Structured Ramp Loss Minimization for Machine Translation

Kevin Gimpel Noah A. Smith

Carnegie Mellon

LEARNING IN MACHINE TRANSLATION

Why is learning in MT different from other tasks?

References often unreachable, so **surrogate references** are used instead (e.g., BLEU-oracles on *k*-best lists; Och & Ney, 2002)

How are learning algorithms affected?

Loss functions are changed:

Perceptron

Latent perceptron loss:

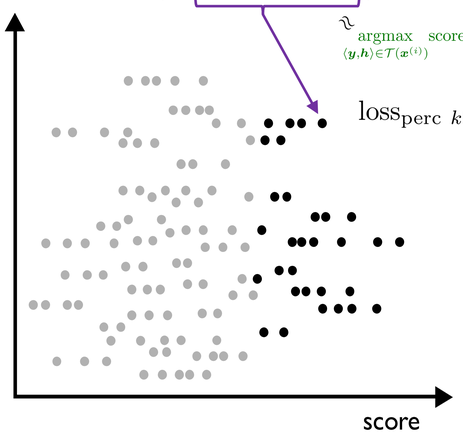
$$\text{loss}_{\text{perc}} = -\max_{\boldsymbol{h}:\langle \boldsymbol{y}^{(i)},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{h}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})$$

Latent perceptron with *k*-best BLEU oracle (Liang et al., 2006):

$$\text{loss}_{\text{perc } k\text{best}} = -\text{score}\left(\boldsymbol{x}^{(i)}, \underbrace{\operatorname{argmin}_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{K}_i} \left(\text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})\right)}\right) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})$$

$$\wr \operatorname{argmax}_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}(\boldsymbol{x}^{(i)})} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})$$

$$\text{loss}_{\text{perc } k\text{best}} \approx \text{loss}_{\text{ramp 2}}$$



MIRA

Latent hinge loss:

$$\text{loss}_{\text{hinge}} = -\max_{\boldsymbol{h}:\langle \boldsymbol{y}^{(i)},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{h}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})$$

MIRA for MT (Chiang et al., 2008; 2009):

$$\text{loss}_{\text{mira MT}} \approx -\max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})$$

$$\text{loss}_{\text{mira MT}} \approx \text{loss}_{\text{ramp 3}}$$

CRF

Latent log loss:

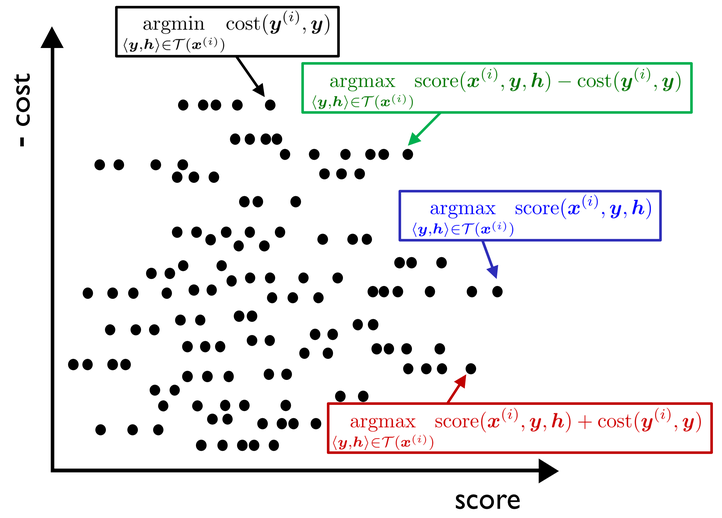
$$\text{loss}_{\log} = -\log \sum_{\boldsymbol{h}:\langle \boldsymbol{y}^{(i)},\boldsymbol{h}\rangle \in \mathcal{T}_i} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{h})\right\} + \log \sum_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})\right\}$$

Log loss for MT (Och & Ney, 2002):

$$\text{loss}_{\log \text{ MT}} \approx -\max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) + \log \sum_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \exp\left\{\text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})\right\}$$

$$\text{loss}_{\log \text{ MT}} \approx \text{loss}_{\text{soft ramp 2}}$$

STRUCTURED RAMP LOSSES



$$\text{loss}_{\text{ramp 1}} = -\max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})$$

$$\text{loss}_{\text{ramp 2}} = -\max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h})$$

$$\text{loss}_{\text{ramp 3}} = -\max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) - \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y}) + \max_{\langle \boldsymbol{y},\boldsymbol{h}\rangle \in \mathcal{T}_i} \text{score}(\boldsymbol{x}^{(i)}, \boldsymbol{y}, \boldsymbol{h}) + \text{cost}(\boldsymbol{y}^{(i)}, \boldsymbol{y})$$

ALGORITHM

We optimize $\text{loss}_{\text{ramp 3}}$ using a concave-convex procedure (CCCP; Yuille & Rangarajan, 2002).
CCCP is a batch optimization algorithm for a sum of a concave and a convex function.

Input: inputs $\{\boldsymbol{x}^{(i)}\}_{i=1}^N$, references $\{\boldsymbol{y}^{(i)}\}_{i=1}^N$, initial weights $\boldsymbol{\theta}_0$, # iterations T, # CCCP iterations T', # SSD iterations T'', k-best list size k, step size η, ℓ_2 regularization coefficient C
Output: learned weights: $\boldsymbol{\theta}$

```
θ ← θ_0;
for iter ← 1 to T do                                   // run Rampion for T iterations
    {K_i}_{i=1}^N ← Decode({x^(i)}_{i=1}^N, θ, k);      // get k-best lists
    for iter' ← 1 to T' do          // run CCCP (concave-convex procedure) for T' iterations
        for i ← 1 to N do                  // impute ''hope'' translations for all sentences
            ⟨y_i^+, h_i^+⟩ ← argmax_{⟨y,h⟩∈K_i} score_i(y, h; θ) − cost_i(y);
        end
        for iter'' ← 1 to T'' do          // run stochastic subgradient descent for T'' epochs
            for i ← 1 to N do
                ⟨y^−, h^−⟩ ← argmax_{⟨y,h⟩∈K_i} score_i(y, h; θ) + cost_i(y);   // find ''fear'' translation
                θ −= ηC(θ − θ_0)/N;                          // regularize back to θ_0, not 0
                θ += η(f(x^(i), y_i^+, h_i^+) − f(x^(i), y^−, h^−));   // subgradient update for loss
            end
        end
    end
end
return θ;                                          // return final parameters (no averaging needed)
```

Algorithm 1. RAMPION *Campanula rapunculus*

"A hardy biennial, cultivated for the use of its fleshy roots in salads, either boiled or in a raw state, generally the latter; the leaves are also used in winter salads" (Nicholson, 1884)

EXPERIMENTS

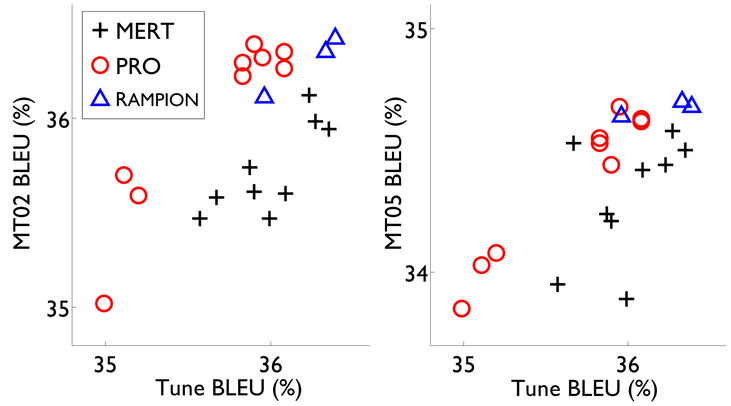
Small-Feature Experiments

Moses phrase-based MT system, 14 default features, default Moses initialization, 3 runs of each algorithm

Method	Urdu-English		Chinese-English		Arabic-English			Avg
	MT08*	MT09	MT02	MT05	MT05	MT08 NW	MT08 WB	
MERT	**24.5** ±0.1	**24.6** ±0.0	35.7 ±0.3	34.2 ±0.2	55.0 ±0.7	**49.8** ±0.3	**32.6** ±0.2	36.6
PRO	24.2 ±0.1	24.2 ±0.1	36.3 ±0.1	34.5 ±0.0	**55.6** ±0.3	49.6 ±0.0	31.7 ±0.0	36.6
RAMPION	**24.5**	**24.6**	**36.4**	**34.7**	55.5	**49.8**	32.1	**36.8**

Sensitivity Analysis

Chinese-English, 14 default Moses features
3 initializers (default Moses initialization + 2 random initializers)
3 runs of each algorithm for each initializer



Large-Feature Experiments

14 default Moses features + 7,200 additional features:
1k most frequent bilingual word pairs
200 most frequent unigrams, 1k most frequent bigrams, 1k most frequent trigrams
4k top trigger pairs, ranked by mutual information (Rosenfeld, 1996)

Method	Urdu-English			Chinese-English		
	Tune	MT08*	MT09	Tune	MT02	MT05
PRO	**29.4**	22.3	23.0	**40.9**	35.7	33.6
RAMPION	27.8	**24.2**	**24.6**	38.8	**36.2**	**34.3**

Code available for Moses: `www.ark.cs.cmu.edu/MT`
Also in cdec: `www.cdec-decoder.org`

REFERENCES

D. Chiang, Y. Marton, and P. Resnik. (2008) "Online large-margin training of syntactic and structural translation features." EMNLP.
D. Chiang, W. Wang, and K. Knight. (2009) "11,001 new features for statistical machine translation." NAACL.
P. Liang, A. Bouchard-Cote, D. Klein, and B. Taskar. (2006) "An end-to-end discriminative approach to machine translation." ACL.
G. Nicholson. (1884) *The Illustrated Dictionary of Gardening, Div. VI.* London: L. Upcott Gill.
F. Och and H. Ney. (2002) "Discriminative training and maximum entropy models for statistical machine translation." ACL.
R. Rosenfeld. (1996) "A maximum entropy approach to adaptive statistical language modeling." *Computer, Speech, and Language*, 10(3).
A. Yuille and A. Rangarajan. (2002) "The concave-convex procedure (CCCP)." NIPS.