin}(\mathbf{r}_1, \mathbf{r}_{11}) > \text{margin}(\mathbf{r}_{11}, \mathbf{r}_{21})$

Full Pairwise with Uneven Margins

- Neither a classification nor an ordinal regression.
- $\text{margin}(\mathbf{r}_1, \mathbf{r}_{21}) > \text{margin}(\mathbf{r}_1, \mathbf{r}_{11}) > \text{margin}(\mathbf{r}_{11}, \mathbf{r}_{21})$
- Further research on the margin function
- For example $m(\mathbf{r}_i, \mathbf{r}_j) = \frac{1}{i} - \frac{1}{j}$
- Use this margin function in our experiments

Algorithm 2 : Ordinal regression with uneven margins

repeat

for (sentence $i = 1, \dots, m$) **do**

compute $\mathbf{w}^t \cdot \mathbf{r}_{i,j}$ and $u_j \leftarrow 0$ for all j 's;

for ($1 \leq j < l \leq n$) **do**

if ($\mathbf{w}^t \cdot \mathbf{r}_{i,j} < \mathbf{w}^t \cdot \mathbf{r}_{i,l} + m(j,l)\tau$) **then**

$u_j \leftarrow u_j + m(j,l)$; $u_l \leftarrow u_l - m(j,l)$;

end if

end for

$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \sum_j u_j \mathbf{r}_{i,j}$; $t \leftarrow t + 1$;

end for

until no updates made in the outer **for** loop

Experiment Setup

- NIST 2003 Chinese-English large data track evaluation
 - the same as the *Smorgasbord* paper on NAACL 04
- Dataset for our perceptron algorithms
 - Training: 993 Chinese sentences with 1000 best translations each
 - Test: 878 English sentences with 1000 best translations each
 - 1000 best translations generated by a phrase-base MT system (Och, 2003)

Experiment Setup

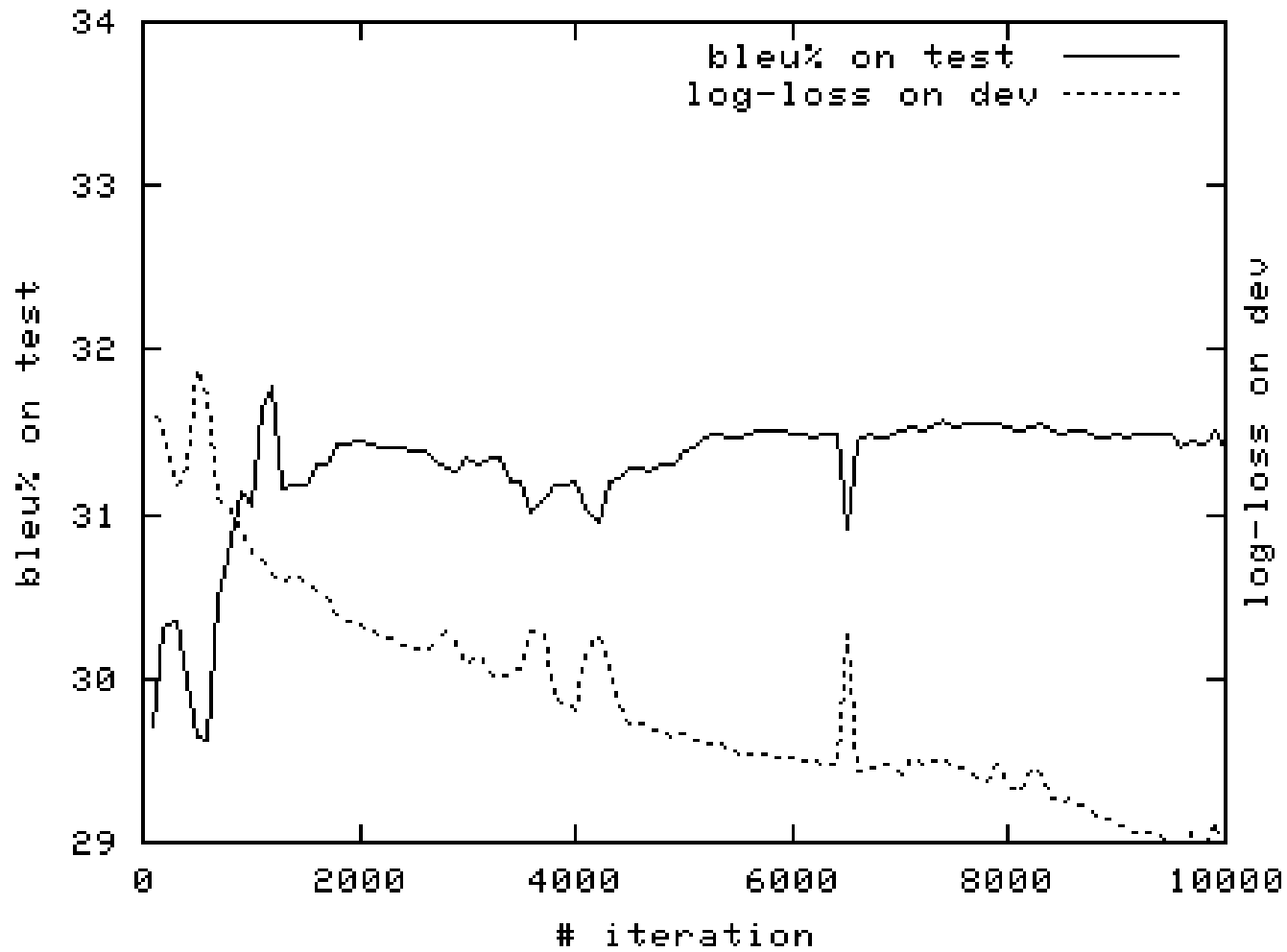
- Using features developed by the SMT team at JHU Summer Workshop 04
 - the same as the *Smorgasbord* paper on NAACL 04
- Four datasets in training and test
 - Baseline set: 11 baseline features in (Och, 2003)
 - Top 4: baseline + the best 4 individual features
 - Top 20: baseline + the best 20 individual features
 - Top 50: baseline + the best 50 individual features

Experiments

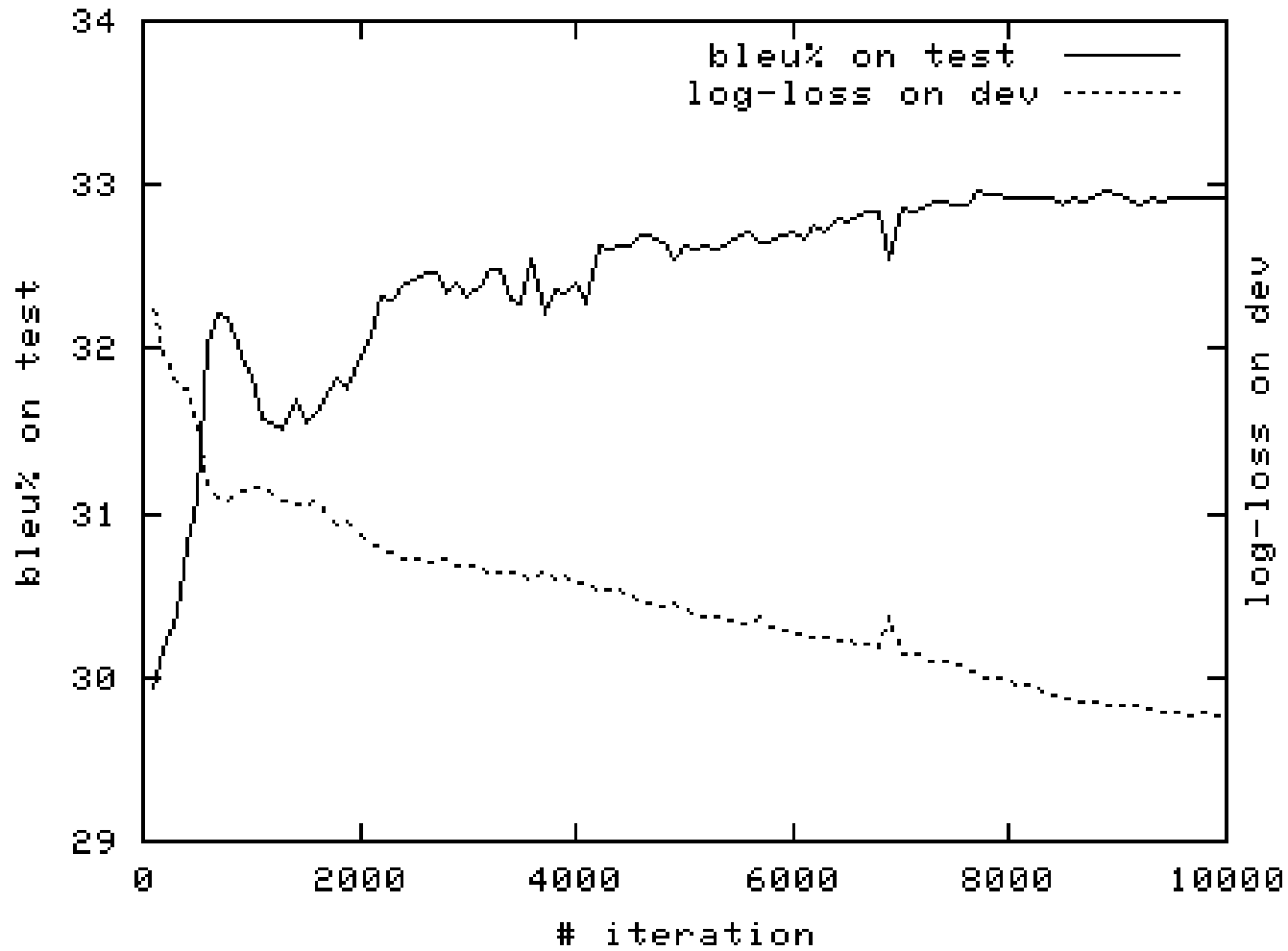
Algorithm	Baseline	Top 4	Top 20
Maximum bleu	31.6	32.6	N/A
Splitting	31.7	32.8	32.6
ORUM	31.4	32.7	32.9

- BLEU% on the test data
- Maximum bleu training achieved BLEU% of **32.9** by combining 12 features, as reported in the *Smorgasbord* paper on NAACL 04.
- Neither Splitting nor ORUM converges on the Top 50.

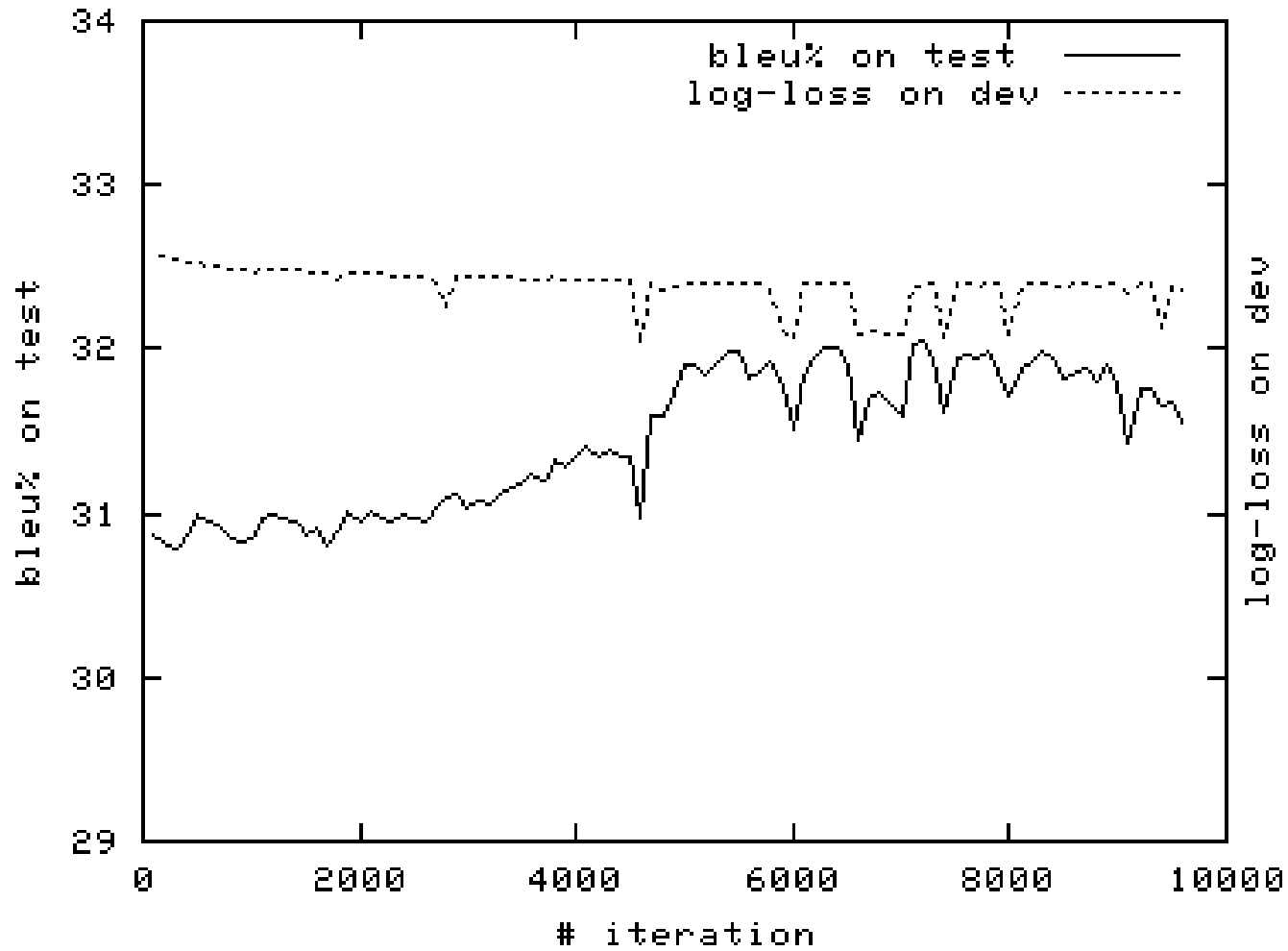
ORUM on Baseline



ORUM on Top 20



ORUM on Top 50



Conclusions

- Two perceptron-like algorithms for MT Reranking
 - dynamic pairing
 - uneven margins, useful in other machine learning algs.
 - theoretically and experimentally verified
- The idea of ORUM provides a way to utilize more information encoded in the training data.
- Experimental results on NIST dataset as good as maximum bleu training

Future Work

- Fix the problem on the top 50 dataset
- Use rich sparse features in the reranking framework
- More research on margin functions
- Apply ORUM to other discriminative learning algorithms with stronger generalization capability.

Thank You

Theoretic Justification

- Stop after finite number of rounds with large margin if the training data is separable.
- Margin based generalization bounds (Vapnik, 1998)
- Pairwise samples are not i.i.d.
 - Using only $n - 1$ pairs for each n best translations for a source sentence (Herbrich et al., 2000)
 - $(\mathbf{r}_1 - \mathbf{r}_2), (\mathbf{r}_2 - \mathbf{r}_3), (\mathbf{r}_3 - \mathbf{r}_4), \dots, (\mathbf{r}_{n-1} - \mathbf{r}_n)$.