Online Learning for CAT applications

Nicolò Cesa-Bianchi   Gabriele Reverberi

Università degli Studi di Milano

SMART
Statistical Multilingual Analysis
for Retrieval and Translation

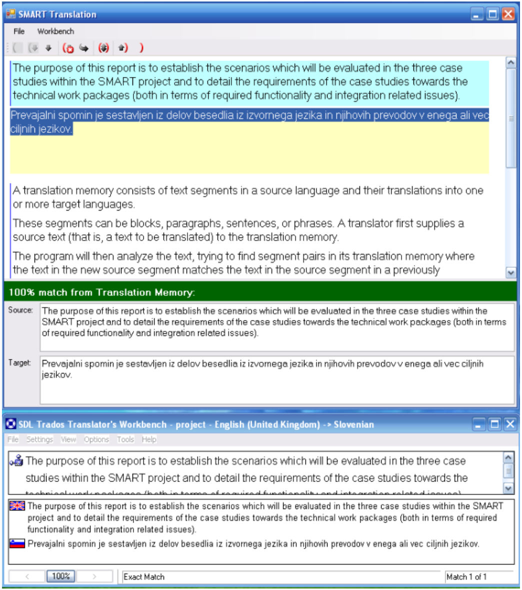
What is SMART about

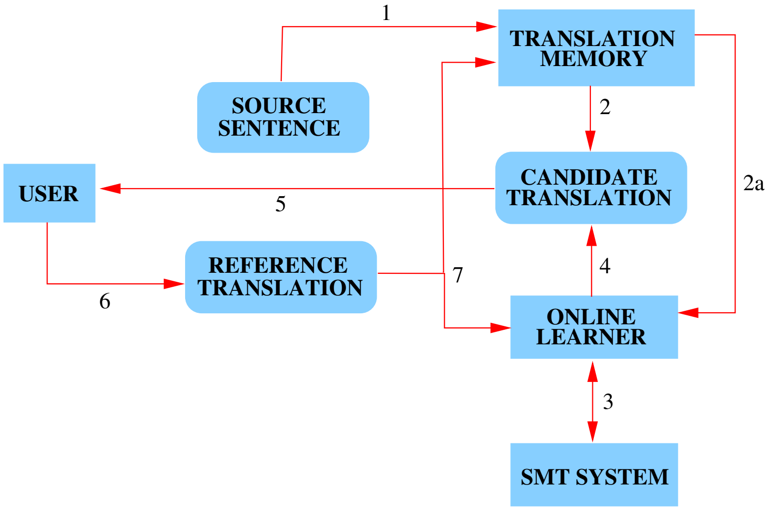
Online learning in SMT

Interactive Machine Translation

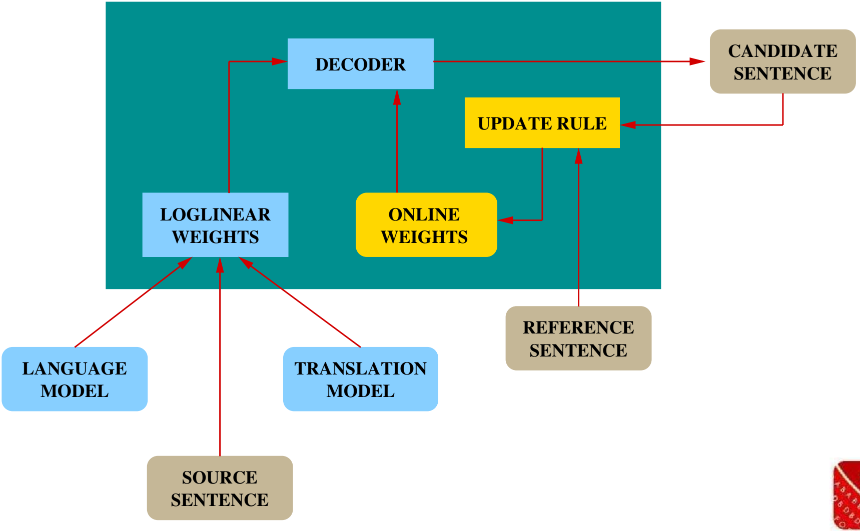
- Learning/optimization techniques are used to tune the parameters of SMT systems
- Online learning adjusts parameters incrementally
  [Lian et al., 2006; Arun and Koehn, 2007; Tillman and Zhang, 2008]
- Especially useful when the system interacts with the user

Computer Assisted Translation (CAT)




TRANSLATION MEMORY
SOURCE SENTENCE
USER
CANDIDATE TRANSLATION
REFERENCE TRANSLATION
ONLINE LEARNER
SMT SYSTEM
1
2
2a
3
4
5
6
7

Adaptive decoding [Liang, Bouchard-Côté, Klein, and Taskar, 2006]



Experimental setup (based on Portage SMT system)

Feature set for online weights

A new feature is created for each phrasetable entry

Experimental setup (based on Portage SMT system)

Feature set for online weights

A new feature is created for each phrasetable entry

Phase 1 – offline mode

- Building of phrasetable on a training corpus
- Tuning of loglinear weights on a development corpus

$\rightarrow$ This gives the baseline system

Experimental setup (based on Portage SMT system)

Feature set for online weights

A new feature is created for each phrasetable entry

Phase 1 – offline mode

- Building of phrasetable on a training corpus
- Tuning of loglinear weights on a development corpus

→ This gives the baseline system

Phase 2 – online mode

Online weights are adapted during CAT process

Adaptive decoding — basic definitions

- $\mathbf{f}(x_t, y)$ is the vector of phrasetable feature values when considering y as candidate translation for the source sentence x_t
- The vector $\boldsymbol{w}$ contains the decoder online weights
- The decoder builds a N-best list Y_t of candidate translations y by ranking them according to margin

$$\boldsymbol{w}^\top \mathbf{f}(x_t, y)$$

Adaptive decoding — basic definitions

- $\mathsf{f}(x_t, y)$ is the vector of phrasetable feature values when considering y as candidate translation for the source sentence x_t
- The vector $\boldsymbol{w}$ contains the decoder online weights
- The decoder builds a N-best list Y_t of candidate translations y by ranking them according to margin

$$\boldsymbol{w}^\top \mathsf{f}(x_t, y)$$

- The 1-best translation is $\widehat{y}_t = \underset{y \in Y_t}{\operatorname{argmax}}\, \boldsymbol{w}^\top \mathsf{f}(x_t, y)$
- The pseudo-target translation is $y_t^* = \underset{y \in Y_t}{\operatorname{argmax}}\, \text{BLEU}(y_t, y)$

Adaptive decoding — basic algorithmic framework

Recall:

Decoder ranks translations y according to $\boldsymbol{w}^\top \mathbf{f}(x_t, y)$

- Margin difference for weight $\boldsymbol{w}$ when y is chosen instead of y^*

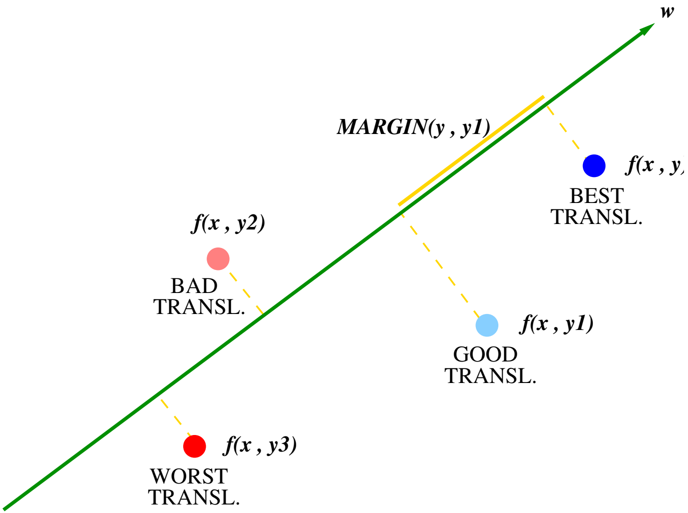
$$\textsc{margin}_t(y^*, y) = \boldsymbol{w}^\top \big(\mathbf{f}(x_t, y^*) - \mathbf{f}(x_t, y)\big)$$

- Linear constraints the learner tries to enforce at each step t

$$\textsc{margin}_t(y^*, y) \geqslant \textsc{bleu}(y_t, y_t^*) - \textsc{bleu}(y_t, y) \quad \forall y \in Y_t$$

- Constraints are approximately enforced by projecting current $\boldsymbol{w}$ onto (some of the) hyperplanes defined by constraints

Cost-sensitive margin condition



Update of parameters

Recall:

y = reference translation
y^* = pseudo-target translation (highest BLEU in N-best)
$\hat{y}$ = guessed translation (1-best)
$\boldsymbol{w}$ = current value of online weights

Enforce margin difference between pseudo-target y^* and 1-best $\hat{y}$

$$\min_{\boldsymbol{w}',\xi} \left\|\boldsymbol{w} - \boldsymbol{w}'\right\|^2 + C\,\xi$$

such that $\quad \text{MARGIN}(y^*, \hat{y}) \geqslant \big(\text{BLEU}(y, y^*) - \text{BLEU}(y, \hat{y})\big) - \xi$

Passive-aggressive update [Crammer et al., 2006]

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta_t \big(\text{BLEU}(y_t, y_t^*) - \text{BLEU}(y_t, \hat{y}_t)\big)$$

Theoretical guarantees

For any sequence $(x_1, y_1), (x_2, y_2), \dots$ of source/reference pairs

- If there exists choice $\mathbf{u}$ for the parameters that satisfies all constraints at each step, then

$$\sum_t \text{BLEU}(y_t, \widehat{y}_t) \geqslant \sum_t \text{BLEU}(y_t, y_t^*) - \|\mathbf{u}\|^2$$

- If no such $\mathbf{u}$ exists, then $\sum_t \text{BLEU}(y_t, \widehat{y}_t)$ is at least

$$\sum_t \text{BLEU}(y_t, y_t^*) - \inf_{\mathbf{u}} \left(1 + \frac{1}{C}\right) \left(\|\mathbf{u}\|^2 + C \sum_t H_t(\mathbf{u})\right)$$

- C is the aggressiveness parameter associated with the constraints
- $H_t(\mathbf{u})$ measures by how much the margin of $\mathbf{u}$ fails the worst constraint at time t

Performance measure

- Learning algorithms and their analysis do not require BLEU
- For robustness reasons, we train and test the system using BLEUMIX, an average of different sentence-level measures (1 BLEUMIX $\approx$ 0.65 BLEU)

Performance measure

- Learning algorithms and their analysis do not require BLEU
- For robustness reasons, we train and test the system using BLEUMIX, an average of different sentence-level measures (1 BLEUMIX $\approx$ 0.65 BLEU)

Cumulative BLEUMIX difference

The cumulative difference in sentence-level BLEUMIX points between online system translations $\widehat{y}_t$ and Portage baseline translations y'_t with respect to the common reference translation y_t

$$\sum_{t=1}^{T} \Big(\text{BLEUMIX}(y_t, \widehat{y}_t) - \text{BLEUMIX}(y_t, y'_t) \Big)$$

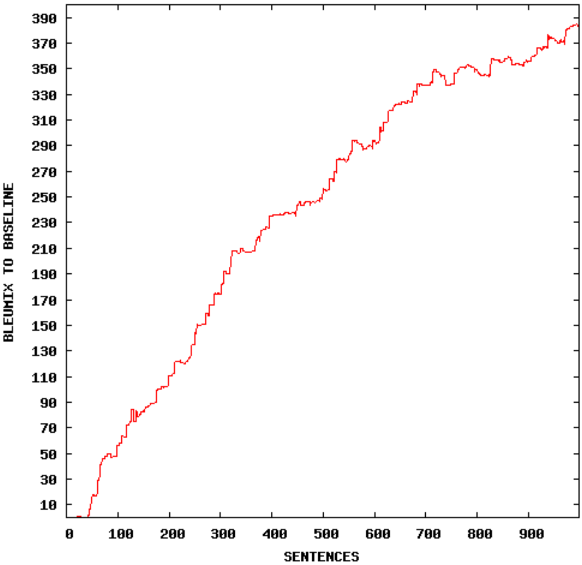
Experimental setup

- Corpus: English → Spanish section of Europarl
- Training set: 165,000 sentences
- Dev set: (used to tune Portage) 6,000 sentences
- Test set: (used for online learning)
  Five adjacent nonoverlapping blocks of 1,000 sentences each

Caveat

- Online learner attempts to improve on tuned Portage performance by a *single run* over 1,000 sentences
  → less than 0.6% of Portage training set!
- Learner does so by simultaneously tuning 1,7M parameters associated with the phrasetable entries
  → about 1,700 parameters per observed sentence!
- We get an improvement of about 0.4 BLEUMIX points per observed sentence

Weight adaptation



Oracolar phrasetable adaptation

Dynamic growth of phrasetable ☺

Problem: on-the-fly alignment of new segments ☹

Oracolar phrasetable adaptation

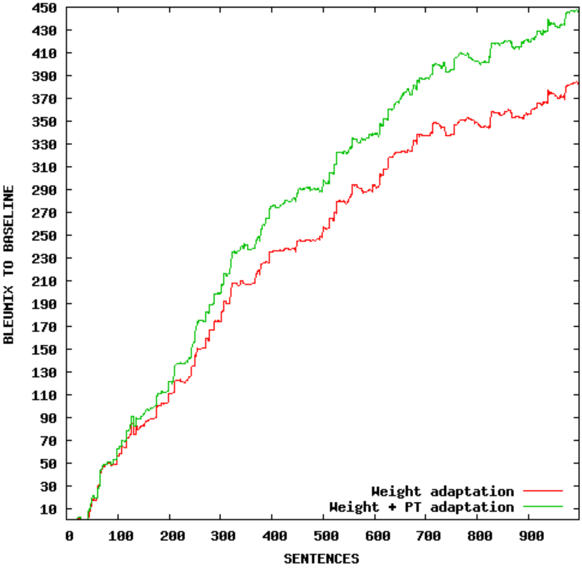
Dynamic growth of phrasetable ☺

Problem: on-the-fly alignment of new segments ☹

Oracolar PT

- Fake alignment by building an oracolar PT on train + test corpora
- After translating each new sentence, the relevant segments are moved from the oracolar PT to the working PT
- The weights associated with new segments are incrementally learned

Weight adaptation + PT adaptation



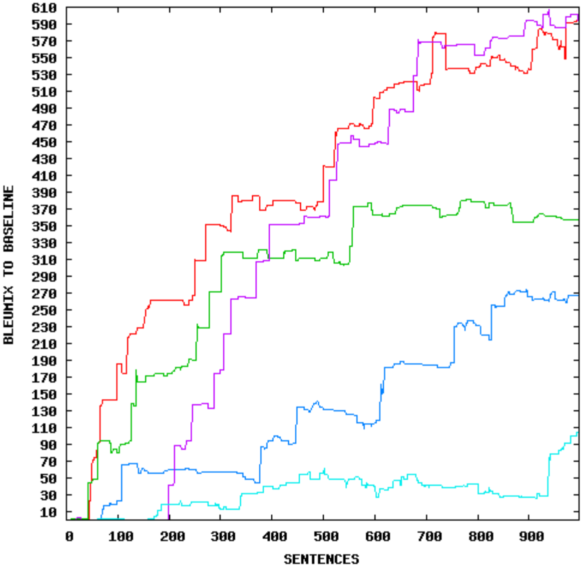
Significance analysis

Nonparametric randomized test [Riezler and Maxwell III, 2005]

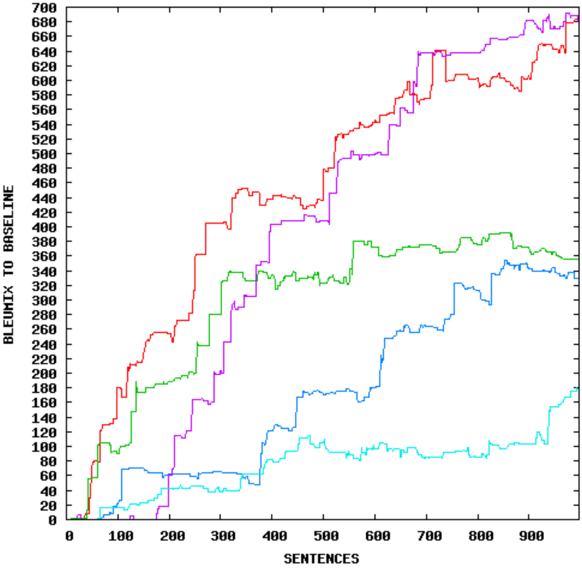
- We estimate the probability p that the performance difference increases when each translation in turn is obtained from a random system (adaptive or baseline)
- This is a p-value for the null hypothesis that baseline and adaptive have the same performance

p-values				
0.01	0.28	0.33	0.18	0.45
0.01	0.40	0.20	0.13	0.41

Weight adaptation — 5 runs



Weight adaptation + PT adaptation — 5 runs



Open issues

- More stable learning curves
- On-the-fly alignment to replace oracolar PTT
- TM's crippling effect