Online Learning Approaches in Computer Assisted Translation

Prashant Mathur‡†, Mauro Cettolo†, Marcello Federico†
‡ University of Trento
† FBK - Fondazione Bruno Kessler
Trento, Italy
{prashant, cettolo, federico}@fbk.eu

Abstract

We present a novel online learning approach for statistical machine translation tailored to the computer assisted translation scenario. With the introduction of a simple online feature, we are able to adapt the translation model on the fly to the corrections made by the translators. Additionally, we do online adaption of the feature weights with a large margin algorithm. Our results show that our online adaptation technique outperforms the static phrase based statistical machine translation system by 6 BLEU points absolute, and a standard incremental adaptation approach by 2 BLEU points absolute.

1 Introduction

The growing needs of the localization and translation industry have recently boosted research around computer assisted translation (CAT) technology. The purpose of CAT is to increase the productivity of a human translator. A CAT tool comes as a package of a Translation Memory (TM), built-in spell checkers, a dictionary, a terminology list etc. which help the translator while translating a sentence. Recent research has led to the integration of CAT tools with statistical machine translation (SMT) engines. SMT makes use of a large available parallel corpus to generate statistical models for translation. Due to their generalization capability, SMT systems are a good fit in this scenario and a seamless integration of SMT engines in CAT have shown to increase translator's productivity (Federico et al., 2012).

Although automatic systems generate reliable translations they are not accurate enough to be used directly and need postedition by human translators. In state-of-the-art CAT tools, the SMT systems are static in nature and so they cannot adapt to these corrections. When a SMT system keeps repeating the same error, productivity of translators as well as their trust in SMT technology are negatively affected. As an example, technical documentation typically contains a lot of *repetitions* due to the employed writing style and pervasive use of terminology. Hence, in order to provide useful hints, SMT systems are expected to behave consistently regarding the translation of domain-specific terms. However, if the user edits the translation of a technical term in the target text, most current SMT systems are incapable to learn from those corrections.

Online learning is a machine learning task where a predictor iteratively: (1) receives an input and outputs a label, (2) receives the correct label from a human and if the two labels do not match, it learns from the mistake. The task of learning from user corrections at the sentence level fits well the online learning scenario, and its expected usefulness is clearly related to the amount of repetitions occurring in the text. The higher the number of repetititions in a document the more the SMT system has chances to translate consistently through the use of online learning.

In this paper, we implemented two online learning methods through which a phrase-based SMT system evolves over time, sentence after sentence, by taking advantage of the post-edition or translation of the previous sentence by the user.[1]

In the first approach, we focus on the translation model aspect of SMT which is represented by five conventional features, namely lexical and phrase translation probabilities in both directed and inverted directions, plus a phrase penalty score. Translation, language and reordering models are combined in a linear fashion to obtain a score for

[1]Moses code is available in the github repository. https://github.com/mtresearcher/mosesdecoder/tree/moses_onlinelearning

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 301–308,
Sofia, Bulgaria, August 8-9, 2013

the translation hypothesis as shown in Equation 1.

$$score(e^*, f) = \Sigma_i \lambda_i h_i(e^*, f) \quad (1)$$

where $h_i(\cdot)$ are the feature functions representing the models and λ_i are the linear weights. The highest scored translation is the best hypothesis e^* output by the system. We extend the translation model with a new feature which provides extra phrase-pair scores changing according to the user feedback. The scores of the new feature are adapted in a discriminative fashion, by rewarding phrase-pairs observed in the search space and in the reference, and penalizing phrase-pairs observed in the search space but not in the reference.

In the second approach, we also adapt the model weights of the linear combination after each test sentence by using a margin infused relaxed algorithm (MIRA).

For assessing the robustness of our methods, we performed experiments on two datasets from different domains and language pairs (§6). Moreover, our online learning approaches are compared against a static baseline system and against the incremental adaptation approach proposed by Levenberg et. al. (2010) (§5).

2 Related Works

Several online adaptation strategies have been proposed in the past, only a few deal with adaptation of post-edited/evaluation data while most works are on adaptation over development data during tuning of parameters (Och and Ney, 2003).

2.1 Online Adaptation during Tuning

Liang et. al. (2006) improved SMT performance by online adaptation of scaling factors (λ in (1)) using averaged perceptron algorithm (Collins, 2002). They presented different strategies to update the SMT models towards reference or oracle translation: (1) aggressively updating towards reference, *bold update*; (2) update towards the oracle translation in N-Best list, *local update*; (3) a hybrid approach in which a *bold* update is performed when the reference is reachable, otherwise a *local* update is performed. Liang and Klein (2009) compared two online EM algorithms, *stepwise online EM* (Sato and Ishii, 2000; Cappé and Moulines, 2007) and *incremental EM* (Neal and Hinton, 1998) which they use to update the alignment models (the generative component of SMT) on the fly. However, stepwise EM is prone to failure if mini-batch size and stepsize parameters are not chosen correctly, while incremental EM requires substantial storage costs because it has to store sufficient statistics for each sample. Other works on online minimum error rate training in SMT (Och and Ney, 2003) that deserve mentioning are (Hopkins and May, 2011; Hasler et al., 2011).

2.2 Online Adaptation during Decoding

Cesa-Bianchi et. al. (2008) proposed an online learning approach during decoding. They construct a layer of online weights over the regular feature weights and update these weights at sentence level using margin infused relaxed algorithm (Crammer and Singer, 2003); to our knowledge, this is the first work on online adaptation during decoding. Martínez-Gómez et. al. (2011; 2012) presented a comparison of online adaptation techniques in post editing scenario. They compared different adaptation strategies on scaling factors and feature functions (respectively, λ and $h(\cdot)$ in (1)). However, they modified the feature values during adaptation without any normalization, which disregards the initial assumption of the feature values being probabilities.

In our approach, the value of the additional *online feature* can be modified during decoding without changing other feature values (probabilities) and thus preserving their probability distribution.

3 Feature Adaptation

In the CAT scenario, the user receives a translation suggestion for each source segment, post-edits it and finally approves it. From the SMT point of view, for each source segment the decoder explores a search space of possible translations and finally returns the best scoring one (*bestHyp*) to the user. The user possibly corrects this suggestion thus generating the final translation (*postedit*).

Our online learning procedure is based on the following idea. For each N-best translation (*candidate*) in the search space, we compute a similarity score against the $postedit$ using the sentence-level BLEU metric (Lin and Och, 2004), a smoothed variant of the popular BLEU metric (Papineni et al., 2001). We hence compare the similarity score of each *candidate* against the similarity score achieved by the *bestHyp*, that was also computed against the *postedit*. If the *candidate*

scores better than the *bestHyp*, then we promote the building blocks, i.e. phrase-pairs, of *candidate* that were not used in *bestHyp* and demote the phrase-pairs used in *bestHyp* that were not used for *candidate*. On the contrary, if the *candidate* scores worse than the *bestHyp*, we promote the building blocks of *bestHyp* that are not in *candidate* and demote those of *candidate* that are not in *bestHyp*.

Our promotion/demotion mechanism could be implemented by updating the features values of the phrase pairs used in the *candidate* and *bestHyp* translations. However, features in the translation models are conditional probabilities and perturbing a subset of them by also preserving their normalization constraints can be computationally expensive. Instead, we propose to introduce an additional *online feature* which represents a goodness score of each phrase-pair in the test set.

We call the set of phrase pairs used to generate a *candidate* as $candidate_{PP}$ and the set of phrase pairs used to generate the *bestHyp* as $best_{PP}$. The online feature value of each phrase-pair is initialized to a constant and is updated according to the perceptron update (Rosenblatt, 1958) method. In particular, the amount by which a current feature value is rewarded or penalized depends on a learning rate α and on the difference between the model scores (i.e. $h \cdot w$) of *candidate* and *bestHyp* as calculated by the MT system. A sketch of our online learning procedure is shown in Algorithm 1.

Algorithm 1: Online Learning

```
foreach sourceSeg do
    bestHyp = Translate(sourceSeg);
    postedit = Human(bestHyp);
    for i = 1 → iterations do
        N-best=Nbest(source);
        foreach candidate ∈ N-best do
            sign = sgn |sBLEU(candidate) -
            sBLEU(bestHyp)| ;
            foreach phrasePair ∈ candidate_PP do
                if phrasePair ∉ best_PP then
                    f^i = f^{i-1} + (α · (Δh · w) ·
                    sign);
                end
            end
            foreach phrasePair ∈ best_PP do
                if phrasePair ∉ candidate_PP then
                    f^i = f^{i-1} - (α · (Δh · w) ·
                    sign);
                end
            end
        end
    end
end
```

In Algorithm 1, $\Delta h \cdot w$ is the above mentioned score difference as computed by the decoder; multiplied by α, it is the *margin*, that is the value with which the online feature score (f) of the phrase pair under processing is modified. We can observe that the feature scores are unbounded and could lead to instability of the algorithm; therefore, we normalise the scores through the sigmoid function:

$$f(x) = \frac{2}{1 + exp(x)} - 1 \tag{2}$$

4 Weight Adaptation

In addition to adapting the online feature values, we can also apply online adaptation on the feature weights of the linear combination (eq. 1). In particular, after translating each sentence we can adapt the parameters depending on how good the last translation was. A commonly used algorithm in this online paradigm for tuning of parameters is the Margin Infused Relaxed Algorithm (MIRA).

MIRA is an online large margin algorithm that updates the parameter $\hat{w}$ of a given model according to the loss that is occurred due to incorrect classification. In the case of SMT this margin can be coupled with the loss function, which in this case is the complement of the sentence level BLEU(sBLEU). Thus, the loss function can be formulated as:

$$l(\hat{y}) = sBLEU(y^*) - sBLEU(\hat{y}) \tag{3}$$

where y^* is the $oracle$ (closest translation to the reference) and $\hat{y}$ is the $candidate$ being processed. Ideally, this loss should correspond to the difference between the model scores:

$$\Delta h \cdot \hat{w} = score(y^*) - score(\hat{y}) \tag{4}$$

MIRA is an ultraconservative algorithm, meaning that the update of the current weight vector is the smallest possible value satisfying the constraint that the variation incurred by the objective function must not be larger than the variation incurred by the model (plus a non-negative slack variable ξ). Formally, weight update at i^{th} iteration is defined as:

$$w_i = \arg\min_{w} \frac{1}{2\eta} \underbrace{||w - w_{i-1}||^2}_{conservative} + \underbrace{C}_{aggressive} \sum_j \xi_j$$

subject to

$$l_j \leq \Delta h_j \cdot w + \xi_j \quad \forall j \in J \subseteq \{1 \dots N\} \tag{5}$$

where j ranges over all $candidates$ in the N-best list, l_j is the loss between $oracle$ and the $candidate\ j$, and $\Delta h_j \cdot w$ is the corresponding difference in the model scores. C is an aggressive parameter which controls the size of the update, η is the learning rate of the algorithm and ξ is usually a very small value (in our experiments we kept it as 0.0001). After partial differentiation and linearizing the loss, equation 5 can be rewritten as:

$$w_i = w_{i-1} + \eta \cdot \sum_j \alpha_j \cdot \Delta h_j$$

where

$$\alpha_j = min \left\{ C, \frac{l_j - \Delta h_j \cdot w}{||\Delta h_j||^2} \right\} \qquad (6)$$

We solve equation 5, by computing α with the optimizer integrated in the Moses toolkit by (Hasler et al., 2011). Algorithm 2 gives an overview of the online margin infused relaxed algorithm we implemented in Moses.

Algorithm 2: Online Margin Infused Relaxed

```
foreach sourceSeg do
    bestHyp = Translate(sourceSeg);
    postedit = Human(bestHyp);
    w_0 = w;
    for i = 1 → iterations do
        N-best=Nbest(sourceSeg,w_{i-1});
        foreach candidate_j ∈ N-best do
            if Δh_j · w + ξ_j ≥ l_j then
                α_j = Optimize(l_j, h_j, w, C);
                w_i = w_{i-1} + η · Σ_j α_j Δh_j;
            end
        end
    end
end
```

In the following section we overview a stream based adaptation method with which we experimentally compared our two online learning approaches as it well fits the framework we are working in.

5 Stream based adaptation

Continuously updating an SMT system to an incoming stream of parallel data comes under stream based adaptation. Levenberg et. al. (2010) proposed an incremental adaptation technique for the core generative component of the SMT system, word alignments and language models (Levenberg and Osborne, 2009). To get the word alignments on the new data they use a *Stepwise online EM* algorithm, where old counts (from previous alignment models) are interpolated with the new counts.

Since we work at the sentence level, on-the-fly computation of probabilities of translation and reordering models is expensive in terms of both computational and memory requirements. To save these costs, we prefer using dynamic suffix array approach described in (Levenberg et al., 2010; Callison-Burch et al., 2005; Lopez, 2008). They are used to efficiently store the source and the target corpus and alignments in efficient data structure, namely the suffix array. When a phrase translation is asked by the decoder, the corpus is searched, the counts are collected and its probabilities are computed on the fly. However, the current implementation in Moses of the stream based MT relying on the suffix arrays is severely limited as it allows the computation of only three translation features, namely the two direct translation probabilities and the phrase penalty. This results in a significant degradation of performance.

6 Experiments

6.1 Datasets

We compared our online learning approaches (Sections 3 and 4) and the stream based adaptation method (Section 5) on two datasets from different domains, namely Information Technology (IT) and TED talks, and two different language pairs. The IT domain dataset is proprietary, it involves the translation of technical documents from English to Italian and has been used in the field test carried out under the MateCat project[2]. Experiments are also conducted on English to French TED talks dataset (Cettolo et al., 2012) to assess the robustness of the proposed approaches in a different scenario and to provide results on a publicly available dataset for the sake of reproducibility. The training, development (`dev2010`) and evaluation (`tst2010`[3]) sets are the same as used in the last IWSLT last evaluation campaigns. In experiments on TED data, we considered the human reference translations as post edits, even if they were

[2] www.matecat.com

[3] As the size of evaluation set in TED data is too large with respect to the current implementation of our algorithms, we performed evaluation on the first 200 sentences only.

actually generated from scratch.

In our experiments, the extent of usefulness of online learning highly depends on the amount of repetition of text. A reasonable way to measure the quantity of repetition in each document is through the *repetition rate* (Bertoldi et al., 2013). It computes the rate of non-singleton *n*-grams, *n*=1...4, averaging the values over sub-samples `S` of thousand words from the text, and then combining the rate of each *n*-gram to a single score by using the geometric mean. Equation 7 shows the formula for calculating the repetition rate of a document, where `dict(n)` represents the total number of different *n*-grams and n_r is the number of different *n*-grams occurring exactly `r` times:

$$RR = \left(\prod_{n=1}^{4} \frac{\sum_S dict(n) - n_1}{\sum_S dict(n)} \right)^{1/4} \quad (7)$$

Statistics of the parallel sets and their repetition rate on both sides are reported in Table 1.

Domain	Set	#srcTok	srcRR	#tgtTok	tgtRR
$IT_{en \rightarrow it}$	Train	57M	na	60M	na
	Dev	3.3k	12.03	3.5k	11.87
	Test	3.3k	15.00	3.3k	14.57
$TED_{en \rightarrow fr}$	Train	2.6M	na	2.8M	na
	Dev	20k	3.43	20k	5.27
	Test	32k	4.08	34k	3.57

Table 1: Statistics of the parallel data along with the corresponding repetition rate (RR).

It can be noted that the repetition rates of IT and TED sets are significantly different, particularly high in IT documents, much lower in the TED talks.

6.2 Systems

The SMT systems were built using the Moses toolkit (Koehn et al., 2007). Training data in each domain was used to create translation and lexical reordering models. We created a 5-gram LM for TED talks and a 6-gram LM for the IT domain using IRSTLM (Federico et al., 2008) with improved Kneser-Ney smoothing (Chen and Goodman, 1996) on the target side of the training parallel corpora. The log linear weights for the baseline systems are optimized using MERT (Och, 2003) provided in the Moses toolkit. To counter the instability of MERT, we averaged the weights of three MERT runs in each case. Performance is measured in terms of BLEU and TER (Snover et al., 2006) computed using the MultEval script (Clark et al., 2011). Since the implementations of standard Giza and of incremental Giza combined with dynamic suffix arrays are not comparable, we constructed two baselines, a standard phrase based SMT system and an incremental Giza baseline (§5). Details on experimental SMT systems we built follow.

Baseline This system was built on the parallel training data for each domain. We run 5 iterations of model 1, 5 of HMM (Vogel et al., 1996), 3 of model 3, 3 of model 4 (Brown et al., 1993) using MGiza (Gao and Vogel, 2008) toolkit to align the parallel corpus at word level. Translation and reordering models were built using Moses, while log-linear weights were optimized with MERT on the corresponding development sets. The same IT baseline system was used in the field test of MateCat and the references in the IT data are actual postedits of its translation.

IncGiza Baseline We trained alignment models with incGiza++[4] with 5 iterations of model 1 and 10 iterations of the HMM model. To build incremental Giza baselines, we used dynamic suffix arrays as implemented in Moses which allow the addition of new parallel data during decoding. In the incremental Giza baseline, once a sentence of the test set is translated, the sentence pair (source and target post-edit/reference) along with the alignment provided by incGiza are added to the models.

Online learning systems We developed several online systems on top of the two aforementioned baseline systems: (1) +O employ the additional online feature (Section 3) updated with Algorithm 1; (2) +O+NS as (1) but with the online feature normalized with the sigmoid function; (3) +W weights updated (Section 4) with Algorithm 2; (4) +O+W combination of online feature and weight update; (5) +O+NS+W as system (4) with normalized online feature score.

In the online learning system we have three additional parameters: a weight for the online feature, a learning rate for features (used in the perceptron update), and a learning rate for feature weights used by MIRA. These additional parameters were optimized by maximizing the BLEU

[4] http://code.google.com/p/inc-giza-pp/

score on the devset and on top of already optimized feature weights. For practical reasons, optimization of the parameters was run with the Simplex algorithm (Nelder and Mead, 1965).

7 Results and Discussion

Tables 2 and 3 collect results by the systems described in Section 6.2 on the IT and TED translation tasks, respectively.

In Table 2, the online system (1st block "+O+NS+W" system with 10 iterations of online learning) shows significant improvements, over 6 BLEU points absolute above the baseline. In this case the online feature can clearly take advantage of the high repetition rates observed in the IT dev and test sets (Table 1). Similarly, in the second block, the online system (2nd block "+O+NS+W" with 10 iterations of online learning) outperforms IncGiza baseline, too. It is interesting to note that by continuously updating the baseline system after each translation step, even the plain translation models are capable to learn from the correction in the post-edited text.

Figure 1 depicts learning curve of *Baseline* system, "+O+NS" (referred as *+online feature*) and "+O+NS+W" (referred as *+MIRA*). We plotted incremental BLEU scores after translation of each sentence, thereby the last point on the plot shows the corpus level BLEU on the whole test set.

In Table 3, from the first block we can observe that online learning systems perform only slightly better than the baseline systems, the main reason being the low repetition rate observed in the evaluation set (as shown in Table 1). The positive results observed in the second block ("+O+W" with 10 iterations) are probably due to the larger room for improvement available for translation models implemented with dynamic suffix arrays, as they only incorporate 3 features instead of 5. Sometimes, online learning systems show worse results with higher numbers of iterations, which seems due to overfitting. It is also interesting to notice that after optimization the weight value of the online feature was 0.509 for the IT task and 0.072 for the TED talk task. This confirms the different use and potential assigned to the online feature by the SMT systems in the two tasks.

8 Conclusion

We have shown a new way to update the translation model on the fly without changing the original probability distribution. We empirically proved that this method is robust and works for different domain datasets be it Information Technology or TED talks. In addition, if the repetition rate is high in the text, online learning works much better than if the rate is low. We tested both with an unbounded and a bounded range on the online feature and found out that bounded values produce more stable and consistent results. From previous works, it has been proven that MIRA works well with sparse features too, so, as for the future plan we would like to treat each phrase pair as a sparse feature and tune the sparse weights using MIRA. From the results, it is evident that we have not used any sort of stopping criterion for online learning; a random of 1, 5 and 10 iterations were chosen in a naive way. Our future plan will extend to working on finding a stopping criterion for online learning process.

Acknowledgements

This work was supported by the MateCat project, which is funded by the EC under the 7^{th} Framework Programme.

References

N. Bertoldi, M. Cettolo, and M. Federico. 2013. Cache-based online adaptation for machine translation enhanced computer assisted translation. In *Proc. of MT Summit*, Nice, France.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

C. Callison-Burch, C. Bannard, and J. Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proc. of ACL*, pages 255–262, Ann Arbor, US-MI.

O. Cappé and E. Moulines. 2009. Online EM algorithm for latent data models. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 71(3):593–613.

N. Cesa-Bianchi, G. Reverberi, and S. Szedmak. 2008. Online learning algorithms for computer-assisted translation. Technical report, SMART project (`www.smart-project.eu`).

M. Cettolo, C. Girardi, and M. Federico. 2012. WIT^3: web inventory of transcribed and translated talks. In *Proc. of EAMT*, Trento, Italy.

S. F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL*, pages 310–318, Santa Cruz, US-CA.

System	Bleu (σ)			TER (σ)		
	1 Iter	5 Iter	10 Iter	1 Iter	5 Iter	10 Iter
Baseline	38.46(1.79)	-	-	39.98(1.35)	-	-
+O	39.88(1.77)	41.22(1.80)	41.16(1.74)	38.69(1.30)	37.78(1.32)	38.37(1.30)
+O+NS	39.91(1.80)	40.54(1.79)	40.71(1.76)	38.67(1.31)	38.21(1.29)	38.17(1.31)
+W	39.76(1.76)	38.16(1.77)	37.57(1.82)	38.58(1.27)	39.53(1.30)	39.93(1.30)
+O+W	41.23(1.66)	40.29(1.54)	29.36(1.45)	37.53(1.26)	38.03(1.24)	49.08(1.25)
+O+NS+W	41.19(1.86)	43.07(1.87)	**45.13(1.74)**	37.60(1.35)	36.43(1.43)	**34.53(1.36)**
IncGiza Baseline	28.48(1.50)	-	-	49.23(1.43)	-	-
+O	29.34(1.51)	27.80(1.49)	27.52(1.38)	47.86(1.41)	48.20(1.30)	51.01(1.53)
+O+NS	28.69(1.53)	29.68(1.45)	29.36(1.49)	48.21(1.45)	47.51(1.45)	47.92(1.45)
+W	28.25(1.56)	27.68(1.53)	27.57(1.50)	49.05(1.43)	48.74(1.36)	48.10(1.23)
+O+W	29.36(1.61)	29.94(1.64)	25.95(1.25)	47.15(1.41)	46.56(1.31)	50.31(1.15)
+O+NS+W	29.76(1.49)	30.28(1.54)	**30.83(1.60)**	46.62(1.39)	45.60(1.28)	**46.54(1.31)**

Table 2: Result on the IT domain task (EN>IT). Baseline is a standard phrase based SMT system, +O has the online feature, +NS adds normalization of online feature, +W has online weight adaptation.

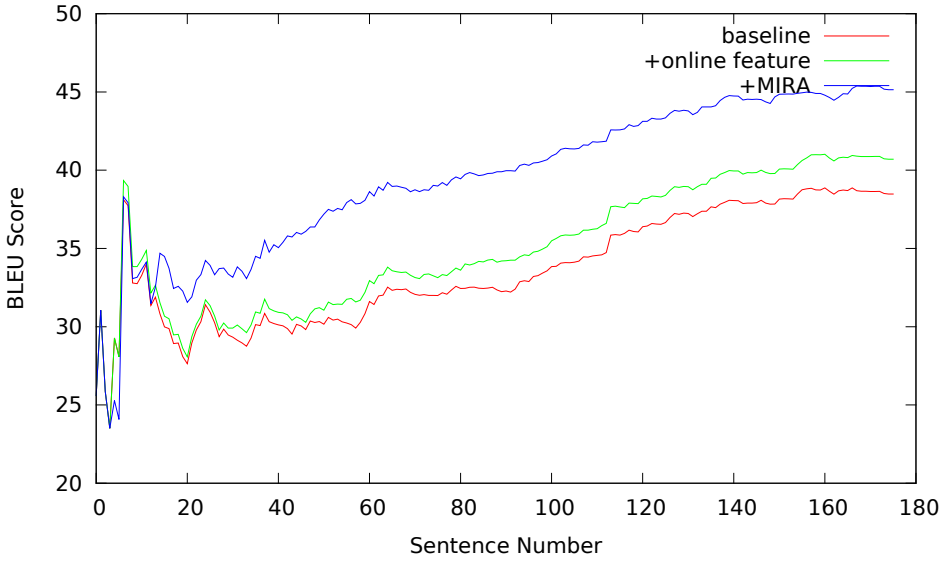

Figure 1: Incremental BLEU vs. evaluation test size on the information-technology task. Three systems are tracked: Baseline, +online feature, +MIRA

System	Bleu (σ)			TER (σ)		
	1 Iter	5 Iter	10 Iter	1 Iter	5 Iter	10 Iter
Baseline	22.18(1.23)	-	-	58.70(1.38)	-	-
+O	22.17(1.19)	21.85(1.25)	21.51(1.23)	58.75(1.35)	59.22(1.36)	60.48(1.35)
+O+NS	21.97(1.20)	22.37(1.20)	22.24(1.22)	58.86(1.37)	58.75(1.37)	59.09(1.40)
+W	22.39(1.23)	21.44(1.20)	21.00(1.13)	58.96(1.40)	58.73(1.34)	58.71(1.28)
+O+W	22.33(1.21)	22.11(1.22)	21.54(1.20)	58.63(1.37)	58.31(1.38)	58.70(1.36)
+O+NS+W	22.34(1.23)	22.09(1.21)	21.62(1.18)	58.60(1.37)	58.48(1.36)	58.40(1.33)
IncGiza Baseline	15.04(1.08)	-	-	72.64(1.34)	-	-
+O	15.30(1.08)	15.47(1.10)	15.86(1.11)	72.33(1.35)	71.68(1.37)	71.09(1.36)
+O+NS	15.21(1.09)	15.48(1.12)	15.48(1.11)	72.19(1.33)	72.06(1.36)	71.65(1.33)
+W	14.81(1.08)	14.61(1.07)	14.73(1.08)	73.03(1.37)	74.69(1.48)	74.28(1.46)
+O+W	15.08(1.08)	15.59(1.09)	**16.42(1.11)**	72.55(1.33)	70.98(1.32)	**70.07(1.27)**
+O+NS+W	15.09(1.08)	15.64(1.08)	**16.15(1.10)**	72.57(1.34)	71.13(1.31)	**70.61(1.33)**

Table 3: Result on the TED talk task (EN>FR). Baseline is a standard phrase based SMT system, +O has the online feature, +NS adds normalization of online feature, +W includes online weight adaptation.

J. Clark, C. Dyer, A. Lavie, and N. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proc. of ACL*, Portland, US-OR.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*, Philadelphia, US-PA.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

M. Federico, N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proc. of Interspeech*, pages 1618–1621, Brisbane, Australia.

M. Federico, A. Cattelan, and M. Trombetti. 2012. Measuring user productivity in machine translation enhanced computer assisted translation. In *Proc. of AMTA*, Bellevue, US-WA.

Q. Gao and S. Vogel. 2008. Parallel implementations of word alignment tool. In *Proc. of SETQA-NLP*, pages 49–57, Columbus, US-OH.

E. Hasler, B. Haddow, and P. Koehn. 2011. Margin infused relaxed algorithm for Moses. *The Prague Bulletin of Mathematical Linguistics*, 96:69–78.

M. Hopkins and J. May. 2011. Tuning as ranking. In *Proc. of EMNLP*, pages 1352–1362, Edinburgh, UK.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proc. of ACL Companion Volume of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

A. Levenberg and M. Osborne. 2009. Stream-based randomised language models for SMT. In *Proc. of EMNLP*, pages 756–764, Singapore.

A. Levenberg, C. Callison-Burch, and M. Osborne. 2010. Stream-based translation models for statistical machine translation. In *Proc. of HLT-NAACL*, Los Angeles, US-CA.

P. Liang and D. Klein. 2009. Online EM for unsupervised models. In *Proc. of NAACL*, pages 611–619, Boulder, US-CO.

P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proc. of ACL*, pages 761–768, Sydney, Australia.

C.-Y. Lin and F. J. Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proc. of COLING*, pages 501–507, Geneva, Switzerland.

A. Lopez. 2008. Tera-scale translation models via pattern matching. In *Proc. of COLING*, pages 505–512, Manchester, UK.

P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta. 2011. Online learning via dynamic reranking for computer assisted translation. In *Proc. of CICLing*, pages 93–105, Tokyo, Japan.

P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recogn.*, 45(9):3193–3203.

R. Neal and G. E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

J. A. Nelder and R. Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pages 160–167, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Research Report RC22176, IBM Research Division, Thomas J. Watson Research Center.

F. Rosenblatt. 1958. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

M.-A. Sato and S. Ishii. 2000. On-line EM algorithm for the normalized Gaussian network. *Neural Comput.*, 12(2):407–432.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA*, Boston, US-MA.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proc. of COLING*, pages 836–841, Copenhagen, Denmark.