Large-scale Discriminative Training

Review

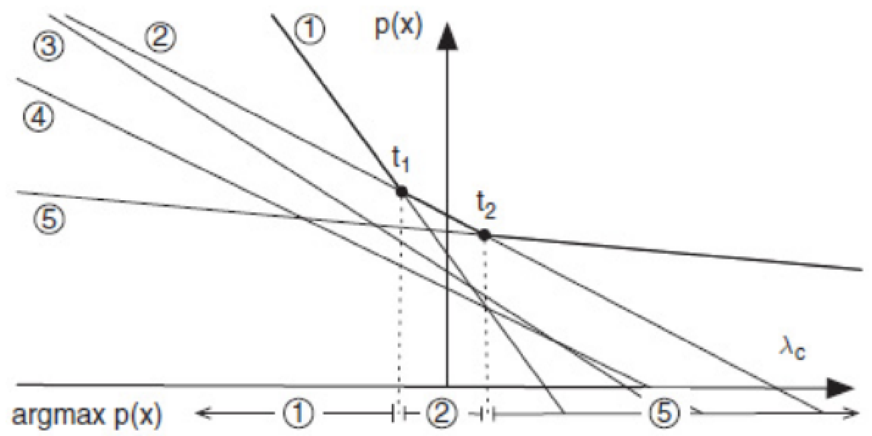
- Weighted linear model

feature vector

$$y = \underset{z \in \mathcal{Y}(x)}{\operatorname{argmax}} \mathbf{W} \cdot \mathbf{h}_z$$

set of hypotheses

- Basic tuning strategy: MERT



Need for More Features

- We can add many more features

$$h(e,f,a)=\begin{cases} 1 & \text{if } f_i = \text{“早上好”}, e_i = \text{“good morning”} \\ 0 & \text{otherwise} \end{cases}$$

$$h(e,f,a)=\begin{cases} 1 & \text{if exists a verb in } e \\ 0 & \text{otherwise} \end{cases}$$

Pros

- Incorporate rich human knowledge
- High-dimension → more linearly separable

Cons

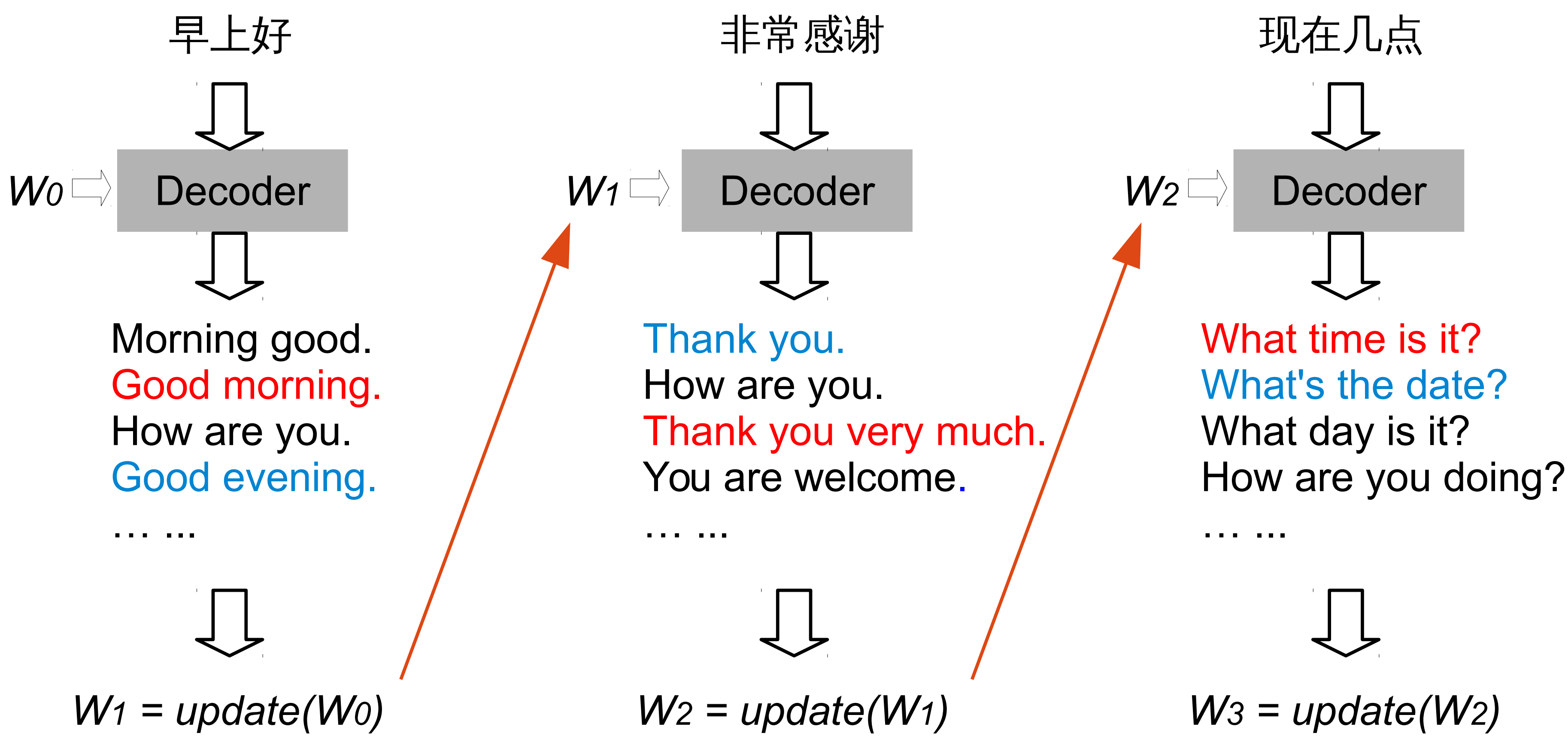
- Need careful feature engineering
- Training becomes much harder
- MERT no longer suitable ⇨

MIRA

- Margin Infused Relaxed Algorithm
- Online learning algorithm
- Capable of handling millions of features
- Theoretically sound
- Easy to implement

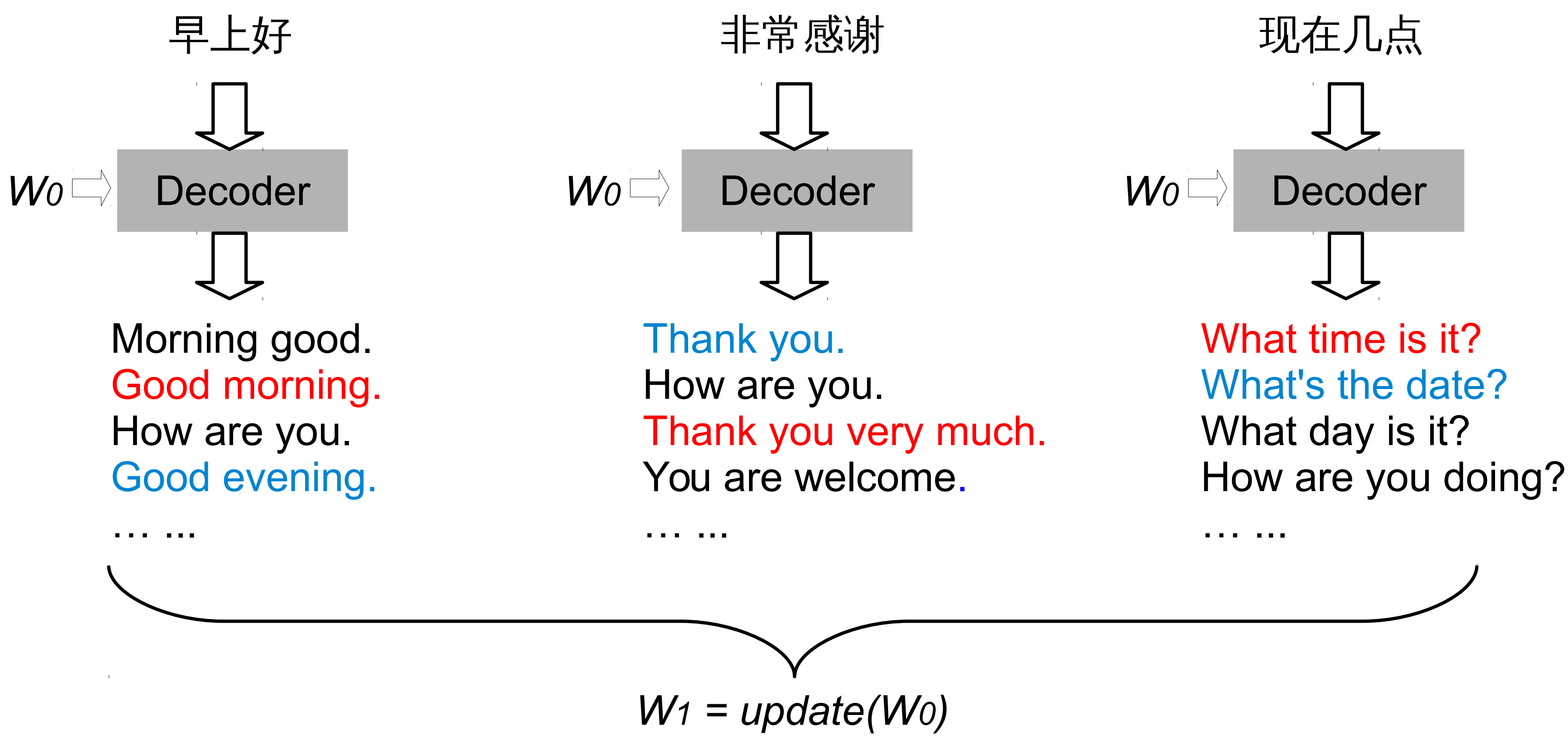
How it works

1. Online learning setting



How it works

1. Online learning setting



Compare: batch learning (eg. MERT)

How it works

2. Updating strategy

天空依然十分清澈

decode

Hypotheses	Features		
	Phrase	LM	Word
The sky remain clear	-192.3	-263.2	4
The sky remained clear	-176.2	-98.7	4
Sky is the clear	-250.5	-505.2	4
The sky is very clear	-187.8	-103.7	5
The sky is still clear	-210.6	-106.4	5
… …	… …		

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \lambda_t(\mathbf{h}_t^o - \mathbf{h}_t^p)$$

learning rate

How it works

3. Learning rate

- When loss is low, hope to decrease λ_t
- When loss is high, hope to increase λ_t
- However, λ_t should be bounded from above. Otherwise the algorithm might diverge.

$\Longrightarrow$ The MIRA learning rate:

metric score difference

$\mathbf{h}_t^o - \mathbf{h}_t^p$

$$\lambda_t = \min\left\{C, \frac{\rho_t - \mathbf{W}_t \cdot \Delta\mathbf{h}_t}{\|\Delta\mathbf{h}_t\|^2}\right\}$$

hyper-parameter

Where does it come from?

Theoretical Foundations

1. Concepts

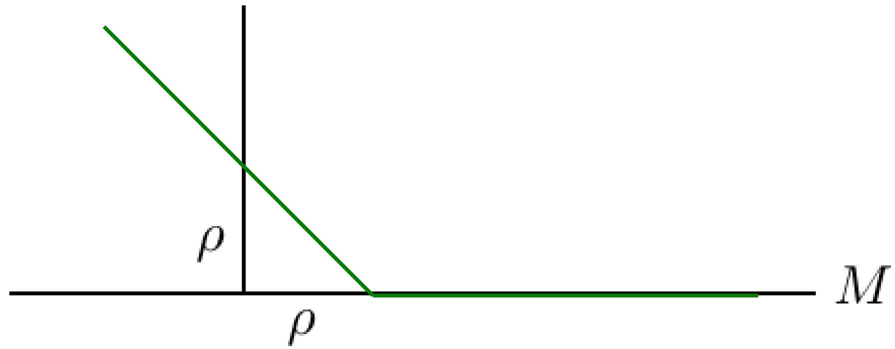
- Margin

$$M = \mathbf{W} \cdot \mathbf{h}^o - \mathbf{W} \cdot \mathbf{h}^p = \mathbf{W} \cdot \Delta\mathbf{h}$$

(Different in binary classification case: $M = y\mathbf{W} \cdot \mathbf{h}$)

- Hinge loss

$$\mathcal{L}(\mathbf{W}) = \begin{cases} 0 & \rho - M < 0 \\ \rho - M = \rho - \mathbf{W} \cdot \Delta\mathbf{h} & otherwise \end{cases}$$



Theoretical Foundations

2. Optimization problem

At round *t*, solve:

Passive

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\xi$$

Aggressive

$$s.t. \quad \mathcal{L}(\mathbf{w}) \leq \xi$$

slack variable

$$\xi \geq 0$$

Solution = MIRA:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \lambda_t \Delta\mathbf{h}_t$$
$$\lambda_t = \min\left\{C, \frac{\rho_t - \mathbf{W}_t \cdot \Delta\mathbf{h}_t}{\|\Delta\mathbf{h}_t\|^2}\right\}$$

- Passive-aggressive in nature
- Can be treated as online-SVM

Theoretical Foundations

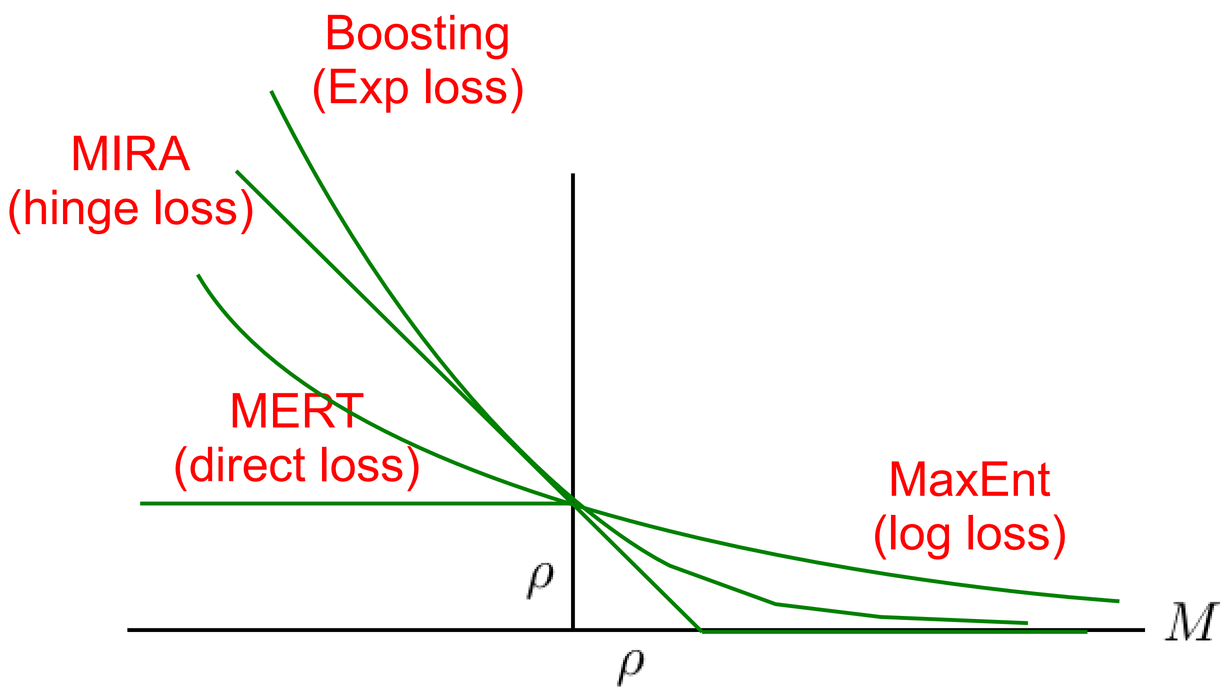
3. Performance guarantee

Theorem 1. *(Crammer et al., 2006) Let* $\|\mathbf{u}\|$ *be an arbitrary weight vector in* $\mathbb{R}^d$, $\mathcal{L}^t(\mathbf{w})$ *be the maximum loss at round* t *given a weight vector* $\mathbf{w}$, *and assume* $\forall t, \forall y \in \mathcal{Y}(x_t)$, $\|\mathbf{h}(x_t, y_t) - \mathbf{h}(x_t, y)\|_2 \leq R_2$. *The cumulative cost obtained by the MIRA algorithm is bounded from above by*

$$\sum_{t=1}^{T} \rho_t \leq \|\mathbf{u}\|_2^2 R_2^2 + 2CR_2^2 \sum_{t=1}^{T} \mathcal{L}^t(\mathbf{u})$$

Theoretical Foundations

4. A comparison of the loss function



- Direct loss is non-convex
- Hinge loss = tightest convex surrogate for direct loss

MIRA: A Summary

1. Updating strategy

when $\lambda_t = 1$: Perceptron

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \lambda_t \Delta \mathbf{h}_t$$
$$\lambda_t = \min\left\{C, \frac{\rho_t - \mathbf{W}_t \cdot \Delta \mathbf{h}_t}{\|\Delta \mathbf{h}_t\|^2}\right\}$$

2. Properties

- Online learning
- Large margin(hinge loss)
- Passive-aggressive
- Error upper-bound

Proper algorithm for large-scale discriminative training

MIRA in Practice

1. Choice of "oracle" and "prediction"

"oracle"

min cost

$$y^* = \operatorname*{argmin}_{z \in \mathcal{Y}(x)} cost(y, z)$$

$\boxed{\text{"hope"}}$

$$y^* = \operatorname*{argmax}_{z \in \mathcal{Y}(x)} \left(\mathbf{W} \cdot \mathbf{h}_z - cost(y, z)\right)$$

"prediction"

max model score

$$y' = \operatorname*{argmax}_{z \in \mathcal{Y}(x)} \mathbf{W} \cdot \mathbf{h}_z$$

$\boxed{\text{"fear"}}$

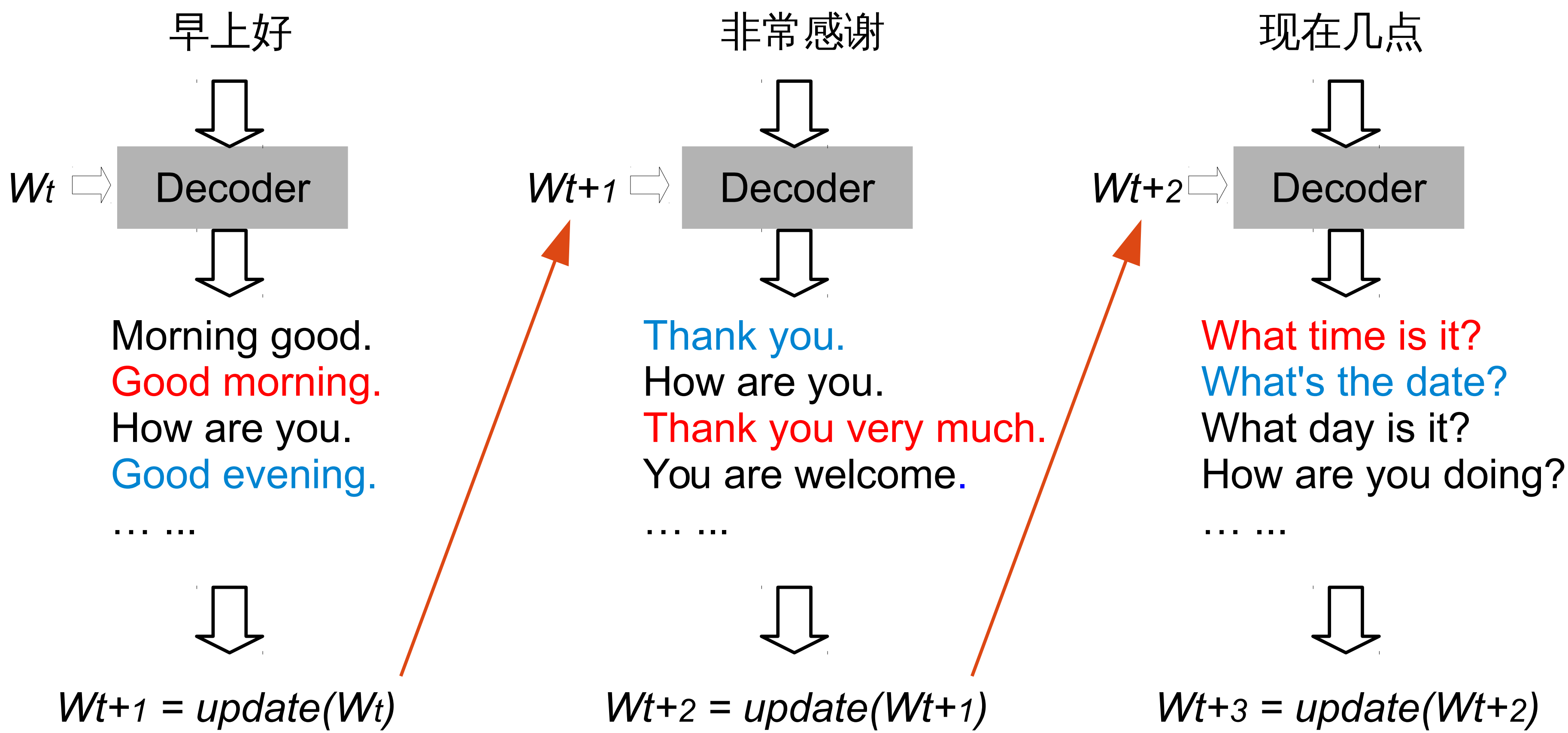
$$y' = \operatorname*{argmax}_{z \in \mathcal{Y}(x)} \left(\mathbf{W} \cdot \mathbf{h}_z + cost(y, z)\right)$$

max cost

$$y' = \operatorname*{argmax}_{z \in \mathcal{Y}(x)} cost(y, z)$$

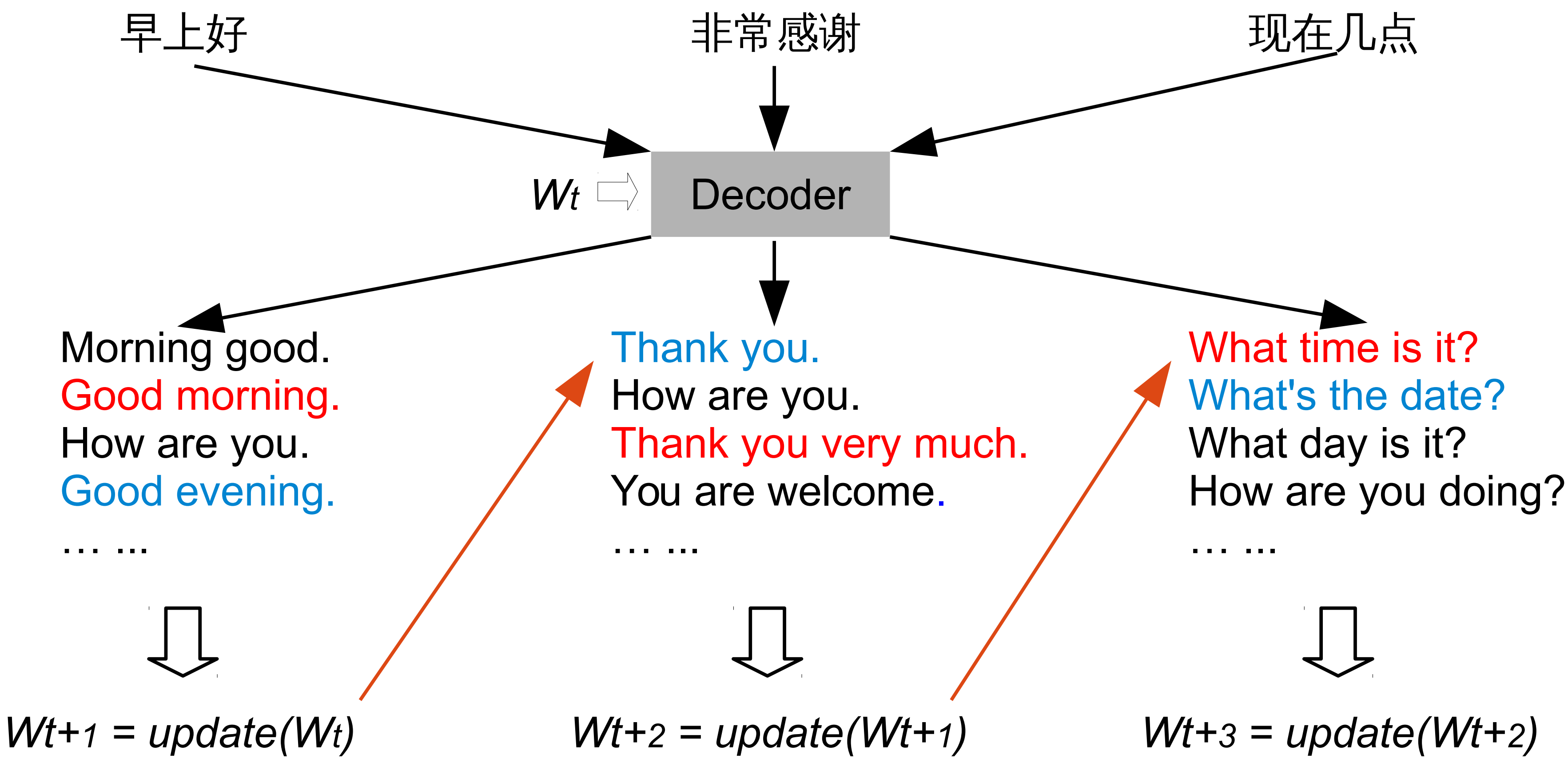
MIRA in Practice

2. Online learning setting



MIRA in Practice

2. Online learning setting



k-best MIRA

MIRA in Practice

3. Pseudo corpus for BLEU

- BLEU is not designed for sentence-level evaluation
- Compute BLEU in a context → create a pseudo-corpus

i-2	i-1	i
Morning good.	Thank you.	What time is it?
Good morning.	How are you.	What's the date?
How are you.	Thank you very much.	What day is it?
Good evening.	You are welcome.	How are you doing?
… ...	… ...	… ...

0.9^2*BLEUStats(i-2) + 0.9*BLEUStats(i-1) + BLEUStats(i)

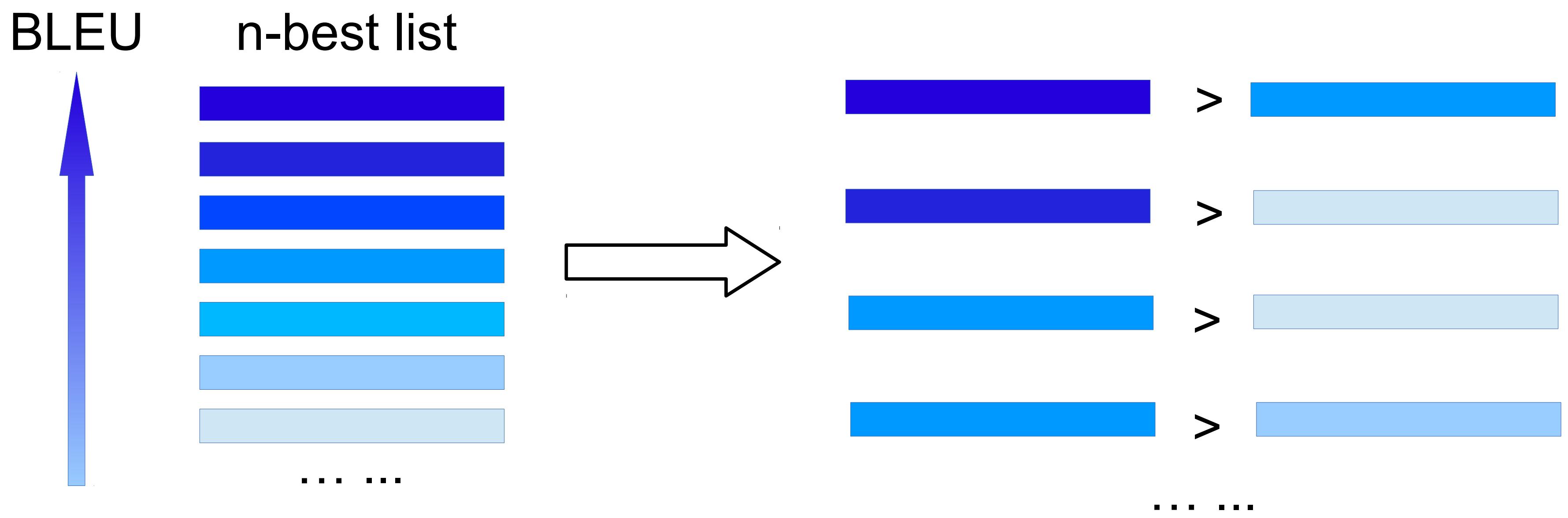
= Pseudo corpus BLEU stats

PRO

- Pairwise Ranking Optimization
- Batch learning algorithm
- Listwise ranking → pairwise ranking
- ≈ Ranking SVM

PRO

Idea: correct listwise ranking = correct pairwise ranking



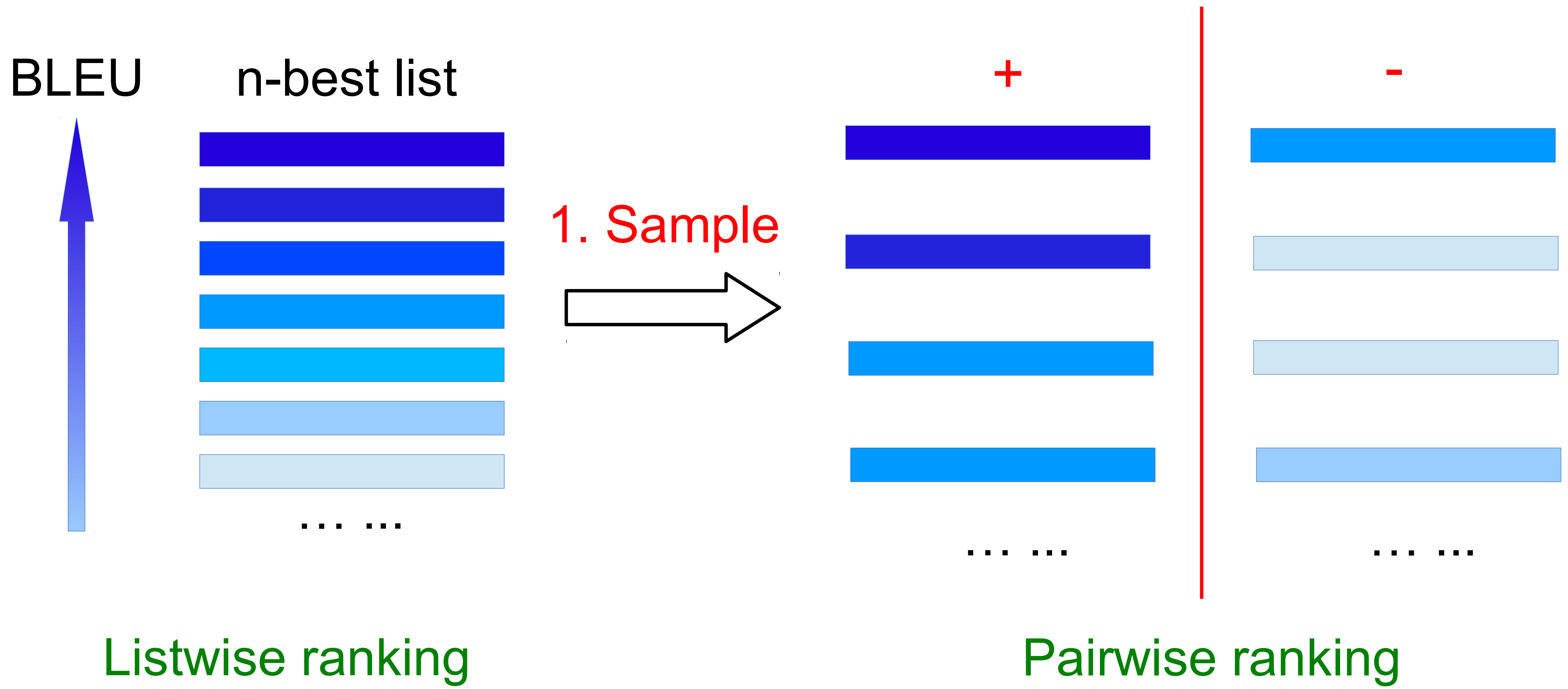
Listwise ranking

Pairwise ranking

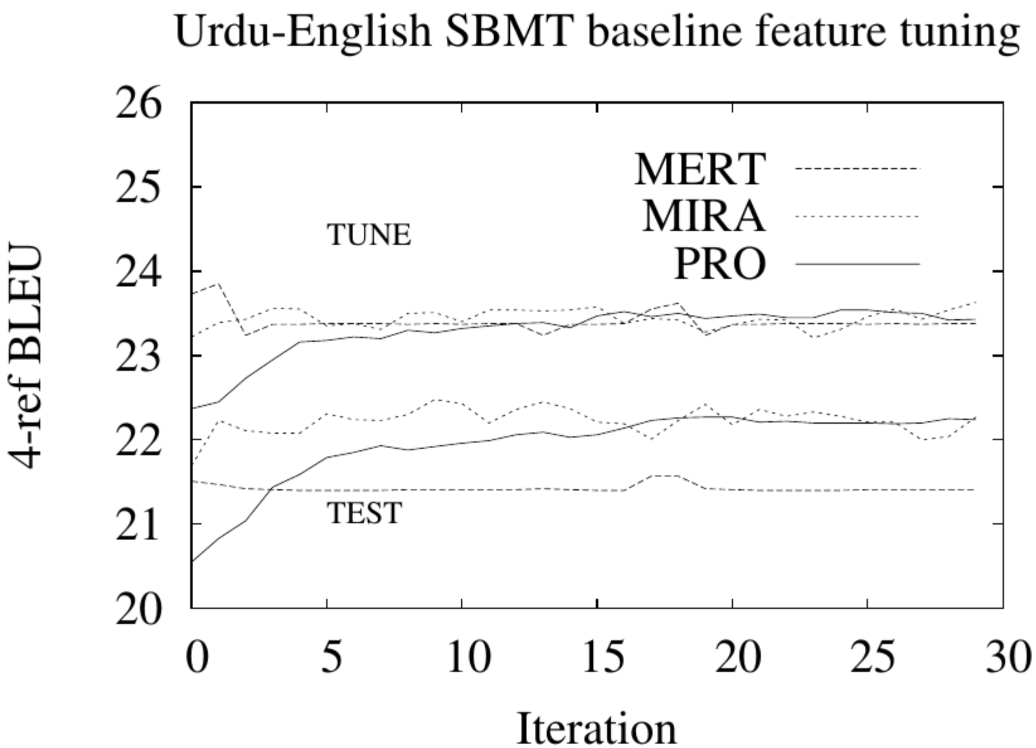
PRO

How to train?

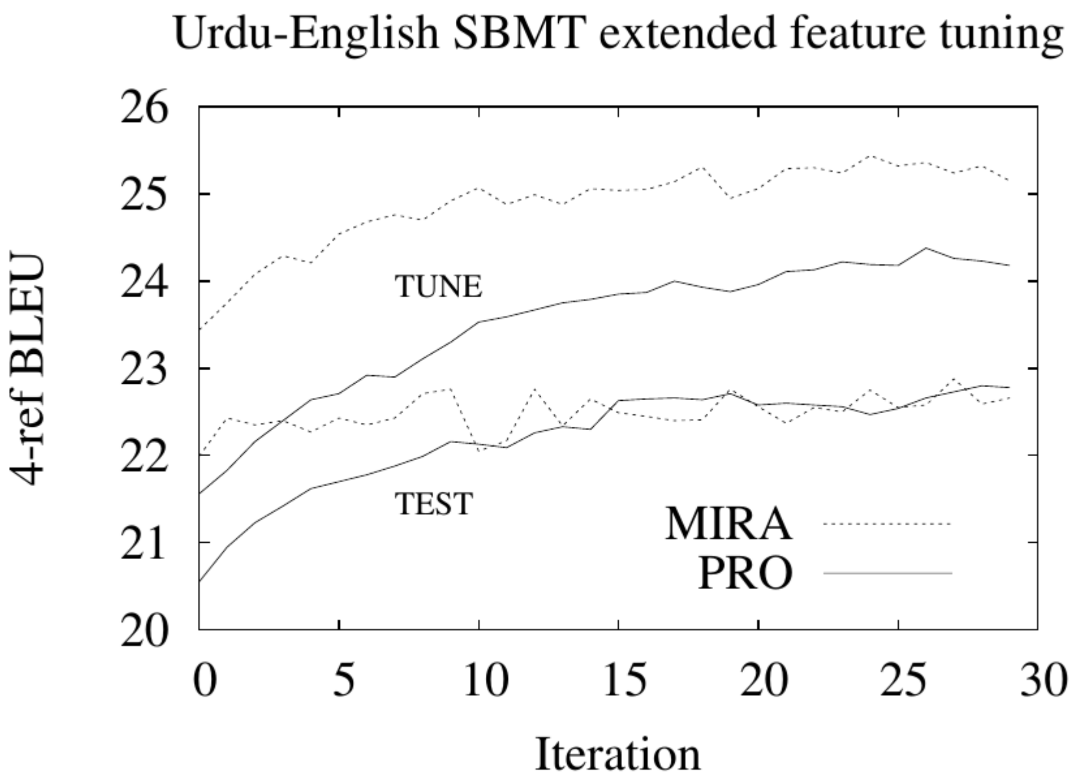
2. Train a binary classifier!
(using any classifier you like)



Performance Comparison



15 features



2250 features

Many More Algorithms...

- Batch learning

  M3N, SSVM, MinRisk, Rampion, HOLS, ...

- Online learning

  AROW, CW, AdaGrad, ORO, ...

- Still far from reaching the oracles