ROBUST ESTIMATION OF FEATURE WEIGHTS IN STATISTICAL MACHINE TRANSLATION

Cristina España-Bonet and Lluís Màrquez

cristinae@lsi.upc.edu lluism@lsi.upc.edu

TALP Research Center - LSI Department - Universitat Politècnica de Catalunya

Summary

Motivation

In phrase-based SMT, weights of the several components are usually estimated via MERT on a development set.

FACT. Weights might not generalise well on different domain test sets.

GOAL. Readjust the weights to be more appropriate on those sets without the need for specialised data.

Method and results

This work combines MERT with a perceptron training to obtain more robust weights.

IN-DOMAIN TRAINING. An improvement of more than 2 points of BLEU with respect to the MERT baseline can be obtained.

OUT-OF-DOMAIN TRAINING. When using out-of-domain sets in both trainings slight improvements are still observed with the perceptron.

Scenario

Application scenario. Arabic-to-English translation.

Definition of In/Out-domain sets. Our criterion to classify the sets relies on their perplexity with respect to the training corpus:

	in-domain			out-domain	
	Trdev	Trtest	N05	N06	N08
ARA perp.	272	270	320	598	568
ENG perp.	129	133	145	205	227

Methodology

Fundamental equation

$$T(f) = \hat{e} = \mathrm{argmax}_e \log P(e|f) = \mathrm{argmax}_e \Sigma_m \lambda_m h_m(f,e) =$$

$$\lambda_{lm} \log P(e) + \lambda_d \log P_d(e,f) + \lambda_{lg} \log lex(f|e) + \lambda_{ld} \log lex(e|f) + \lambda_g \log P(f|e) + \lambda_d \log P(e|f) + \lambda_{ph} \log ph(e) + \lambda_w \log w(e)$$

System development

After the SMT training, weights are fitted on a development set:

STAGE 1

Minimum Error Rate Training.
Fitted weights: $\vec{\lambda}_0$

STAGE 2

Perceptron Training.
Update of each feature weight λ_0^j sentence by sentence so that the translation is closer to the best attainable one (see algorithm).

The algorithm

INPUT: Training data $\{(f_i,e_i)\}_{i=1}^T$, MERT initial weights $\vec{\lambda}_0$, N epochs, learning rate ϵ.

```
for each epoch n = 1, ..., N
  for each example f_i  i = 1, ..., T
      ê = decode(f_i, λ_i)
      guess:  ê[1]
      tgt:  argmax_j (BLEU(ê[j]))
      if h(f_i, guess) ≠ h(f_i, tgt) then
         λ_i := λ_i + ε·Δh(f_i, tgt, guess)
      end if
      Λ := Λ + λ_i
   end for
 end for
return (Λ/NT)
```

GOLD STANDARD (tgt)

Sentence with the highest (smoothed) BLEU score in the n-best list.

UPDATE RULE

Constant update rule only depending on the direction of change:

$$\Delta\vec{h} = \mathrm{sign}\left(\vec{h}(f,\mathrm{tgt}) - \vec{h}(f,\mathrm{guess})\right)$$

In-domain TRAINING

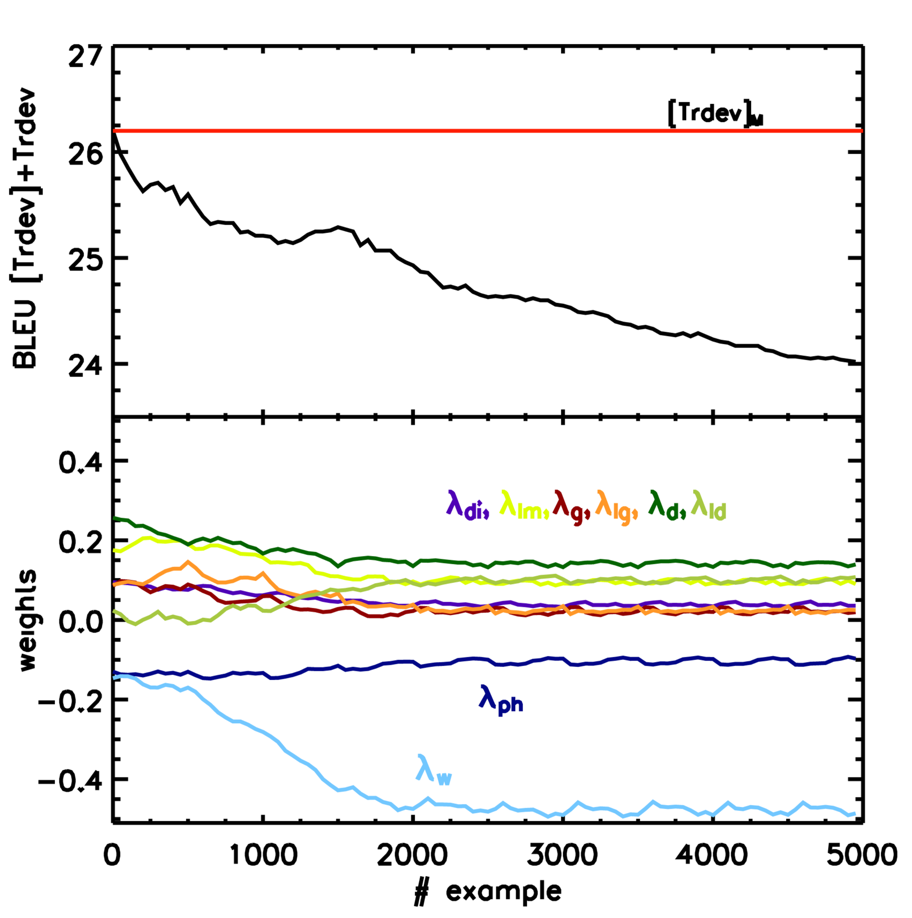

Figure: BLEU and weights evolution along training with Trdev.

The quality of the translation worsens on development along in-domain training with Trdev while perturbing the weights.

ON TEST

Still, the quality improves significantly on out-of-domain tests:

	in-domain		out-domain	
	Trtest	N05	N06	N08
M	23.87	43.76	30.24	29.06
M+P	22.77	**44.06**	**32.08**	**31.52**
M on test	24.27	45.46	32.96	32.77

Table: BLEU scores obtained by MERT (M) and by the combined training with the perceptron (M+P).

Out-of-domain TRAINING

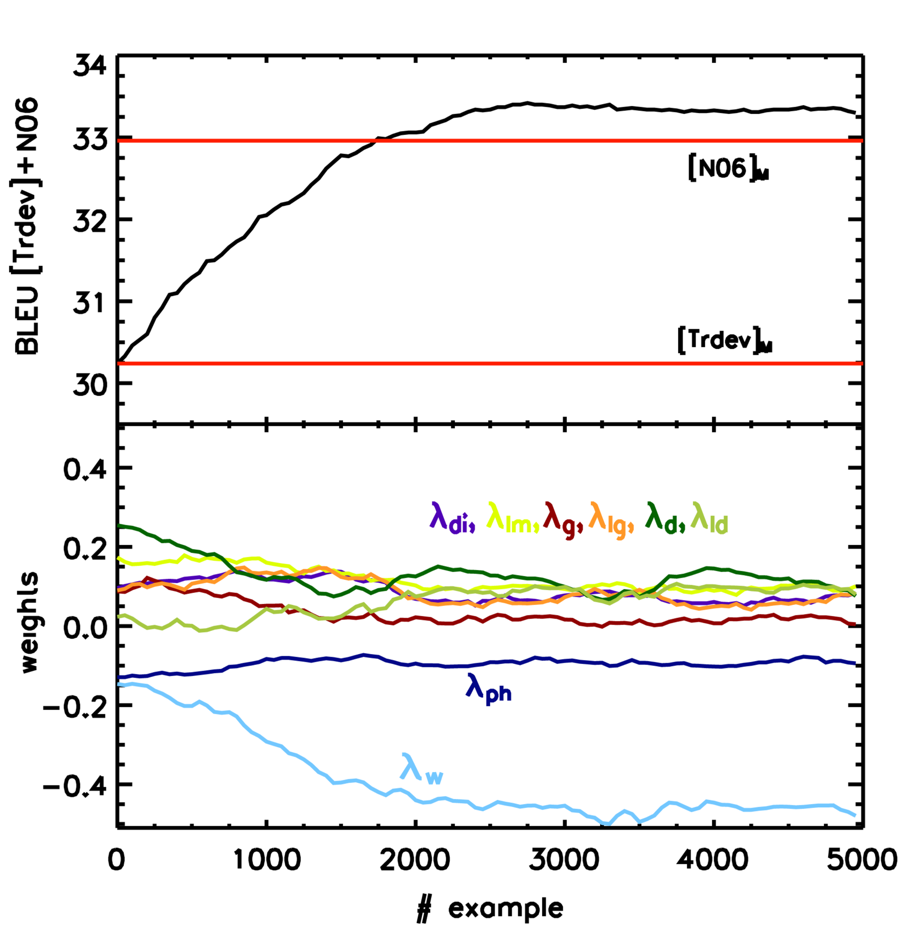

Figure: BLEU and weights evolution along training with N06.

During the perceptron training on N06 the quality of the translation is being improved. It gets a stable value over that of MERT on the same data set.

ON TEST

The improvement on out-of-domain test sets is even more evident in this case:

	in-domain		out-domain	
	Trtest	N05	N06	N08
M	23.87	43.76	-	29.06
M+P	21.98	43.10	-	**32.83**
M on test	24.27	45.46	-	32.77

Table: BLEU scores obtained by MERT (M) and by the combined training with the perceptron (M+P).

Comparison In/Out-domain Training on Out-of-domain N08 TEST

On an out-of-domain test set, both in-domain (blues) and out-of-domain (reds & greens) perceptron trainings improve MERT scores. The latter even surpass the *ficticius* value that MERT would obtain on N08, $[\mathrm{N08}]_\mathrm{M}$.

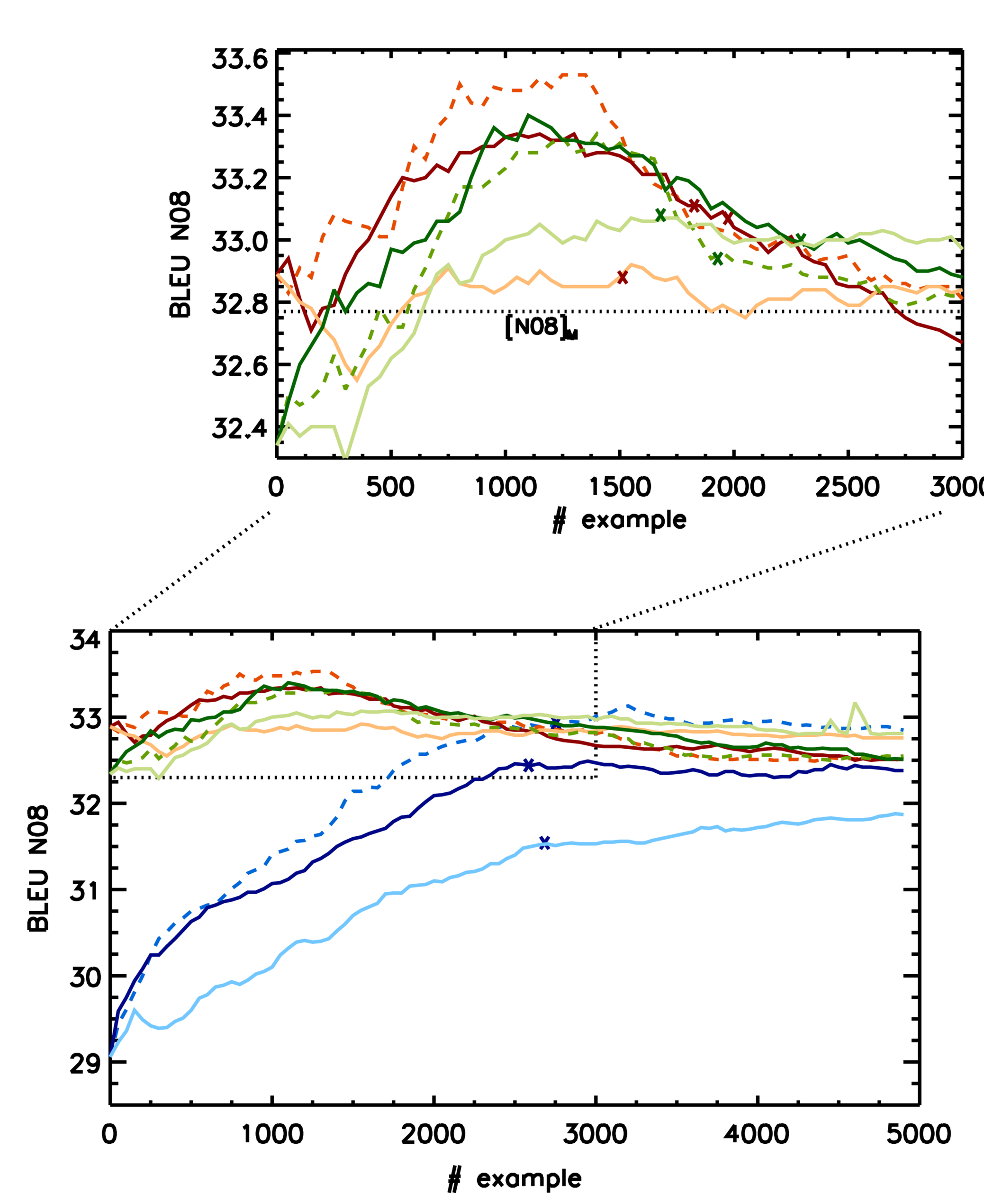

Figure: Evolution during the perceptron training of the BLEU score on the test set N08 for 9 different configurations: $[\mathrm{Trdev}]_\mathrm{M}$+Trdev, $[\mathrm{Trdev}]_\mathrm{M}$+N06, $[\mathrm{Trdev}]_\mathrm{M}$+TrdevN06, $[\mathrm{N06}]_\mathrm{M}$+Trdev, $[\mathrm{N06}]_\mathrm{M}$+N06, $[\mathrm{N06}]_\mathrm{M}$+TrdevN06, $[\mathrm{TrdevN06}]_\mathrm{M}$+Trdev, $[\mathrm{TrdevN06}]_\mathrm{M}$+N06, $[\mathrm{TrdevN06}]_\mathrm{M}$+TrdevN06.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement numbers 247914 and 247762 and from the Spanish Ministry of Science and Innovation (TIN2009-14675-C03).