Online Multitask Learning for Machine Translation Quality Estimation

José G. C. de Souza[1,2], **Matteo Negri**[1], **Elisa Ricci**[1], **Marco Turchi**[1]
[1] FBK - Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
[2] University of Trento, Italy
{desouza,negri,eliricci,turchi}@fbk.eu

Abstract

We present a method for predicting machine translation output quality geared to the needs of computer-assisted translation. These include the capability to: *i)* continuously learn and self-adapt to a stream of data coming from multiple translation jobs, *ii)* react to data diversity by exploiting human feedback, and *iii)* leverage data similarity by learning and transferring knowledge across domains. To achieve these goals, we combine two supervised machine learning paradigms, online and multitask learning, adapting and unifying them in a single framework. We show the effectiveness of our approach in a regression task (HTER prediction), in which online multitask learning outperforms the competitive online single-task and pooling methods used for comparison. This indicates the feasibility of integrating in a CAT tool a single QE component capable to simultaneously serve (and continuously learn from) multiple translation jobs involving different domains and users.

1 Introduction

Even if not perfect, machine translation (MT) is now getting reliable enough to support and speed-up human translation. Thanks to this progress, the work of professional translators is gradually shifting from full translation from scratch to MT post-editing. Advanced computer-assisted translation (CAT) tools[1] provide a natural framework for this activity by proposing, for each segment in a source document, one or more suggestions obtained either from a translation memory (TM) or from an MT engine. In both cases, accurate mechanisms to indicate the reliability of a suggestion are extremely useful to let the user decide whether to post-edit a given suggestion or ignore it and translate the source segment from scratch. However, while scoring TM matches relies on standard methods based on fuzzy matching, predicting the quality of MT suggestions at run-time and without references is still an open issue.

This is the goal of MT quality estimation (QE), which aims to predict the quality of an automatic translation as a function of the estimated number of editing operations or the time required for manual correction (Specia et al., 2009; Soricut and Echihabi, 2010; Bach et al., 2011; Mehdad et al., 2012). So far, QE has been mainly approached in controlled settings where homogeneous training and test data is used to learn and evaluate static predictors. Cast in this way, however, it does not fully reflect (nor exploit) the working conditions posed by the CAT framework, in which:

1. The QE module is exposed to a continuous stream of data. The amount of such data and the tight schedule of multiple, simultaneous translation jobs prevents from (theoretically feasible but impractical) complete re-training procedures in a batch fashion and advocate for continuous learning methods.

2. The input data can be diverse in nature. Continuous learning should be sensitive to such differences, in a way that each translation job and user is supported by a reactive model that is robust to variable working conditions.

3. The input data can show similarities with previous observations. Continuous learning should leverage such similarities, so that QE can capitalize from all the previously processed segments even if they come from different domains, genres or users.

While previous QE research disregarded these challenges or addressed them in isolation, our

[1] See for instance the open source MateCat tool (Federico et al., 2014).

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 219–228, Beijing, China, July 26-31, 2015.

work tackles them in a single unifying framework based on the combination of two paradigms: *online* and *multitask* learning. The former provides continuous learning capabilities that allow the QE model to be robust and self-adapt to a stream of potentially diverse data. The latter provides the model with the capability to exploit the similarities between data coming from different sources. Along this direction our contributions are:

- The first application of online multitask learning to QE, geared to the challenges posed by CAT technology. In this framework, our models are trained to predict MT quality in terms of HTER (Snover et al., 2006).[2]
- The extension of current online multitask learning methods to regression. Prior works in the machine learning field applied this paradigm to classification problems, but its use for HTER estimation requires real-valued predictions. To this aim, we propose a new regression algorithm that, at the same time, handles positive and negative transfer and performs online weight updates.
- A comparison between online multitask and alternative, state-of-the-art online learning strategies. Our experiments, carried out in a realistic scenario involving a stream of data from four domains, lead to consistent results that prove the effectiveness of our approach.

2 Related Work

In recent years, sentence-level QE has been mainly investigated in controlled evaluation scenarios such as those proposed by the shared tasks organized within the WMT workshop on SMT (Callison-Burch et al., 2012; Bojar et al., 2013; Bojar et al., 2014). In this framework, systems trained from a collection of (*source*, *target*, *label*) instances are evaluated based on their capability to predict the correct label[3] for new, unseen test items. Compared to our application scenario, the shared tasks setting differs in two main aspects. First, the data used are substantially homogeneous (usually they come from the same domain, and target translations are produced by the same MT system). Second, training and test are carried out as distinct, sequential phases. Instead, in the CAT environment, a QE component should ideally serve, adapt to and continuously learn from simultaneous translation jobs involving different MT engines, domains, genres and users (Turchi et al., 2013).

These challenges have been separately addressed from different perspectives in few recent works. Huang et al. (2014) proposed a method to adaptively train a QE model for document-specific MT post-editing. Adaptability, however, is achieved in a batch fashion, by re-training an *ad hoc* QE component for each document to be translated. The adaptive approach proposed by Turchi et al. (2014) overcomes the limitations of batch methods by applying an online learning protocol to continuously learn from a stream of (potentially heterogeneous) data. Experimental results suggest the effectiveness of online learning as a way to exploit user feedback to tailor QE predictions to their quality standards and to cope with the heterogeneity of data coming from different domains. However, though robust to user and domain changes, the method is solely driven by the distance computed between predicted and true labels, and it does not exploit any notion of similarity between tasks (*e.g.* domains, users, MT engines).

On the other way round, task relatedness is successfully exploited by Cohn and Specia (2013), who apply multitask learning to jointly learn from data obtained from several annotators with different levels of expertise and reliability. A similar approach is adopted by de Souza et al. (2014a), who apply multitask learning to cope with situations in which a QE model has to be trained with scarce data from multiple domains/genres, different from the actual test domain. The two methods significantly outperform both individual single-task (in-domain) models and single pooled models. However, operating in batch learning mode, none of them provides the continuous learning capabilities desirable in the CAT framework.

The idea that online and multitask learning can complement each other if combined is suggested by (de Souza et al., 2014b), who compared the two learning paradigms in the same experimental setting. So far, however, empirical evidence of this complementarity is still lacking.

[2]The HTER is the minimum edit distance between a translation suggestion and its manually post-edited version in the [0,1] interval. Edit distance is calculated as the number of edits (word insertions, deletions, substitutions, and shifts) divided by the number of words in the reference.

[3]Possible label types include *post-editing effort* scores (*e.g.* 1-5 Likert scores indicating the estimated percentage of MT output that has to be corrected), *HTER* values, and post-editing *time* (*e.g.* seconds per word).

3 Online Multitask Learning for QE

Online learning takes place in a stepwise fashion. At each step, the learner processes an instance (in our case a feature vector extracted from source and target sentences) and predicts a label for it (in our case an HTER value). After the prediction, the learner receives the "true" label (in our case the actual HTER computed from a human post-edition) and computes a *loss* that indicates the distance between the predicted and the true label. Before going to the next step, the weights are updated according to the suffered loss.

Multitask learning (MTL) aims to simultaneously learn models for a set of possibly related tasks by exploiting their relationships. By doing this, improved generalization capabilities are obtained over models trained on the different tasks in isolation (single-task learning – STL). The relationships among tasks are provided by a shared structure, which can encode three types of relationships based on their correlation (Zhang and Yeung, 2010). Positive correlation indicates that the tasks are related and knowledge transfer should lead to similar model parameters. Negative correlation indicates that the tasks are likely to be unrelated and knowledge transfer should force an increase in the distance between model parameters. No correlation indicates that the tasks are independent and no knowledge transfer should take place. In our case, a task is a set of (instance, label) pairs obtained from source sentences coming from different translation jobs, together with their translations produced by several MT systems and the relative post-editions from various translators. In this paper the terms task and domain are used interchangeably.

Early MTL methods model only positive correlation (Caruana, 1997; Argyriou et al., 2008), which results in a positive knowledge transfer between all the tasks, with the risk of impairing each other's performance when they are unrelated or negatively correlated. Other methods (Jacob et al., 2009; Zhong and Kwok, 2012; Yan et al., 2014) cluster tasks into different groups and share knowledge only among those in the same cluster, thus implicitly identifying outlier tasks. A third class of algorithms considers all the three types of relationships by learning task interaction via the covariance of task-specific weights (Bonilla et al., 2008; Zhang and Yeung, 2010). All these methods, however, learn the task relationships in batch mode. To overcome this limitation, recent works propose the "lifelong learning" paradigm (Eaton and Ruvolo, 2013; Ruvolo and Eaton, 2014), in which all the instances of a task are given to the learner sequentially and the previously learned tasks are leveraged to improve generalization for future tasks. This approach, however, is not applicable to our scenario as it assumes that all the instances of each task are processed as separate blocks.

In this paper we propose a novel MTL algorithm for QE that learns the structure shared by different tasks in an online fashion and from an input stream of instances from all the tasks. To this aim, we extend the online passive aggressive (PA) algorithm (Crammer et al., 2006) to the multitask scenario, learning a set of task-specific regression models. The multitask component of our method is given by an "interaction matrix" that defines to which extent each encoded task can "borrow" and "lend" knowledge from and to the other tasks. Opposite to previous methods (Cavallanti et al., 2010) that assume fixed dependencies among tasks, we propose to learn the interaction matrix instance-by-instance from the data. To this aim we follow the recent work of Saha et al. (2011), extending it to a regression setting. The choice of PA is motivated by practical reasons. Indeed, by providing the best trade-off between accuracy and computational time (He and Wang, 2012) compared to other algorithms such as OnlineSVR (Parrella, 2007), it represents a good solution to meet the demand of efficiency posed by the CAT framework.

3.1 Passive Aggressive Algorithm

PA follows the typical online learning protocol. At each round t the learner receives an instance, $\mathbf{x}_t \in \mathbb{R}^d$ (d is the number of features), and predicts the label $\hat{y}_t$ according to a function parametrized by a set weights $\mathbf{w}_t \in \mathbb{R}^d$. Next, the learner receives the true label y_t, computes the ϵ-insensitive loss, ℓ_ϵ, measuring the deviation between the prediction $\hat{y}_t$ and the true label y_t and updates the weights. The weights are updated by solving the optimization problem:

$$\begin{aligned} \mathbf{w}_t = \arg\min_{\mathbf{w}} \quad & \mathcal{C}_{PA}(\mathbf{w}) + C\xi \\ \text{s.t.} \quad & \ell_\epsilon(\mathbf{w}, (\mathbf{x}_t, y_t)) \leq \xi \text{ and } \xi \geq 0 \end{aligned} \tag{1}$$

where $\mathcal{C}_{PA}(\mathbf{w}) = \frac{1}{2}||\mathbf{w} - \mathbf{w}_{t-1}||^2$ and ℓ_ϵ is the ϵ-insensitive hinge loss defined as:

$$\ell_\epsilon(\mathbf{w}, (\mathbf{x}, y)) = \begin{cases} 0, & \text{if } |y - \mathbf{w} \cdot \mathbf{x}| \leq \epsilon \\ |y - \mathbf{w} \cdot \mathbf{x}| - \epsilon, & \text{otherwise} \end{cases} \quad (2)$$

The loss is zero when the absolute difference between the prediction and the true label is smaller or equal to ϵ, and grows linearly with this difference otherwise. The ϵ parameter is given as input and regulates the sensitivity to mistakes. The slack variable ξ acts as an upper-bound to the loss, while the C parameter is introduced to control the aggressiveness of the weights update. High C values lead to more aggressive weight updates. However, when the labels present some degree of noise (a common situation in MT QE), they might cause the learner to drastically change the weight vector in a wrong direction. In these situations, setting C to small values is desirable. As shown in (Crammer et al., 2006), a closed form solution for the weights update in Eq.1 can be derived as:

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \text{sgn}(y_t - \hat{y}_t)\tau_t \mathbf{x}_t \quad (3)$$

with $\tau_t = \min(C, \frac{\ell_t}{||\mathbf{x}_t||^2})$ and $\ell_t = \ell_\epsilon(\mathbf{w}, (\mathbf{x}_t, y_t))$.

3.2 Passive Aggressive MTL Algorithm

Our Passive Aggressive Multitask Learning (PAMTL) algorithm extends the traditional PA for regression to multitask learning. Our approach is inspired by the Online Task Relationship Learning algorithm proposed by Saha et al. (2011) which, however, is only defined for classification.

The learning process considers one instance at each round t. The random sequence of instances belongs to a fixed set of K tasks and the goal of the algorithm is to learn K linear models, one for each task, parametrized by weight vectors $\widetilde{\mathbf{w}}_{t,k}$, $k \in \{1, \ldots, K\}$. Moreover, the algorithm also learns a positive semidefinite matrix $\mathbf{\Omega} \in \mathbb{R}^{K \times K}$, modeling the relationship among tasks. Algorithm 1 summarizes our approach. At each round t, the learner receives a pair $(\mathbf{x}_t, i_t)$ where $\mathbf{x}_t \in \mathbb{R}^d$ is an instance and $i_t \in \{1, \ldots, K\}$ is the task identifier. Each incoming instance is transformed to a compound vector $\boldsymbol{\phi}_t = [0, \ldots, 0, \mathbf{x}_t, 0, \ldots, 0] \in \mathbb{R}^{Kd}$. Then, the algorithm predicts the HTER score corresponding to the label $\hat{y}$ by using the weight vector $\widetilde{\mathbf{w}}_t$. The weight vector is a compound vector $\widetilde{\mathbf{w}}_t = [\widetilde{\mathbf{w}}_{t,1}, \ldots, \widetilde{\mathbf{w}}_{t,K}] \in \mathbb{R}^{Kd}$, where $\widetilde{\mathbf{w}}_{t,k} \in \mathbb{R}^d, k \in \{1, \ldots, K\}$. Next, the learner receives the true HTER label y and computes the loss ℓ_ϵ (Eq. 2) for round t.

Algorithm 1 PA Multitask Learning (PAMTL)

Input: instances from K tasks, number of rounds $R > 0$, $\epsilon > 0$, $C > 0$
Output: $\mathbf{w}$ and $\mathbf{\Omega}$, learned after T rounds

Initialization: $\mathbf{\Omega} = \frac{1}{K} \times \mathbf{I}_k$, $\mathbf{w} = \mathbf{0}$
for t = 1 to T **do**
  receive instance $(\mathbf{x}_t, i_t)$
  compute $\boldsymbol{\phi}_t$ from $\mathbf{x}_t$
  predict HTER $\hat{y}_t = (\widetilde{\mathbf{w}}_t^T \cdot \boldsymbol{\phi}_t)$
  receive true HTER label y_t
  compute ℓ_t (Eq. 2)
  compute $\tau_t = \min(C, \frac{\ell_t}{||\boldsymbol{\phi}_t||^2})$
  `/* update weights */`
  $\widetilde{\mathbf{w}}_t = \widetilde{\mathbf{w}}_{t-1} + \text{sgn}(y_t - \hat{y}_t)\tau_t(\mathbf{\Omega}_{t-1} \otimes \mathbf{I}_d)^{-1}\boldsymbol{\phi}_t$
  `/* update task matrix */`
  if $t > R$ **then**
    update $\mathbf{\Omega}_t$ with Eq. 6 or Eq. 7
  end if
end for

We propose to update the weights by solving:

$$\widetilde{\mathbf{w}}_t, \mathbf{\Omega}_t = \underset{\mathbf{w}, \mathbf{\Omega} \succ 0}{\text{argmin}} \quad \mathcal{C}_{MTL}(\mathbf{w}, \mathbf{\Omega}) + C\xi + \mathcal{D}(\mathbf{\Omega}, \mathbf{\Omega}_{t-1})$$
$$\text{s.t.} \quad \ell_\epsilon(\mathbf{w}, (\mathbf{x}_t, y_t)) \leq \xi, \xi \geq 0 \quad (4)$$

The first term models the joint dependencies between the task weights and the interaction matrix and it is defined as $\mathcal{C}_{MTL}(\mathbf{w}, \mathbf{\Omega}) = \frac{1}{2}(\mathbf{w} - \widetilde{\mathbf{w}}_t)^T \mathbf{\Omega}_\otimes (\mathbf{w} - \widetilde{\mathbf{w}}_t)$, where $\mathbf{\Omega}_\otimes = \mathbf{\Omega} \otimes \mathbf{I}_d$. The function $\mathcal{D}(\cdot)$ represents the divergence between a pair of positive definite matrices. Similar to (Saha et al., 2011), to define $\mathcal{D}(\cdot)$ we also consider the family of Bregman divergences and specifically the LogDet and the Von Neumann divergences. Given two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$, the LogDet divergence is $\mathcal{D}_{LD}(\mathbf{X}, \mathbf{Y}) = \text{tr}(\mathbf{X}\mathbf{Y}^{-1}) - \log|\mathbf{X}\mathbf{Y}^{-1}| - n$, while the Von Neumann divergence is computed as $\mathcal{D}_{VN}(\mathbf{X}, \mathbf{Y}) = \text{tr}(\mathbf{X}\log\mathbf{X} - \mathbf{Y}\log\mathbf{Y} - \mathbf{X} + \mathbf{Y})$.

The optimization process to solve Eq.4 is performed with an alternate scheme: first, with a fixed $\mathbf{\Omega}$, we compute $\mathbf{w}$; then, given $\mathbf{w}$ we optimize for $\mathbf{\Omega}$. The closed-form solution for updating $\mathbf{w}$, which we derived similarly to the PA update (Crammer et al., 2006), becomes:

$$\widetilde{\mathbf{w}}_t = \widetilde{\mathbf{w}}_{t-1} + \text{sgn}(y_t - \hat{y}_t)\tau_t(\mathbf{\Omega}_{t-1} \otimes \mathbf{I}_d)^{-1}\boldsymbol{\phi}_t \quad (5)$$

In practice, the interaction matrix works as a learning rate when updating the weights of each task. Similarly, following previous works (Tsuda et al., 2005), the update steps for the interaction matrix $\mathbf{\Omega}$ can be easily derived. For the Log-Det divergence we have:

$$\mathbf{\Omega}_t = (\mathbf{\Omega}_{t-1} + \eta\, \text{sym}(\widetilde{\mathbf{W}}_{t-1}^T \widetilde{\mathbf{W}}_{t-1}))^{-1} \quad (6)$$

while for the Von Neumann we obtain:

$$\mathbf{\Omega}_t = \exp(\log \mathbf{\Omega}_{t-1} - \eta \, \mathrm{sym}(\widetilde{\mathbf{W}}_{t-1}^T \widetilde{\mathbf{W}}_{t-1})) \quad (7)$$

where $\widetilde{\mathbf{W}}_t \in \mathbb{R}^{d \times K}$ is a matrix obtained by column-wise reshaping the weight vector $\widetilde{\mathbf{w}}_t$, $\mathrm{sym}(\mathbf{X}) = (\mathbf{X} + \mathbf{X}^T)/2$ and η is the learning rate parameter. The sequence of steps to compute $\mathbf{\Omega}_t$ and $\widetilde{\mathbf{w}}_t$ is summarized in Algorithm 1. Importantly, the weight vector is updated at each round t, while $\mathbf{\Omega}_t$ is initialized to a diagonal matrix and it is only computed after R iterations. In this way, at the beginning, the tasks are assumed to be independent and the task-specific regression models are learned in isolation. Then, after R rounds, the interaction matrix is updated and the weights are refined considering tasks dependencies. This leads to a progressive increase in the correlation of weight vectors of related tasks. In the following, PAMTL$_{vn}$ refers to PAMTL with the Von Neumann updates and PAMTL$_{ld}$ to PAMTL with LogDet updates.

4 Experimental Setting

In this section, we describe the data used in our experiments, the features extracted from the source and target sentences, the evaluation metric and the baselines used for comparison.

Data. We experiment with English-French datasets coming from Technology Entertainment Design talks (TED), Information Technology manuals (IT) and Education Material (EM). All datasets provide a set of tuples composed by (source, translation and post-edited translation).

The TED dataset is distributed in the Trace corpus[4] and includes, as source sentences, the subtitles of several talks spanning a range of topics presented in the TED conferences. Translations were generated by two different MT systems: a phrase-based statistical MT system and a commercial rule-based system. Post-editions were collected from four different translators, as described by Wisniewski et al. (2013).

The IT manuals data come from two language service providers, henceforth $LSP1$ and $LSP2$. The IT_{LSP1} tuples belong to a software manual translated by an SMT system trained using the Moses toolkit (Koehn et al., 2007). The post-editions were produced by one professional translator. The IT_{LSP2} data includes a software manual from the automotive industry; its source sentences are translated with an adaptive proprietary MT system and post-edited by several professional translators. The EM corpus is also provided by $LSP2$ and regards educational material (*e.g.* courseware and assessments) of various text styles. The translations and post-editions are produced in the same way as for IT_{LSP2}. The IT_{LSP2} and the EM datasets are derived from the Autodesk Post-Editing Data corpus.[5]

Domain	No. tokens	Vocab. Size	Avg. Snt. Length
TED src	20,048	3,452	20
TED tgt	21,565	3,940	22
IT$_{LSP1}$ src	12,791	2,013	13
IT$_{LSP1}$ tgt	13,626	2,321	13
EM src	15,327	3,200	15
EM tgt	17,857	3,149	17
IT$_{LSP2}$ src	15,128	2,105	13
IT$_{LSP2}$ tgt	17,109	2,104	14

Table 1: Data statistics for each domain.

In total, we end up with four domains (TED, IT$_{LSP1}$, EM and IT$_{LSP2}$), which allows us to evaluate the PAMTL algorithm in realistic conditions where the QE component is exposed to a continuous stream of heterogeneous data. Each domain is composed by 1,000 tuples formed by: *i)* the English source sentence, *ii)* its automatic translation in French, and *iii)* a real-valued quality label obtained by computing the HTER between the translation and the post-edition with the TERCpp open source tool.[6]

Table 1 reports some macro-indicators (number of tokens, vocabulary size, average sentence length) that give an idea about the similarities and differences between domains. Although they contain data from different software manuals, similar vocabulary size and sentence lengths for the two IT domains seem to reflect some commonalities in their technical style and jargon. Larger values for TED and EM evidence a higher lexical variability in the topics that compose these domains and the expected stylistic differences featured by speech transcriptions and non-technical writing. Overall, these numbers suggest a possible dissimilar-

[4] http://anrtrace.limsi.fr/trace_postedit.tar.bz2

[5] https://autodesk.app.box.com/Autodesk-PostEditing

[6] http://sourceforge.net/projects/tercpp/

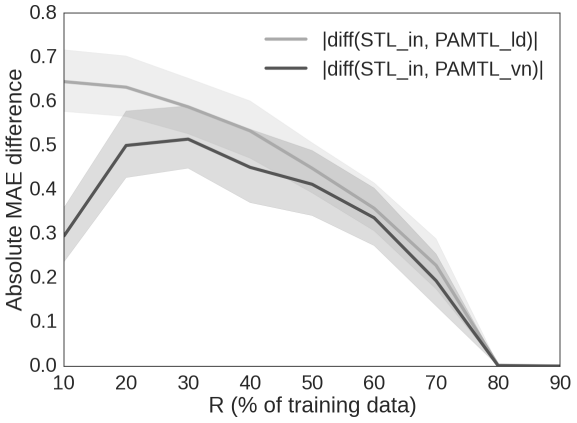

Figure 1: Validation curves for the R parameter.

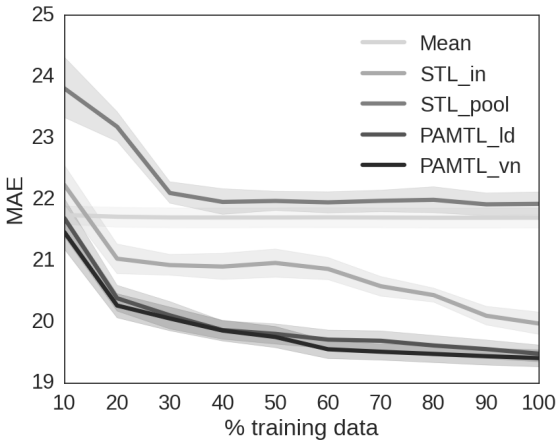

Figure 2: Learning curves for all the domains, computed by calculating the mean MAE (↓) of the four domains.

ity between IT_{LSP1} and IT_{LSP2} and the other two domains, which might make knowledge transfer across them more difficult and QE model reactivity to domain changes particularly important.

Features. Our models are trained using the 17 baseline features proposed in (Specia et al., 2009), extracted with the online version of the QuEst feature extractor (Shah et al., 2014). These features take into account the complexity of the source sentence (*e.g.* number of tokens, number of translations per source word) and the fluency of the translation (*e.g.* language model probabilities). Their description is available in (Callison-Burch et al., 2012). The results of previous WMT QE shared tasks have shown that these features are particularly competitive in the HTER prediction task.

Baselines. We compare the performance of PAMTL against three baselines: *i)* pooling mean, *ii)* pooling online single task learning (STL_{pool}) and *iii)* in-domain online single task learning (STL_{in}). The pooling mean is obtained by assigning a fixed prediction value to each test point. This value is the average HTER computed on the entire pool of training data. Although assigning the same prediction to each test instance would be useless in real applications, we compare against the mean baseline since it is often hard to beat in regression tasks, especially when dealing with heterogeneous data distributions (Rubino et al., 2013).

The two online single task baselines implement the PA algorithm described in Section 3.1. The choice of PA is to make them comparable to our method, so that we can isolate more precisely the contribution of multitask learning. STL_{pool} results are obtained by a single model trained on the entire pool of available training data presented in random order. STL_{in} results are obtained by separately training one model for each domain. These represent two alternative strategies for the integration of QE in the CAT framework. The former would allow a single model to simultaneously support multiple translation jobs in different domains, without any notion about their relations. The latter would lead to a more complex architecture, organized as a pool of independent, specialized QE modules.

Evaluation metric. The performance of our regression models is evaluated in terms of mean absolute error (MAE), a standard error measure for regression problems commonly used also for QE (Callison-Burch et al., 2012). The MAE is the average of the absolute errors $e_i = |\hat{y}_i - y_i|$, where $\hat{y}_i$ is the prediction of the model and y_i is the true value for the i^{th} instance. As it is an error measure, lower values indicate better performance (↓).

5 Results and Discussion

In this Section we evaluate the proposed PAMTL algorithm. First, by analyzing how the number of rounds R impacts on the performance of our approach, we empirically find the value that will be used to train the model. Then, the learned model is run on test data and compared against the baselines. Performance is analyzed both by averaging the MAE results computed on all the domains, and by separately discussing in-domain behavior. Finally, the capability of the algorithm to learn task correlations and, in turn, transfer knowledge across them, is analysed by presenting the correla-

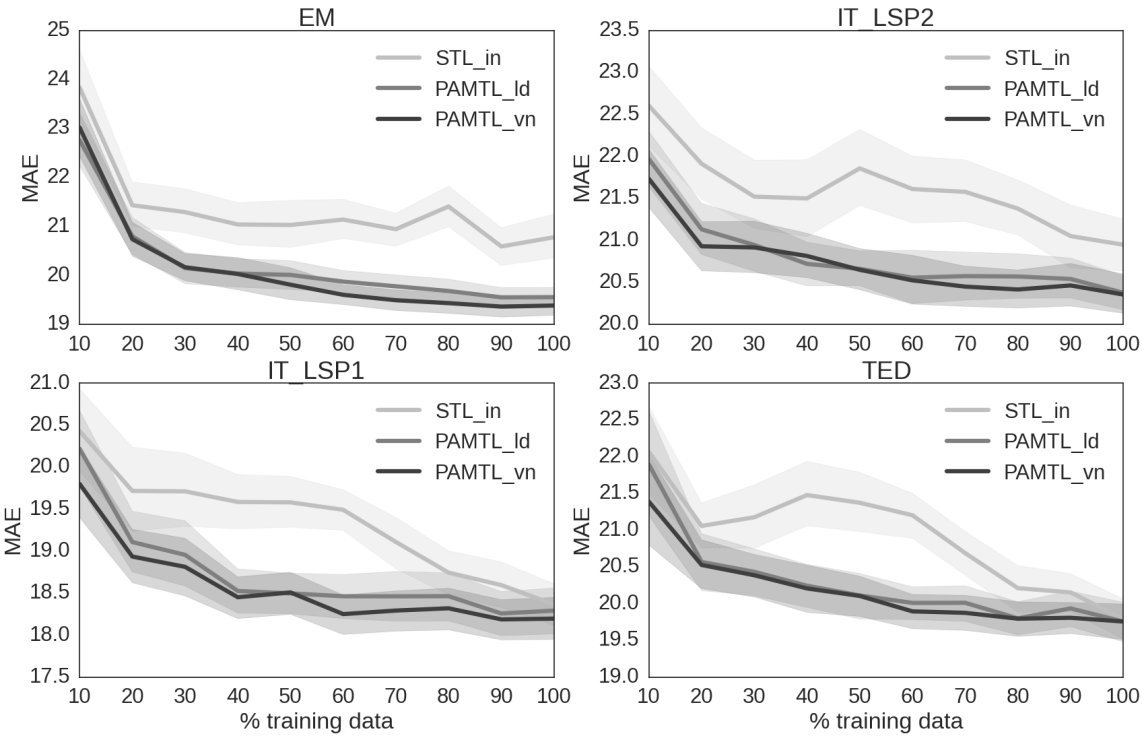

Figure 3: Learning curves showing MAE (↓) variations for each domain.

tion matrix of the task weights.

For the evaluation, we uniformly sample 700 instances from each domain for training, leaving the remaining 300 instances for test. The training sets of all the domains are concatenated and shuffled to create a random sequence of points. To investigate the impact of different amounts of data on the learning process, we create ten subsets of 10 to 100% of the training data. We optimize the parameters of all the models with a grid search procedure using 5-fold cross-validation. This process is repeated for 30 different train/test splits over the whole data. Results are presented with 95% confidence bands.[7]

Analysis of the R parameter. We empirically study the influence of the number of instances required to start updating the interaction matrix (the R parameter in Algorithm 1). For that, we perform a set of experiments where R is initialized with nine different values (expressed as percentage of training data). Figure 1 shows the validation curves obtained in cross-validation over the training data using the LogDet and Von Neumann updates. The curves report the performance (MAE) difference between STL_{in} and PAMTL_{ld} (black curve) and STL_{in} and PAMTL_{vn} (grey curve). The higher the difference, the better. The PAMTL_{vn} curve differs from PAMTL_{ld} one only for small values of R (< 20), showing that the two divergences are substantially equivalent. It is interesting to note that with only 20% of the training data ($R = 20$), PAMTL is able to find a stable set of weights and to effectively update the interaction matrix. Larger values of R harm the performance, indicating that the interaction matrix updates require a reasonable amount of points to reliably transfer knowledge across tasks. We use this observation to set R for our final experiment, in which we evaluate the methods over the test data.

Evaluation on test data. Global evaluation results are summarized in Figure 2, which shows five curves: one for each baseline (Mean, STL_{in}, STL_{pool}) and two for the proposed online multitask method (PAMTL_{vn} and PAMTL_{ld}). The curves are computed by calculating the average MAE achieved with different amounts of data on each domain's test set.

The results show that PAMTL_{ld} and PAMTL_{vn} have similar trends (confirming the substantial equivalence previously observed), and that both outperform all the baselines in a statistically significant manner. This holds for all the training set

[7] Confidence bands are used to show whether performance differences between the models are statistically significant.

sizes we experimented with. The maximum improvement over the baselines (+1.3 MAE) is observed with 60% of the training data when comparing PAMTL$_{vn}$ with STL$_{in}$. Even if this is the best baseline, also with 100% of the data its results are not competitive and of limited interest with respect to our application scenario (the integration of effective QE models in the CAT framework). Indeed, despite the STL$_{in}$ downward error trend, it's worth remarking that an increased competitiveness would come at the cost of: *i)* collecting large amounts of annotated data and *ii)* integrating the model in a complex CAT architecture organized as a pool of independent QE components. Under the tested conditions, it is also evident that the alternative strategy of using a single QE component to simultaneously serve multiple translation jobs is not viable. Indeed, STL$_{pool}$ is the worst performing baseline, with a constant distance of around 2 MAE points from the best PAMTL model for almost all the training set sizes. The fact that, with increasing amounts of data, the STL$_{pool}$ predictions get close to those of the simple mean baseline indicates its limitations to cope with the noise introduced by a continuous stream of diverse data. The capability to handle such stream by exploiting task relationships makes PAMTL a much better solution for our purposes.

Per-domain analysis. Figure 3 shows the MAE results achieved on each target domain by the most competitive baseline (STL$_{in}$) and the proposed online multitask method (PAMTL$_{vn}$, PAMTL$_{ld}$).

For all the domains, the behavior of PAMTL$_{ld}$ and PAMTL$_{vn}$ is consistent and almost identical. With both divergences, the improvement of PAMTL over online single task learning becomes statistically significant when using more than 30% of the training data (210 instances). Interestingly, in all the plots, with 20% of the training data (140 instances for each domain, *i.e.* a total of 560 instances adding data from all the domains), PATML results are comparable to those achieved by STL$_{in}$ with 80% of the training data (*i.e.* 560 in-domain instances). This confirms that PATML can effectively leverage data heterogeneity, and that a limited amount of in-domain data is sufficient to make it competitive. Nevertheless, for all domains except EM, the PATML and STL$_{in}$ curves converge to comparable performance when trained with 100% of the data. This is not surprising if we consider that EM has a varied vocabulary

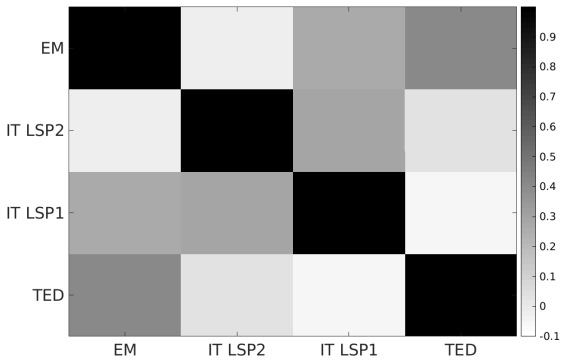

Figure 4: Correlation among the weights predicted by PATML$_{vn}$ using all the training data.

(see Table 1), which may be evidence of the presence of different topics, increasing its similarity with other domains. The same assumption should also hold for TED, given that its source sentences belong to talks about different topics. The results for the TED domain, however, do not present the same degree of improvement as for EM.

To better understand the relationships learned by the PAMTL models, we compute the correlation between the weights inferred for each domain (as performed by Saha et al. (2011)). Figure 4 shows the correlations computed on the task weights learned by PATML$_{vn}$ with all the training data. In the matrix, EM is the domain that presents the highest correlation with all the others. Instead, TED and IT$_{LSP2}$ are the less correlated with the other domains (even though, being close to the other IT domain, IT$_{LSP2}$ can share knowledge with it). This explains why the improvement measured on TED is smaller compared to EM. Although there is no canonical way to measure correlation among domains, the weights correlation matrix and the improvements achieved by PAMTL show the capability of the method to identify task relationships and exploit them to improve the generalization properties of the model.

6 Conclusion

We addressed the problem of developing quality estimation models suitable for integration in computer-assisted translation technology. In this framework, on-the-fly MT quality prediction for a stream of heterogeneous data coming from different domains/users/MT systems represents a major challenge. On one side, processing such stream calls for supervised solutions that avoid the bot-

tleneck of periodically retraining the QE models in a batch fashion. On the other side, handling data heterogeneity requires the capability to leverage data similarities and dissimilarities. While previous works addressed these two problems in isolation, by proposing approaches respectively based on online and multitask learning, our solution unifies the two paradigms in a single online multitask approach. To this aim, we developed a novel regression algorithm, filling a gap left by current online multitask learning methods that only operate in classification mode. Our approach, which is based on the passive aggressive algorithm, has been successfully evaluated against strong online single-task competitors in a scenario involving four domains. Our future objective is to extend our evaluation to streams of data coming from a larger number of domains. Finding reasonably-sized datasets for this purpose is currently difficult. However, we are confident that the gradual shift of the translation industry towards human MT post-editing will not only push for further research on these problems, but also provide data for larger scale evaluations in a short time.

To allow for replicability of our results and promote further research on QE, the features extracted from our data, the computed labels and the source code of the method are available at `https://github.com/jsouza/pamtl`.

Acknowledgements

This work has been partially supported by the EC-funded H2020 project QT21 (grant agreement no. 645452). The authors would like to thank Dr. Ventsislav Zhechev for his support with the Autodesk Post-Editing Data corpus.

References

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Massimo Pontil. 2008. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, January.

Nguyen Bach, F. Huang, and Y. Al-Onaizan. 2011. Goodness: A method for measuring machine translation confidence. In *49th Annual Meeting of the Association for Computational Linguistics*.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, USA, June.

Edwin Bonilla, Kian Ming Chai, and Christopher Williams. 2008. Multi-task Gaussian Process Prediction. In *Advances in Neural Information Processing Systems 20: NIPS'08*.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June.

Rich Caruana. 1997. Multitask learning. In *Machine Learning*, pages 41–75.

Giovanni Cavallanti, N Cesa-Bianchi, and C Gentile. 2010. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11:2901–2934.

Trevor Cohn and Lucia Specia. 2013. Modelling Annotator Bias with Multi-task Gaussian Processes: An application to Machine Translation Quality Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 32–42, Sofia, Bulgaria, August.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *The Journal of Machine Learning Research*, 7:551–585.

José G. C. de Souza, Marco Turchi, and Matteo Negri. 2014a. 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, August.

José G. C. de Souza, Marco Turchi, and Matteo Negri. 2014b. Towards a Combination of Online and Multitask Learning for MT Quality Estimation: a Preliminary Study. In *Proceedings of Workshop on Interactive and Adaptive Machine Translation in 2014 (IAMT 2014)*, Vancouver, BC, Canada, October.

Eric Eaton and PL Ruvolo. 2013. ELLA: An efficient lifelong learning algorithm. In *Proceedings of the 30th International Conference on Machine Learning*, pages 507–515, Atlanta, Georgia, USA, June.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico

Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frederic Blain, Philipp Koehn, Christian Buck, and Ulrich Germann. 2014. THE MATECAT TOOL. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132, Dublin, Ireland, August.

Fei Huang, Jian-Ming Xu, Abraham Ittycheriah, and Salim Roukos. 2014. Adaptive HTER Estimation for Document-Specific MT Post-Editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–870, Baltimore, Maryland, June.

Laurent Jacob, Jean-philippe Vert, Francis R Bach, and Jean-philippe Vert. 2009. Clustered Multi-Task Learning: A Convex Formulation. In D Koller, D Schuurmans, Y Bengio, and L Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 745–752. Curran Associates, Inc.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zenz, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the Machine Translation Workshop (WMT2012)*, pages 171–180, Montréal, Canada, June.

Francesco Parrella. 2007. Online support vector regression. *Master's Thesis, Department of Information Science, University of Genoa, Italy*.

Raphael Rubino, José G. C. de Souza, and Lucia Specia. 2013. Topic Models for Translation Quality Estimation for Gisting Purposes. In *Machine Translation Summit XIV*, pages 295–302.

Paul Ruvolo and Eric Eaton. 2014. Online Multi-Task Learning via Sparse Dictionary Optimization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, Québec City, Québec, Canada, July.

Avishek Saha, Piyush Rai, Hal Daumé, and Suresh Venkatasubramanian. 2011. Online Learning of Multiple Tasks and their Relationships. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA, April.

Kashif Shah, Marco Turchi, and Lucia Specia. 2014. An Efficient and User-friendly Tool for Machine Translation Quality Estimation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, May.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Association for Machine Translation in the Americas*, Cambridge, MA, USA, August.

Radu Soricut and A Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, number July, pages 612–621.

Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009. Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proceedings of the 13th Annual Conference of the EAMT*, pages 28–35, Barcelona, Spain, May.

Koji Tsuda, Gunnar Rätsch, and Manfred K Warmuth. 2005. Matrix exponentiated gradient updates for online learning and bregman projection. In *Journal of Machine Learning Research*, pages 995–1018.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the Subjectivity of Human Judgements in MT Quality Estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation (WMT)*, pages 240–251, Sofia, Bulgaria, August.

Marco Turchi, Antonios Anastasopoulos, José G. C. de Souza, and Matteo Negri. 2014. Adaptive Quality Estimation for Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 710–720, Baltimore, Maryland, USA, June.

Guillaume Wisniewski, Anil Kumar Singh, Natalia Segal, and François Yvon. 2013. Design and Analysis of a Large Corpus of Post-Edited Translations: Quality Estimation, Failure Analysis and the Variability of Post-Edition. In *Machine Translation Summit XIV*, pages 117–124.

Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, and Nicu Sebe. 2014. Multitask linear discriminant analysis for view invariant action recognition. *IEEE Transactions on Image Processing*, 23(12):5599–5611.

Yu Zhang and Dit-yan Yeung. 2010. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 733–742, Catalina Island, CA, USA, July.

Leon Wenliang Zhong and James T. Kwok. 2012. Convex multitask learning with flexible task clusters. In *Proceedings of the 29 th International Conference on Machine Learning*, Edinburgh, Scotland, June.