Online Multi-User Adaptive Statistical Machine Translation

Prashant Mathur prashant@fbk.eu
FBK Trento, Italy
DISI, University of Trento, Italy

Mauro Cettolo cettolo@fbk.eu
FBK Trento, Italy

Marcello Federico federico@fbk.eu
FBK Trento, Italy

José G. C. de Souza desouza@fbk.eu
FBK Trento, Italy
DISI, University of Trento, Italy

Abstract

In this paper we investigate the problem of adapting a machine translation system to the feedback provided by multiple post-editors. It is well know that translators might have very different post-editing styles and that this variability hinders the application of online learning methods, which indeed assume a homogeneous source of adaptation data. We hence propose *multi-task learning* to leverage bias information from each single post-editors in order to constrain the evolution of the SMT system. A new framework for significance testing with sentence level metrics is described which shows that Multi-Task learning approaches outperforms existing online learning approaches, with significant gains of 1.24 and 1.88 TER score over a strong online adaptive baseline, on a test set of post-edits produced by four translators texts and on a popular benchmark with multiple references, respectively.

1 Introduction

In a professional localization environment, a document is post-edited by several professional translators with assistance of tools such as translation memory, dictionary, spell checkers etc. To speed up the process, lately localization companies have started using computer assisted translation (CAT) tools with statistical machine translation (SMT) systems in the backend. The role played by the SMT engine is to provide a translation hypothesis that the translator can post edit to produce high quality translations (Federico et al., 2012).

In recent works on online adaptation by Mathur et al. (2013) and Denkowski et al. (2014), the SMT is fed with the post edited sentence, allowing the models to adapt to the corrections made by the translators. These kind of systems works well if the document is being post edited by a single translator because models can adapt to the style of that translator. Problems arise when a document is being post edited by a group of translators which is usually the case with big size documents. In fact, if the SMT system adapts to the corrections of all translators together, it will likely mix or overlap stylistic features of the post-editors and thus not learn to mimic well any of them. On the other side, if the system adapts to each individual post-editor, then clearly useful feedback from other post-editors gets wasted.

The main motivation for adapting a SMT system in the backend of CAT tool is that the translation improves over time since fewer mistakes are made after learning from post editions. For example, translator A does not post edit phrase *Fibre Channel* because he thinks that the phrase is a named entity, while B post edits *Fibre Channel* → *Canale a fibre* because he does not recognize it as a named entity. In the backend the SMT system first updates the model such that it keeps the named entity intact but after second post-edition the system adapt the model to translate *Fibre Channel* → *Canale a fibre*. Now, if the system receives again the input *Fibre Channel*, it will prefer to output *Canale a fibre* which will be an incorrect suggestion for translator A. This repetition of translation error slows down the process of post-editing which is completely opposite to the idea of using SMT system in the background.

In this paper, we aim at building a SMT system which can solve this dilemma of contrasting updates. A localization company would expect the SMT system to incorporate these updates and improve the translation quality with time. To do so, we propose using multi-task learning (henceforth MTL) (Caruana, 1993) in machine translation systems. Here, we can consider the translators as different tasks and their post edits as an incoming stream of data the system wants to adapt to. Moreover, this system also maintains a prior relationship between the translators, according to the framework specified in multi-task learning.

The paper is structured as follows. First, we describe previous work on using online learning algorithm in CAT scenario and the generic online multi-task learning algorithm developed by Cavallanti et al. (2010). Then, Section 4 describes the online multi-task learning algorithm which can be applied in CAT scenario. Experiments and results are shown in Section 5. We conclude the paper with a preview of interesting related works and few words about the future work.

2 Background: Online Large Margin Training

Previous work by Mathur et al. (2013) applies an online large margin algorithm (MIRA), that updates the weights w of a phrase-based SMT model according to the loss that is occurred due to an incorrect translation. The margin is coupled with the following loss function based on the complement of the sentence level BLEU (BLEU+1, henceforth sBLEU) (Lin and Och, 2004; Nakov et al., 2012):

$$l_j = sBLEU(y^*) - sBLEU(y_j) \tag{1}$$

where y^* is the $oracle$ (closest translation to the reference) and y_j is the j-th $candidate$ being processed inside an N-best list. According to (Watanabe et al., 2007), weights are updated so that the loss is not larger than the difference between the scores given by the model:

$$l_j \leq w^T \Delta h_j \tag{2}$$

where Δh_j is the difference between the feature vectors of the oracle and the candidate, and w is the weight vector. Hence, the size of the weight update is:

$$\begin{aligned} \arg\min_w ||w - w'|| + C \sum_j \xi_j \\ \text{subject to} \\ w^T \Delta h_j + \xi_j \geq l_j \\ \xi_j \geq 0 \quad \forall j \in \{1 \dots N\} \end{aligned} \tag{3}$$

C is an aggressiveness parameter which controls the size of the update and ξ are slack variables. Following (Watanabe et al., 2007), the Lagrangian dual form of criterion (3) can be derived:

$$\begin{aligned}&\max_{\alpha(\cdot)\geq 0} -\frac{1}{2}||\sum_j \alpha_j \cdot \Delta h_j||^2 + \sum_j \alpha_j l_j - \sum_j \alpha_j w'^T \Delta h_j \\ &\text{subject to} \qquad \sum_j \alpha_j \leq C \end{aligned} \tag{4}$$

which leads to a quadratic programming problem and to the weight vector update:

$$w = w' + \sum_j \alpha_j \cdot \Delta h_j. \tag{5}$$

We determine the lagrangian multipliers α_j at each iteration by applying a QP-solver based on gradient descent.

3 Online Multi-Task Learning

In online multi-task learning (henceforth OMTL) (Cavallanti et al., 2010), training is done jointly on k tasks so as to improve generalization capability for all tasks. Here, the task can be either binary classification or linear regression. The overall goal of OMTL is to learn the k weight vectors simultaneously, one for each task in an online fashion.
The protocol for OMTL at each time t is as follows:

1. receive an input pair (x, s), where x is the example and s is the task id
2. predict the value $\hat{y} = w_s^T h_s(x)$, using current weights w_s for task s
3. receive the correct label y
4. update all the k weight vectors w_s with $s = 1, \ldots, k$

OMTL is a matrix-based regularization approach described in details in Cavallanti et al. (2010). The update step in online learning is the standard Perceptron rule (Rosenblatt, 1958) with different learning rates for each task. These learning rates are defined in an *interaction matrix* which encodes the relatedness among the different tasks. The a_{s_1,s_2} element of the interaction matrix is the learning rate for task s_1 when task s_2 is being executed.

$$A^{-1} = \frac{1}{k+1}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \tag{6}$$

with update rule for weight vector w_s equal to:

$$w_s = w'_s + \hat{y}(A \otimes I_d)_s^{-1} H_s(x) \tag{7}$$

where $\otimes$ denotes the Kronecker product[1] of the interaction matrix (A^{-1}) of dimension $k \times k$ and identity matrix (I_d) of dimensions $d \times d$, making a $kd \times kd$ matrix. The $kd \times kd$ matrix in the update rule co-regularizes the weight vector (w_s) by forcing the learner to account for the relatedness between the tasks. $H_s(x) = (\underbrace{0, \ldots, 0}_{(s-1)d\ times}, h_s(x), \underbrace{0, \ldots, 0}_{(k-s)d\ times}) \in \mathbb{R}^{kd}$ with d being the number of features.

[1] $\otimes$ shows mixed-product property, so one can calculate A^{-1} and then compute the Kronecker product of $A^{-1} \otimes I_d^{-1}$.

4 MIRA with multitasking

MIRA has been successfully applied to tune the log linear weights of SMT model in post editing scenario by Mathur et al. (2013). Here, we extend the online algorithm to fit the same scenario where input comes from k different translators (tasks[2]) and the learner has to predict the weights of all tasks simultaneously.

We modify Equation 5 by adding the matrix co-regularization factor of $(A \otimes I_d)^{-1}$ (from Equation 7), such that the difference of feature vector from j^{th} candidate translation (i.e. Δh_j) affecting the change in weights for current task s take into account the bias from each task. After substitution, our update rule becomes:

$$\begin{aligned} w_s &= w'_s + \sum_j \alpha_j \cdot \langle \Delta h_{s,j} \rangle \qquad \text{where} \\ \langle \Delta h_{s,j} \rangle &= (A \otimes I_d)^{-1} \cdot \Delta H_{s,j} \qquad \text{and} \\ \Delta H_{s,j} &= (\underbrace{0, \ldots, 0}_{(s-1)d \ times}, \Delta h_{s,j}, \underbrace{0, \ldots, 0}_{(k-s)d \ times}) \end{aligned} \tag{8}$$

Here, $\Delta H_{s,j}$ is a compound row vector for candidate translation j of size kd with d being the size of the standard log linear features used in SMT[3]. $(A \otimes I_d)_s^{-1}$ is the co-regularization factor of $kd \times kd$ dimensions. A^{-1} as seen from Equation 6 defines the task relatedness. In CAT scenario we can see the interaction matrix as the matrix which defines relatedness between different translators. This relatedness can be captured by finding a correlation between the translators on their previous post-editions of a given dataset. The similar their post-editions on a particular dataset (with that the machine translation suggestion coming from one SMT system) the more is the relatedness between the translator. In Section 5.2, we show a way to compute the interaction matrix.

5 Experiments and Results

5.1 Data

We evaluated our method on three translation tasks defined over three different domains, namely Information Technology (IT), Travel domain (BTEC) and Legal domain.

The IT test set involves the translation of technical documents from English into Italian and has been used in the field test carried out under the MateCat[4] project. It has been translated by four translators, i.e. four different translations of the source document are available.

BTEC is a publicly available corpus in the travel domain, proposed as translation task in the IWSLT evaluation campaigns up to 2010. In addition to its availability, BTEC is of interest for us because the test set contains six human references, allowing to simulate the multi-task scenario.

Legal domain data has been release as a part of JRC-acquis corpus (Steinberger et al., 2006). The dataset contains translation of legal documents from English to Italian. This dataset was also a part of the field test carried out under the same MateCat project, so essentially we have post-edited data from 4 different translators on a test set of 90 sentences.

Since our methods regard the adaptation of MT models, the potential impact strictly depends on how much the considered text is repetitive. For measuring that text feature, we use the repetition rate proposed by Bertoldi et al. (2013). Equation 9 shows the formula for calculating the repetition rate of a document, where `dict(n)` represents the total number of different

[2] In this paper we use the terms tasks and translators interchangeably as the tasks are translators in the CAT scenario.

[3] To keep the notation light we again drop the dependency of h from x.

[4] http://www.matecat.com

n-grams and n_r is the number of different n-grams occurring exactly $\mathtt{r}$ times. Statistics of the parallel sets on source and target sides along with the repetition rates are reported in Table 1.

$$RR = \left(\prod_{n=1}^{4} \frac{\sum_S dict(n) - n_1}{\sum_S dict(n)} \right)^{1/4} \tag{9}$$

Domain	Set	#srcTok	SrcRR	#tgtTok	TgtRR
IT$_{en \rightarrow it}$	Train	57M	na	60M	na
	Dev	3.3K	19.08	3.6K	18.01
	Test	3K	31.32	3.3K	22.18
BTEC$_{en \rightarrow it}$	Train	0.14M	na	0.13M	na
	Dev	2K	9.47	1.9K	6.73
	Test	1.9K	12.5	1.8K	7.76
Legal$_{en \rightarrow it}$	Train	63M	na	65M	na
	Dev	2.9K	14.37	3.2K	11.25
	Test	2.7K	13.59	2.85K	12.00

Table 1: Statistics of parallel data.

Preparing Data for MTL Since we have k translations for a source document, we shuffle the references/post-editions such that we have one source document and one target document with the sentences containing meta information for the translators who produced these translations. Table 2 shows a sample of source and target document from IT dataset. The figure reads: sentence #1 is translated by translator #0, then feedback (sentence #2) goes to the system with its post-edited translation, system performs multi-task learning and so on. If one removes the meta-information about the translator's ID, the resulting development set is used for online learning (refer Section 2). If one also removes the feedback, then the development set is used for baseline system (refer Section 5.2).

This shuffling of data also impacts the repetition rate. In fact, the repetition rates on the target side of IT test set for each translator varied from 26.95 to 28.70, while the repetition rate on the shuffled target side is 22.18, as reported in Table 1; this could be due to the fact that translators tend to be not consistent among themselves, yielding less repetitions in each post-edited test set than in the shuffled test set.

#Sentence	Sentence	OnlineLearning	Translator ID
1	Input Date_#_0	Not Activated	0
2	Input Date_#_Data di input_#_0	Activated	0
3	Evaluates conditionally_#_1	Not Activated	1
4	Evaluates conditionally_#_Valuta in modo condizionale_#_1	Activated	1

Table 2: Excerpt from IT development set tagged with meta data.

5.2 Experiments

The SMT systems were built using the Moses toolkit (Koehn et al., 2007). Domain specific training data was used to create translation and lexical reordering models. 5-gram language models for each task were estimated by means of IRSTLM toolkit (Federico et al., 2008), with improved Kneser-Ney smoothing (Chen and Goodman, 1998), on the target side of the training parallel corpora. After the training of MT models, the log linear weights were optimized using

MERT (Och, 2003) implementation provided in the Moses toolkit. Performance is computed not with corpus level metrics but with sentence level metrics. We decided to do this to avoid a metric mismatch between the evaluation and actual optimization where the margin is calculated by the sentence level BLEU scores (refer to Section 2). Therefore, we computed sBLEU scores and sentence level TER (Snover et al., 2006) scores and reported their average over the whole documents. We call them avg-sBLEU and avg-sTER.

Calculating A^{-1} matrix: Interaction matrix can be computed in different ways. It basically conveys the relatedness/correlation between the translators who are post-editing a particular document. Usually a localization company keeps a ranking of the hired translators with them; either we can use the ranking to exploit the relatedness between the translators or we can calculate their correlation based on a known previous post-edited data set. Here, we assume that the relatedness between the translators can be seen as the similarity between their post-edited segments given that the MT suggestions were from the same system for all translators. This assumption is quite intuitive.

To compute the similarity, we calculate sentence level TER scores between the MT suggestions and the post-edited segments. In the cases where we do not have post-edited MT suggestions, for example BTEC where only multiple references are available, we simulate the conditions of post-editing by using the SMT translations provided by our own baseline system as MT suggestions. Now, the relatedness can be seen as the correlation between the sentence wise TER scores. We compute the correlation using a widely accepted correlation metric, namely the Pearson correlation coefficient (henceforth r).

Once it is calculated, we rescale these coefficients so that the values are between [0,1], instead of [-1,1] as given by r. We do this rescaling of correlations because matrix-based regularization is not able to handle the negative relatedness between the tasks. These values are computed on the corresponding development sets (which also contain post-edited segments from same translators) and are used to construct the A^{-1} matrix. Since the r is bi-directional, the interaction matrix is symmetric in nature. r values between the translators for IT and BTEC datasets are shown in Tables 3 and 4 respectively.

Translators	**T1**	**T2**	**T3**	**T4**
T1	1	0.82	0.83	0.70
T2	0.82	1	0.86	0.79
T3	0.83	0.86	1	0.77
T4	0.70	0.79	0.77	1

Table 3: Pearson correlation amongst translators on IT dataset.

Translators	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
T1	1	0.69	0.68	0.92	0.96	0.97
T2	0.69	1	0.57	0.64	0.64	0.66
T3	0.68	0.57	1	0.71	0.66	0.67
T4	0.92	0.64	0.71	1	0.90	0.91
T5	0.96	0.64	0.66	0.90	1	0.98
T6	0.97	0.66	0.67	0.91	0.98	1

Table 4: Pearson correlation amongst translators on BTEC dataset. These correlations are computed on a simulated environment.

Now, we give a brief description of the various SMT systems involved in the experiments:

Baseline: SMT models are trained on the domain specific training data; log linear weights are tuned on shuffled development set without any feedback and meta data about translator's ID.

Online: Feedback is added to the development set without the translator's ID. First, log linear weights are tuned on this development data by means of MERT; then, keeping them fixed to the optimal values, additional hyper parameters (used in *Online* system) are tuned again on the development set by means of the Simplex algorithm (Nelder and Mead, 1965). This system contains a single weight vector for all the translators and is the same as explained in (Mathur et al., 2013).

MTL-pearson: Meta-information is added to the development set, and log linear weights are tuned on the dev set. There is an additional bias feature while using multi-task learning which is tuned using Simplex algorithm on the dev set. The elements of interaction matrix are the scaled rs. This system keeps track of k different weight vectors for each translator.

MTL-halfupdate: The diagonal elements of the interaction matrix are set to 1, the off-diagonal elements to 0.5. This means that for every update in the current task $j \in 1 \ldots k$ we do a half-update to rest of the tasks. Note that this system does not need a development set to calculate the interaction matrix unlike MTL-pearson.

K-independent: The interaction matrix is set to be the identity matrix; it means that the tasks are independent of each other because no correlation is assumed between the translators. This system differs from *Online* system because here there is a separate instance of online learning for every translator, while in *Online* system there is a single instance of online learning for all the translators.

5.3 Results

Table 5 shows the avg-sTER[5] and avg-sBLEU scores over whole test set for all the systems. On the IT test set MTL-pearson shows gain of 1 avg-sBLEU points and 3.3 avg-sTER points over the Baseline system and 1.24 avg-sTER points over the strong *Online* system.

However, MTL-pearson does not perform well on BTEC test set, that is we are not able to capture well the task-relatedness in this scenario. Since the actual post-edit translations for BTEC are not available, we simulated them by generating MT suggestions from baseline system, which likely affects the effectiveness of the method. Nevertheless, MTL-halfupdate being a default system is able to capture quite well the correlation between the translators and significantly outperforms all the other systems. We can then conclude that if one does not have access to prior information about the translators for calculating the relatedness amongst them it is a good idea to back-off to use the default half-updates option.

On the Legal domain test set Multi-Task learning is not able to significantly improve over the online learning system. One reason for this could be the total number of sentences in the test set (90), that is each post-editor post edits only 22-25 sentences which is quite less in number as compared to other dataset where total number of sentences are 176 (IT) and 250 (BTEC) and hence each post-editor edits 44 and 42 sentences respectively. The other reason could be the relatively low repetition rate observed on the Legal test set.

System	**IT**		**BTEC**		**Legal**	
	avg-sTER	**avg-sBLEU**	**avg-sTER**	**avg-sBLEU**	**avg-sTER**	**avg-sBLEU**
Baseline	46.91	38.28	42.76	46.69	39.44	41.09
Online	44.86	39.21	42.64	46.72	38.96	41.56
MTL-pearson	**43.62**	**39.27**	41.76	47.17	**38.93**	**41.58**
MTL-halfupdate	44.63	38.94	**40.76**	**47.71**	38.93	41.58
K-independent	46.55	38.04	42.25	47.05	38.93	41.55

Table 5: BLEU scores achieved by using different techniques of online learning. Best BLEU and TER scores are marked in bold fonts.

Significance Testing: Here, we employ a non-parametric multiple hypothesis testing framework such as Friedman tests. The strategy for significance testing is as follows:

[5] It has been shown in the past by Snover et al. (2006) that in post-edit scenario TER has higher correlation than BLEU against the post-editing effort, and so we fix our primary metric to be avg-sTER.

1. We mark epochs at every 10% of test set i.e. t epochs at 10%, 20% .. 100%.

2. At every epoch we measure the average performance of the system in question i.e. calculate avg-sTER.

3. In the end we have avg-sTER scores of five different systems at t different epochs.

4. The average performance of the aforementioned methods on the epochs can be seen as multiple systems trying to solve multiple problems. To calculate the p-values of these multiple systems, we use Friedman test (Friedman, 1937).

5. Once p-values are calculated, we use a post-hoc Holm's procedure (Holm, 1979) to check for the significance.

Results are reported in Table 6.

	P-Value		
Algorithm	IT	BTEC	Legal
MTL-pearson vs. Online	$0.022^\diamond$	$0.028^\diamond$	$0.066^\diamond$
MTL-halfupdate vs. Online	$0.003^\diamond$	$0.000^\diamond$	$0.066^\diamond$
K-Independent vs. Online	0.311	0.479	0.160
MTL-pearson vs. K-Independent	0.200	0.137	0.670
MTL-halfupdate vs. K-Independent	0.050	$0.000^\diamond$	0.670
MTL-pearson vs. MTL-halfupdate	0.500	$0.007^\diamond$	1.000

Table 6: p-values given by Friedman test. $\diamond$ depicts a significant difference between the systems that are being compared.

We plotted the incremental avg-sTER scores over t different epochs on all test sets in Figure 1.

First of all, it is worth to compare the plots of *MTL-pearson* and of *Online* systems on IT test set, for which the improvement of over 3 avg-sTER points reported in Table 5 is significant (Table 6). In fact, the gap between the *MTL-pearson* system and the *Online* system is visible in the plot only after the 6th epoch, that is for 6 out of 10 epochs differences are not big enough; nevertheless; the difference is significant. *MTL-halfupdate* performs better than any other system at least on 6 out of 10 epochs, but even on all epochs with respect to the *Online* system: this is why it outperforms the *Online* at 95% of confidence interval. Interesting to note that MTL-halfupdate is the best performing system till 6 epochs; after that, MTL-pearson becomes the best one: this basically says that for the starting 60% of the data the translators had a correlation of half with each other, while later they were as coherent as they were when they post-edited the development set (because MTL-pearson correlation is calculated on development set). This also means that the relatedness between the translators is evolving even while post-editing the same dataset.

On BTEC test set, MTL-halfupdate consistently outperforms all other SMT systems on each epoch; this explains why it is significantly better than all other systems. On 9 epochs out of 10, MTL-pearson is better than the Online system; hence, the difference is significant. Results on BTEC put in evidence the importance of estimating a reliable interaction matrix to allow multi-task learning working at its best, but also that half-update is an effective back-off solution.

Significance tests on Legal test set[6] shows that MTL-* systems are better than the Online

[6] The error curve in Legal domain shows an apparently surprising increasing trend. This is due to the nature of the test set where the starting sentences are easier to translate than the later ones.

system with a p-value of 0.066.

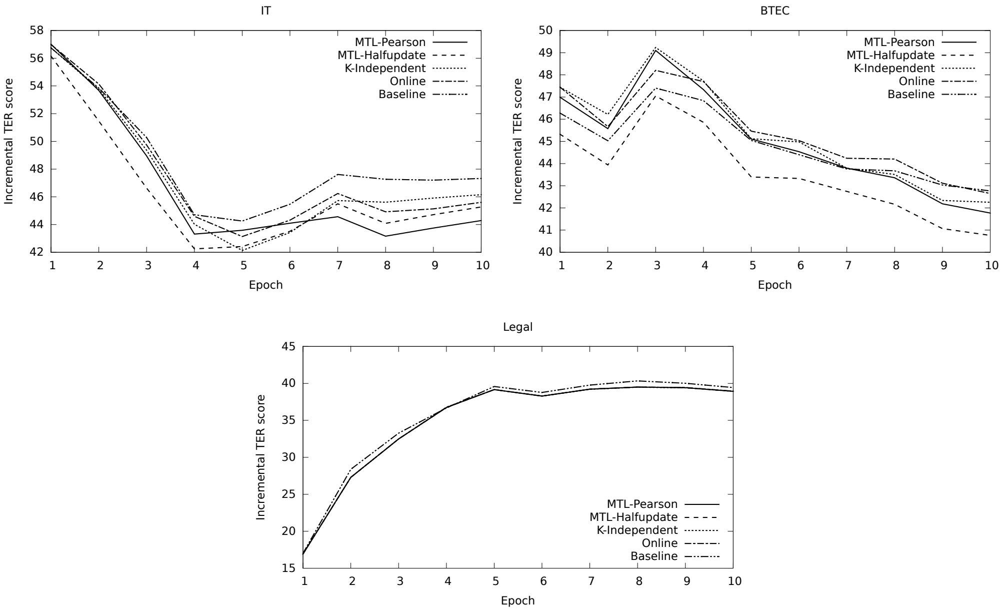

Figure 1: Learning curve of different systems on IT (top left), BTEC (top right) and Legal (bottom) test sets.

So far, the evaluations were done on a shuffle of test set where the translators were assigned in sequence, i.e. first sentence to first translator, second to second and so on. This is usually not the case in a real world scenario, because a sentence can be assigned to any of the translators and not necessarily in a sequence. To replicate such scenario, we developed an assigning scheme through which each translator is assigned equal number of segments from a document to post-edit. The scheme is as follows:

1. For n translators, all possible permutations of the series $1...n$ is computed (total of $n!$).
2. The document to be post edited is divided in blocks of n sentences.
3. For each block we randomly pick a permutation series among the $n!$ choices, and assign it to the block in question.

Following this scheme, we created 100 different shuffles of the IT test set which are closer to the real life setting. Similar to the learning curve we built before, we averaged out avg-sTER scores over 100 shuffles on sequential epochs i.e. (10%, 20%...100% of data). Figure 2 reports the learning curves of different adaptive systems over epochal data.

Here, unlike in the previous case, for each of 176 sentences we have 100 different sentence wise TER scores using 5 different systems. Since just the IT domain is considered, data are more homogeneous and then we could apply *Approximate Randomization* (Noreen, 1989), a statistical test that is well established in the NLP community (Chinchor et al., 1993). The test has been shown (Riezler and Maxwell, 2005) to be less prone to type-I errors than the boostrap method (Efron and Tibshirani, 1993). We report the significance results in Table 7.

Even after the shuffling, we see that MTL-pearson system resistant to the shuffles and still performs significantly better than any other system. However, we observe a contradictory infor-

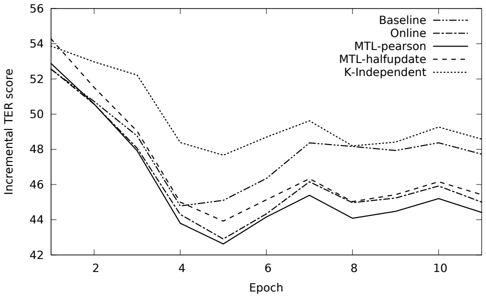

Figure 2: Learning curves of different systems on *shuffled* IT test set.

Systems Compared	P-Value
Baseline vs. Online	0.001
Baseline vs. MTL-pearson	0.001
MTL-pearson vs. K-Independent	0.001
MTL-HalfUpdate vs. Online	0.04
MTL-pearson vs. MTL-HalfUpdate	0.001
Online vs. MTL-pearson	0.007

Table 7: p-values given by Approximate Randomization test. All the reported results in the table are significant.

mation from the previous results; MTL-HalfUpdate system performs significantly worse than Online system over 100 shuffles, which means that quality of translation from MTL-HalfUpdate system can degrade if the translators are randomly assigned and not sequential as in the previous case. The same behaviour is observed in K-independent system where the system's performance is significantly worse than the Baseline system. All the other results remain consistent to what we observed in the previous case.

Table 8 shows an excerpt from the IT test set. The phrase *backup* in the source sentence (#21) is translated to *copia di riserva* by both K-Independent and MTL-pearson systems but the translator post-edits the phrase in both translation hypotheses to *backup*. Later, in sentence #23 the phrase appears again and this time Multi-Task correctly outputs the translation of the phrase *backup* to *backup* but K-independent system is not able to correct the mistake. Reiterating, K-Independent system runs a single instance of online learning for each of the post-editors. In the example the first sentence is post-edited by translator #3 and the latter by translator#1, thus, the system is not able to recognize the mistake committed for the translator #1 and consequently cannot correct it for translator #3. While the system MTL-pearson learns jointly over the corrections by all the translators and thus able to correct the translation hypothesis the next time.

Overall, the results show that Multi-Task learning outperforms the existing standard SMT and the strong online learning systems. If we have the meta information on the post-editors apriori i.e. their mutual correlation, we can boost the performance of the adaptive system. One can use the MTL-pearson system if the correlation matrix can be calculated accurately; if not, it is preferable to back-off to MTL-halfupdate system.

Source - 21	with minimal copying of data from the production volume to **backup** volume ._#_3
K-Ind - 21	con un minimo la copia di dati dal volume di produzione per il volume della **copia di riserva** .
Multi-Task - 21	con un minimo la copia di dati dal volume di produzione per il volume della **copia di riserva** .
Post-Edit - 21	con una copia minima dei dati dal volume di produzione nel volume di **backup** .
Source - 23	you create a **backup** and after it completes _#_1
K-Ind - 23	possibile creare una **copia di riserva** e dopo il completamento
Multi-Task - 23	creare un **backup** e dopo il completamento
Post-Edit - 23	possibile creare un **backup** e , al suo completamento

Table 8: Example from the IT test set. Here *Multi-Task* refers to MTL-pearson system and *K-ind* is K-independent system.

6 Related Works

Despite 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).

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 feature weights and features itself.

Multi-Task learning has been explored in SMT in the context of tuning the sparse log linear weights by Simianer et al. (2012) where they split the training set in random shards and perform a joint feature selection over these shards using ℓ_1/ℓ_2 regularization. In this way after each epoch, the size of feature vector decreases and only the important features are taken into account. In our paper instead of ℓ_1/ℓ_2 regularization we have use a matrix-based regularization approach on the core features for online adaptation of all the translation models.

Multi-Task learning has also been used in re-ranking the N-best list by Duh et al. (2010). Each N-Best list is considered as a different task and the weights are jointly learnt over a large set of sparse features. Simianer et al. (2011) trained a discriminative model using multi-task learning over a set of k documents belonging to different topics but with strong commonalities.

Recent application of multi-task learning has been in quality estimation for machine translation by Cohn and Specia (2013) where the authors model annotator bias using multi-task Gaussian processes. Their model outperforms the annotator specific model and thus boosting the use of Multi-Task learning in NLP applications. Another application of MTL has been in supervised domain adaptation for quality estimation (C. de Souza et al., 2014). In this work the authors leverage all available training labels from different domains in order to learn a robust model for a target domain with very little labeled data. The approach proposed outperforms independent models trained separetely on each domain.

7 Conclusion

We addressed the problem of adapting in a CAT framework a single SMT system to multiple post-editions, i.e. to an incoming stream of feedback from different translators. In such a situation, standard online learning methods can lead to incoherent translations by the SMT system. To the best of our knowledge, this kind of problem has never been addressed before for adapting SMT systems in CAT scenario. As a solution we propose to adopt a multi-task learning scheme, which relies on the correlation amongst the translators computed using prior knowledge; the online learner is then constrained to take into account the relatedness amongst the translators.

Different online systems have been compared against each other, and online multi-task learning SMT system outperformed in most cases the strong online learning SMT system taken as baseline. Whenever not enough information about the correlation amongst the translators is available, our experimental outcomes suggest to use multi-task learning with half-updates, which is a good generalization of the interaction between the translators. We also compared the Multi-Task approach to the K-Independent system where each translator has been alloted an online learning SMT system; evidently, multi-task also fared better against this system setup. Moreover, MTL can also be applied to tune the log-linear weights of SMT models when multiple references are given.

In our approach, once the correlation matrix has been computed, it is kept fixed throughout the learning process. Instead, as evinced by our experiments, the interaction between translators can evolve over time; we plan to further investigate this aspect in the future.

Acknowledgements

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

References

Bertoldi, N., Cettolo, M., and Federico, M. (2013). Cache-based Online Adaptation for Machine Translation Enhanced Computer Assisted Translation. In *Proceedings of the MT Summit XIV*, pages 35–42, Nice, France.

C. de Souza, J. G., Turchi, M., and Negri, M. (2014). Machine translation quality estimation across domains. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 409–420, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann.

Cavallanti, G., Cesa-Bianchi, N., and Gentile, C. (2010). Linear algorithms for online multitask classification. *J. Mach. Learn. Res.*, 11:2901–2934.

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

Chen, S. F. and Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University.

Chinchor, N., Hirschman, L., and Lewis, D. D. (1993). Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3). *Computational Linguistics*, 19(3):409–449.

Cohn, T. and Specia, L. (2013). Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *ACL (1)*, pages 32–42. The Association for Computer Linguistics.

Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.

Denkowski, M., Dyer, C., and Lavie, A. (2014). Learning from post-editing: Online model adaptation for statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 395–404, Gothenburg, Sweden. Association for Computational Linguistics.

Duh, K., Sudoh, K., Tsukada, H., Isozaki, H., and Nagata, M. (2010). N-best reranking by multitask learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 375–383, Stroudsburg, PA, USA. Association for Computational Linguistics.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, pages 1618–1621, Brisbane, Australia.

Federico, M., Cattelan, A., and Trombetti, M. (2012). Measuring user productivity in machine translation enhanced computer assisted translation. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

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

Martínez-Gómez, P., Sanchis-Trilles, G., and Casacuberta, F. (2011). Online learning via dynamic reranking for computer assisted translation. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part II*, CICLing'11, pages 93–105, Berlin, Heidelberg. Springer-Verlag.

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

Mathur, P., Cettolo, M., and Federico, M. (2013). Online Learning Approaches in Computer Assisted Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 301–308, Sofia, Bulgaria. Association for Computational Linguistics.

Nakov, P., Guzmán, F., and Vogel, S. (2012). Optimizing for sentence-level bleu+1 yields short translations. In *COLING*, pages 1979–1994.

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

Noreen, E. W. (1989). *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley Interscience.

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

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In Hinrichs, E. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Riezler, S. and Maxwell, J. T. (2005). On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

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

Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*.

Simianer, P., Wschle, K., and Riezler, S. (2011). Multi-task minimum error rate training for smt. *Prague Bull. Math. Linguistics*, 96:99–108.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *5th Conference of the Association for Machine Translation in the Americas (AMTA)*, Boston, Massachusetts.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., and Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 2142–2147, Genoa, Italy.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 764–773.