Online Learning for CAT applications

Nicolò Cesa-Bianchi    Gabriele Reverberi

Università degli Studi di Milano

SMARTEST
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
The purpose of this report is to establish the scenarios which will be evaluated in the three case studies within the SMART project and to detail the requirements of the case studies towards the technical work packages (both in terms of required functionality and integration related issues).

A translation memory consists of text segments in a source language and their translations into one or more target languages.

These segments can be blocks, paragraphs, sentences, or phrases. A translator first supplies a source text (that is, a text to be translated) to the translation memory.

The program will then analyze the text, trying to find segment pairs in its translation memory where the text in the new source segment matches the text in the source segment in a previously
CAT meets online learning

```
USER

SOURCE SENTENCE

REFERENCE TRANSLATION

1

TRANSLATION MEMORY

CANDIDATE TRANSLATION

2

ONLINE LEARNER

3

SMT SYSTEM

4

2a

5

6

7

N. Cesa-Bianchi and G. Reverberi (UNIMI)  Online Learning for CAT

5 / 22
```
Adaptive decoding

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

The diagram illustrates the process of adaptive decoding, which involves updating the model online based on incoming data. The key components include:

- **Decoder**: Receives source sentence and references and outputs a candidate sentence.
- **Loglinear Weights**: Used to combine language model and translation model weights.
- **Online Weights**: Updated dynamically based on new data.
- **Language Model**: Provides language-based probabilities.
- **Translation Model**: Provides translation-based probabilities.
- **Update Rule**: Used to update weights based on new data.
- **Reference Sentence**: Used as a reference for evaluation.

The diagram shows the flow of information from source sentence through the decoder to the candidate sentence, with feedback loops for updating weights.
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

→ 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

- $f(x_t, y)$ is the vector of *phrasetable feature values* when considering $y$ as candidate translation for the source sentence $x_t$

- The vector $\mathbf{w}$ contains the decoder **online weights**

- The decoder builds a $N$-best list $Y_t$ of candidate translations $y$ by ranking them according to **margin**
  
  $$\mathbf{w}^\top f(x_t, y)$$
Adaptive decoding — basic definitions

- \( f(x_t, y) \) is the vector of phrasetable feature values when considering \( y \) as candidate translation for the source sentence \( x_t \).
- The vector \( \mathbf{w} \) contains the decoder online weights.
- The decoder builds a \( N \)-best list \( Y_t \) of candidate translations \( y \) by ranking them according to margin
  \[ \mathbf{w}^\top f(x_t, y) \]
  The 1-best translation is
  \[ \hat{y}_t = \arg\max_{y \in Y_t} \mathbf{w}^\top f(x_t, y) \]
- The pseudo-target translation is
  \[ y^*_t = \arg\max_{y \in Y_t} \text{BLEU}(y_t, y) \]
Adaptive decoding — basic algorithmic framework

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

- **Margin difference** for weight $\mathbf{w}$ when $y$ is chosen instead of $y^*$
  \[
  \text{MARGIN}_t(y^*, y) = \mathbf{w}^\top (f(x_t, y^*) - f(x_t, y))
  \]

- **Linear constraints** the learner tries to enforce at each step $t$
  \[
  \text{MARGIN}_t(y^*, y) \geq \text{BLEU}(y_t, y^*) - \text{BLEU}(y_t, y) \quad \forall y \in Y_t
  \]

- Constraints are approximately enforced by projecting current $\mathbf{w}$ onto (some of the) hyperplanes defined by constraints
Cost-sensitive margin condition

\[ f(x, y) \]

\[ MARGIN(y, y_1) \]

\[ f(x, y_1) \]

\[ f(x, y_2) \]

\[ f(x, y_3) \]

BEST TRANSL.

GOOD TRANSL.

WORST TRANSL.
**Update of parameters**

**Recall:**

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

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

\[
\min_{w', \xi} \| w - w' \|^2 + C \xi
\]

such that \( \text{MARGIN}(y^*, \hat{y}) \geq (\text{BLEU}(y, y^*) - \text{BLEU}(y, \hat{y})) - \xi \)

**Passive-aggressive update**

[Cràmmer et al., 2006]

\[
w \leftarrow w + \eta_t \left( \text{BLEU}(y_t, y^*_t) - \text{BLEU}(y_t, \hat{y}_t) \right)
\]
Theoretical guarantees

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

- If there exists choice \(u\) for the parameters that satisfies all constraints at each step, then
  \[
  \sum_t \text{BLEU}(y_t, \hat{y}_t) \geq \sum_t \text{BLEU}(y_t, y^*_t) - \|u\|^2
  \]

- If no such \(u\) exists, then \(\sum_t \text{BLEU}(y_t, \hat{y}_t)\) is at least
  \[
  \sum_t \text{BLEU}(y_t, y^*_t) - \inf_u \left(1 + \frac{1}{C}\right) \left(\|u\|^2 + C \sum_t H_t(u)\right)
  \]

- \(C\) is the aggressiveness parameter associated with the constraints
- \(H_t(u)\) measures by how much the margin of \(u\) fails the worst constraint at time \(t\)
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 ≈ 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 ≈ 0.65 BLEU).

Cumulative BLEUMIX difference

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

$$
\sum_{t=1}^{T} \left( \text{BLEUMIX}(y_t, \hat{y}_t) - \text{BLEUMIX}(y_t, y'_t) \right)
$$
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
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
Dynamic growth of phrasetable

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

Oracolar phrasetable adaptation
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
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.

<table>
<thead>
<tr>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01  0.28  0.33  0.18  0.45</td>
</tr>
<tr>
<td>0.01  0.40  0.20  0.13  0.41</td>
</tr>
</tbody>
</table>
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