Adaptive Quality Estimation for Machine Translation

Marco Turchi[1] **Antonios Anastasopoulos**[3]
José G. C. de Souza[1,2] **Matteo Negri**[1]
[1] FBK - Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
[2] University of Trento, Italy
[3] National Technical University of Athens, Greece
{turchi,desouza,negri}@fbk.eu
anastasopoulos.ant@gmail.com

Abstract

The automatic estimation of machine translation (MT) output quality is a hard task in which the selection of the appropriate algorithm and the most predictive features over reasonably sized training sets plays a crucial role. When moving from controlled lab evaluations to real-life scenarios the task becomes even harder. For current MT quality estimation (QE) systems, additional complexity comes from the difficulty to model user and domain changes. Indeed, the instability of the systems with respect to data coming from different distributions calls for adaptive solutions that react to new operating conditions. To tackle this issue we propose an online framework for adaptive QE that targets reactivity and robustness to user and domain changes. Contrastive experiments in different testing conditions involving user and domain changes demonstrate the effectiveness of our approach.

1 Introduction

After two decades of steady progress, research in statistical machine translation (SMT) started to cross its path with translation industry with tangible mutual benefit. On one side, SMT research brings to the industry improved output quality and a number of appealing solutions useful to increase translators' productivity. On the other side, the market needs suggest concrete problems to solve, providing real-life scenarios to develop and evaluate new ideas with rapid turnaround. The evolution of computer-assisted translation (CAT) environments is an evidence of this trend, shown by the increasing interest towards the integration of suggestions obtained from MT engines with those derived from translation memories (TMs).

The possibility to speed up the translation process and reduce its costs by post-editing good-quality MT output raises interesting research challenges. Among others, these include deciding *what* to present as a suggestion, and *how* to do it in the most effective way.

In recent years, these issues motivated research on automatic QE, which addresses the problem of estimating the quality of a translated sentence given the source and without access to reference translations (Blatz et al., 2003; Specia et al., 2009; Mehdad et al., 2012). Despite the substantial progress done so far in the field and in successful evaluation campaigns (Callison-Burch et al., 2012; Bojar et al., 2013), focusing on concrete market needs makes possible to further define the scope of research on QE. For instance, moving from controlled lab testing scenarios to real working environments poses additional constraints in terms of adaptability of the QE models to the variable conditions of a translation job. Such variability is due to two main reasons:

1. **The notion of MT output quality is highly subjective** (Koponen, 2012; Turchi et al., 2013; Turchi and Negri, 2014). Since the quality standards of individual users may vary considerably (*e.g.* according to their knowledge of the source and target languages), the estimates of a static QE model trained with data collected from a group of post-editors might not fit with the actual judgements of a new user;

2. **Each translation job has its own specificities** (domain, complexity of the source text, average target quality). Since data from a new job may differ from those used to train the QE model, its estimates on the new instances might result to be biased or uninformative.

The ability of a system to self-adapt to the be-

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 710–720,
Baltimore, Maryland, USA, June 23-25 2014.

haviour of specific users and domain changes is a facet of the QE problem that so far has been disregarded. To cope with these issues and deal with the erratic conditions of real-world translation workflows, we propose **an adaptive approach to QE** that is sensitive and robust to differences between training and test data. Along this direction, our main contribution is a framework in which QE models can be trained and can continuously evolve over time accounting for knowledge acquired from post editors' work.

Our approach is **based on the online learning paradigm** and exploits a key difference between such framework and the batch learning methods currently used. On one side, the QE models obtained with batch methods are learned exclusively from a predefined set of training examples under the assumption that they have similar characteristics with respect to the test data. This makes them suitable for controlled evaluation scenarios where such condition holds. On the other side, online learning techniques are designed to learn in a stepwise manner (either from scratch, or by refining an existing model) from new, unseen test instances by taking advantage of external feedback. This makes them suitable for real-life scenarios where the new instances to be labelled can considerably differ from the data used to train the QE model.

To develop our approach, different online algorithms have been embedded in the backbone of a QE system. This required the adaptation of its standard batch learning workflow to:

1. Perform online feature extraction from a source–target pair (*i.e.* one instance at a time instead of processing an entire training set);
2. Emit a prediction for the input instance;
3. Gather user feedback for the instance (*i.e.* calculating a "true label" based on the amount of user post-editions);
4. Send the true label back to the model to update its predictions for future instances.

Focusing on the adaptability to user and domain changes, we report the results of comparative experiments with two online algorithms and the standard batch approach. The evaluation is carried out by measuring the global error of each algorithm on test sets featuring different degrees of similarity with the data used for training. Our results show that the sensitivity of online QE models to different distributions of training and test instances makes them more suitable than batch methods for integration in a CAT framework.

Our adaptive QE infrastructure has been released as open source. Its C++ implementation is available at `http://hlt.fbk.eu/technologies/aqet`.

2 Related work

QE is generally cast as a supervised machine learning task, where a model trained from a collection of (*source, target, label*) instances is used to predict labels[1] for new, unseen test items (Specia et al., 2010).

In the last couple of years, research in the field received a strong boost by the shared tasks organized within the WMT workshop on SMT,[2] which is also the framework of our first experiment in §5. Current approaches to the tasks proposed at WMT have mainly focused on three main directions, namely: *i)* feature engineering, as in (Hardmeier et al., 2012; de Souza et al., 2013a; de Souza et al., 2013b; Rubino et al., 2013b), *ii)* model learning with a variety of classification and regression algorithms, as in (Bicici, 2013; Beck et al., 2013; Soricut et al., 2012), and *iii)* feature selection as a way to overcome sparsity and overfitting issues, as in (Soricut et al., 2012).

Being optimized to perform well on specific WMT sub-tasks and datasets, current systems reflect variations along these directions but leave important aspects of the QE problem still partially investigated or totally unexplored.[3] Among these, the necessity to model the diversity of human quality judgements and correction strategies (Koponen, 2012; Koponen et al., 2012) calls for solutions that: *i)* account for annotator-specific behaviour, thus being capable of learning from inherently noisy datasets produced by multiple annotators, and *ii)* self-adapt to changes in data distribution, learning from user feedback on new, unseen test items.

[1] 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* (Snover et al., 2006), and post-editing *time* (*e.g.* seconds per word).

[2] `http://www.statmt.org/wmt13/`

[3] For a comprehensive overview of the QE approaches proposed so far we refer the reader to the WMT12 and WMT13 QE shared task reports (Callison-Burch et al., 2012; Bojar et al., 2013).

These interconnected issues are particularly relevant in the CAT framework, where translation jobs from different domains are routed to professional translators with different idiolect, background and quality standards.

The first aspect, modelling annotators' individual behaviour and interdependences, has been addressed by Cohn and Specia (2013), who explored multi-task Gaussian Processes as a way to jointly learn from the output of multiple annotations. This technique is suitable to cope with the unbalanced distribution of training instances and yields better models when heterogeneous training datasets are available.

The second problem, the adaptability of QE models, has not been explored yet. A common trait of all current approaches, in fact, is the reliance on batch learning techniques, which assume a "static" nature of the world where new unseen instances that will be encountered will be similar to the training data.[4] However, similarly to translation memories that incrementally store translated segments and evolve over time incorporating users style and terminology, all components of a CAT tool (the MT engine and the mechanisms to assign quality scores to the suggested translations) should take advantage of translators feedback.

On the MT system side, research on adaptive approaches tailored to interactive SMT and CAT scenarios explored the online learning protocol (Littlestone, 1988) to improve various aspects of the decoding process (Cesa-Bianchi et al., 2008; Ortiz-Martínez et al., 2010; Martínez-Gómez et al., 2011; Martínez-Gómez et al., 2012; Mathur et al., 2013; Bertoldi et al., 2013).

As regards QE models, our work represents the first investigation on incremental adaptation by exploiting users feedback to provide targeted (system, user, or project specific) quality judgements.

3 Online QE for CAT environments

When operating with advanced CAT tools, translators are presented with suggestions (either matching fragments from a translation memory or automatic translations produced by an MT system) for each sentence of a source document. Before being approved and published, translation suggestions may require different amounts of post-editing operations depending on their quality.

[4]This assumption holds in the WMT evaluation scenario, but it is not necessarily valid in real operating conditions.

Each post-edition brings a wealth of dynamic knowledge about the whole translation process and the involved actors. For instance, adaptive QE components could exploit information about the distance between automatically assigned scores and the quality standards of individual translators (inferred from the amount of their corrections) to "profile" their behaviour.

The online learning paradigm fits well with this research objective. In the online framework, differently from the batch mode, the learning algorithm sequentially processes an unknown sequence of instances $X = x_1, x_2, ..., x_n$, returning a prediction $p(x_i)$ as output at each step. Differences between $p(x_i)$ and the true label $\hat{p}(x_i)$ obtained as feedback are used by the learner to refine the next prediction $p(x_{i+1})$.

In our experiments on adaptive QE we aim to predict the quality of the suggested translations in terms of HTER, which measures the minimum edit distance between the MT output and its manually post-edited version in the [0,1] interval.[5] In this scenario:

- The set of instances X is represented by (*source, target*) pairs;
- The prediction $p(x_i)$ is the automatically estimated HTER score;
- The true label $\hat{p}(x_i)$ is the actual HTER score calculated over the target and its post-edition.

At each step of the process, the goal of the learner is to exploit user post-editions to reduce the difference between the predicted HTER values and the true labels for the following (*source, target*) pairs.

As depicted in Figure 1, this is done as follows:

1. At step *i*, an unlabelled (*source, target*) pair x_i is sent to a feature extraction component. To this aim, we used an adapted version (Shah et al., 2014) of the open-source QuEst[6] tool (Specia et al., 2013). The tool, which implements a large number of features proposed by participants in the WMT QE shared tasks, has been modified to process one sentence at a time as requested for integration in a CAT environment;

[5]Edit distance is calculated as the number of edits (word insertions, deletions, substitutions, and shifts) divided by the number of words in the reference. Lower HTER values indicate better translations.

[6]`http://www.quest.dcs.shef.ac.uk/`

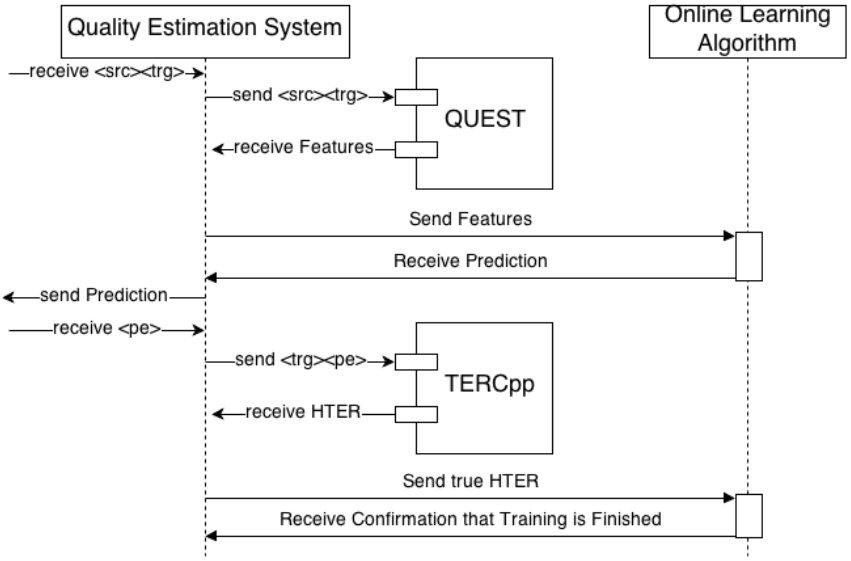

Figure 1: Online QE workflow. <src>, <trg> and <pe> respectively stand for the source sentence, the target translation and the post-edited target.

2. The extracted features are sent to an online regressor, which returns a QE prediction score $p(x_i)$ in the [0,1] interval (set to 0 at the first round of the iteration);

3. Based on the post-edition done by the user, the true HTER label $\hat{p}(x_i)$ is calculated by means of the TERCpp[7] open source tool;

4. The true label is sent back to the online algorithm for a stepwise model improvement. The updated model is then ready to process the following instance x_{i+1}.

This new paradigm for QE makes it possible to: *i)* let the QE system learn from one point at a time without complete re-training from scratch, *ii)* customize the predictions of an existing QE model with respect to a specific situation (post-editor or domain), or even *iii)* build a QE model from scratch when training data is not available.

For the sake of clarity it is worth observing that, at least in principle, a model built in a batch fashion could also be adapted to new test data. For instance, this could be done by running periodic re-training routines once a certain amount of new labelled instances has been collected (*de facto* mimicking an online process). Such periodic updates, however, would not represent a viable solution in the CAT framework where post-editors' work cannot be slowed by time-consuming procedures to re-train core system components from scratch.

[7] goo.gl/nkh2rE

4 Evaluation framework

To measure the adaptation capability of different QE models, we experiment with a range of conditions defined by variable degrees of similarity between training and test data.

The degree of similarity depends on several factors: the MT engine used, the domain of the documents to be translated, and the post-editing style of individual translators. In our experiments, the degree of similarity is measured in terms of ΔHTER, which is computed as the absolute value of the difference between the average HTER of the training and test sets. Large values indicate a low similarity between training and test data and a more challenging scenario for the learning algorithms.

4.1 Experimental setup

In the range of possible evaluation scenarios, our experiments cover:

- One artificial setting (§5) obtained from the WMT12 QE shared task data, in which training/test instances are arranged to reflect homogeneous distributions of the HTER labels.

- Two settings obtained from data collected with a CAT tool in real working conditions, in which different facets of the adaptive QE problem interact with each other. In the first (`user_change`, §6.1), training and test data from the same domain are obtained from different users. In the sec-

ond (user+domain_change, §6.2), training and test data are obtained from different users and domains.

For each setting, we compare an *adaptive* and an *empty* model against a system trained in *batch* mode. The *adaptive* model is built on top of an existing model created from the training data and exploits the new test instances to refine its predictions in a stepwise manner. The *empty* model only learns from the test set, simulating the worst condition where training data is not available. The *batch* model is built by learning only from the training data and is evaluated on the test set without exploiting information from the test instances.

Each model is also compared against a common baseline for regression tasks, which is particularly relevant in settings featuring different data distributions between training and test sets. This baseline (μ henceforth) is calculated by labelling each instance of the test set with the mean HTER score of the training set. Previous works (Rubino et al., 2013a) demonstrated that its results can be particularly hard to beat.

4.2 Performance indicator and feature set

To measure the adaptability of our model to a given test set we compute the Mean Absolute Error (MAE), a metric for regression problems also used in the WMT QE shared tasks. The MAE is the average of the absolute errors $e_i = |f_i - y_i|$, where f_i is the prediction of the model and y_i is the true value for the i^{th} instance.

As our focus is on the algorithmic aspect, in all experiments we use the same feature set, which consists of the seventeen features proposed in (Specia et al., 2009). This feature set, fully described in (Callison-Burch et al., 2012), takes into account the complexity of the source sentence (*e.g.* number of tokens, number of translations per source word) and the fluency of the target translation (*e.g.* language model probabilities). The results of previous WMT QE shared tasks have shown that these baseline features are particularly competitive in the regression task (with only few systems able to beat them at WMT12).

4.3 Online algorithms

In our experiments we evaluate two online algorithms, OnlineSVR (Parrella, 2007)[8] and Passive-Aggressive Perceptron (Crammer et al., 2006),[9] by comparing their performance with a batch learning strategy based on the Scikit-learn implementation of Support Vector Regression (SVR).[10]

The choice of the OnlineSVR and Passive-Aggressive (OSVR and PA henceforth) is motivated by different considerations. From a **performance** point of view, as an adaptation of ϵ-SVR which proved to be one of the top performing algorithms in the regression QE tasks at WMT, OSVR seems to be the best candidate. For this reason, we use the online adaptation of ϵ-SVR proposed by (Ma et al., 2003). The goal of OnlineSVR is to find a way to add each new sample to one of three sets (support, empty, error) maintaining the consistency of a set of conditions known as Karush-Kuhn Tucker (KKT) conditions. For each new point, OSVR starts a cycle where the samples are moved across the three sets until the KKT conditions are verified and the new point is assigned to one of the sets. If the point is identified as a support vector, the parameters of the model are updated. This allows OSVR to benefit from the prediction capability of ϵ-SVR in an online setting.

From a **practical** point of view, providing the best trade off between accuracy and computational time (He and Wang, 2012), PA represents a good solution to meet the demand of efficiency posed by the CAT framework. For each instance *i*, after emitting a prediction and receiving the true label, PA computes the ϵ-insensitive hinge loss function. If its value is larger than the tolerance parameter (ϵ), the weights of the model are updated as much as the aggressiveness parameter C allows. In contrast with OSVR, which keeps track of the most important points seen in the past (support vectors), the update of the weights is done without considering the previously processed *i-1* instances. Although it makes PA faster than OSVR, this is a riskier strategy because it may lead the algorithm to change the model to adapt to outlier points.

5 Experiments with WMT12 data

The motivations for experiments with training and test data featuring homogeneous label distributions are twofold. First, since in this artificial scenario adaptation capabilities are not required for the QE component, batch methods operate in the ideal conditions (as training and test are indepen-

[8] http://www2.imperial.ac.uk/~gmontana/onlinesvr.htm

[9] https://code.google.com/p/sofia-ml/

[10] http://scikit-learn.org/

WMT Dataset								
Train	Test	Δ HTER	μ MAE	Batch MAE	Adaptive MAE	Adaptive Alg.	Empty MAE	Empty Alg.
200	754	0.39	13.7	**13.2**	**13.2***	OSVR	13.5*	OSVR
600	754	1.32	13.8	**12.7**	12.9*	OSVR	13.5*	OSVR
1500	754	1.22	13.8	**12.7**	12.8*	OSVR	13.5*	OSVR

Table 1: MAE of the best performing *batch*, *adaptive* and *empty* models on WMT12 data. Training sets of different size and the test set have been arranged to reflect homogeneous label distributions.

dent and identically distributed). This makes possible to obtain from batch models the best possible performance to compare with. Second, this scenario provides the fairest conditions for such comparison because, in principle, online algorithms are not favoured by the possibility to learn from the diversity of the test instances.

For our controlled experiments we use the WMT12 English-Spanish corpus, which consists of 2,254 source-target pairs (1,832 for training, 422 for test). The HTER labels for our regression task are calculated from the post-edited version and the target sentences provided in the dataset.

To avoid biases in the label distribution, the WMT12 training and test data have been merged, shuffled, and eventually separated to generate three training sets of different size (200, 600, and 1500 instances), and one test set with 754 instances. For each algorithm, the training sets are used for learning the QE models, optimizing parameters (*i.e.* C, ϵ, the kernel and its parameters for SVR and OSVR; tolerance and aggressiveness for PA) through grid search in 10-fold cross-validation.

Evaluation is carried out by measuring the performance of the *batch* (learning only from the training set), the *adaptive* (learning from the training set and adapting to the test set), and the *empty* (learning from scratch from the test set) models in terms of global MAE scores on the test set.

Table 1 reports the results achieved by the best performing algorithm for each type of model (*batch*, *adaptive*, *empty*). As can be seen, close MAE values show a similar behaviour for the three types of models.[11] With the same amount of training data, the performance of the batch and the adaptive models (in this case always obtained with OSVR) is almost identical. This demonstrates that, as expected, the online algorithms do not take advantage of test data with a label distribution similar to the training set. All the models outperform the baseline, even if the minimal differences confirm the competitiveness of such a simple approach.

Overall, these results bring some interesting indications about the behaviour of the different online algorithms. First, the good results achieved by the empty models (less than one MAE point separates them from the best ones built on the largest training set) suggest their high potential when training data are not available. Second, our results show that OSVR is always the best performing algorithm for the adaptive and empty models. This suggests a lower capability of PA to learn from instances similar to the training data.

6 Experiments with CAT data

To experiment with adaptive QE in more realistic conditions we used a CAT tool[12] to collect two datasets of (*source*, *target*, *post_edited target*) English-Italian tuples.The source sentences in the datasets come from two documents from different domains, respectively legal (L) and information technology (IT). The L document, which was extracted from a European Parliament resolution published on the EUR-Lex platform,[13] contains 164 sentences. The IT document, which was taken from a software user manual, contains 280 sentences. The source sentences were translated with two SMT systems built by training the Moses toolkit (Koehn et al., 2007) on parallel data from the two domains (about 2M sentences for IT and 1.5M for L). Post-editions were collected from eight professional translators (four for each document) operating with the CAT tool in real working conditions.

According to the way they are created, the two datasets allow us to evaluate the adaptability of different QE models with respect to user changes

[11] Results marked with the "*" symbol are NOT statistically significant compared to the corresponding batch model. The others are always statistically significant at $p \leq 0.005$, calculated with approximate randomization (Yeh, 2000).

[12] MateCat – `http://www.matecat.com/`

[13] `http://eur-lex.europa.eu/`

user_change								
Legal Domain								
Train	Test	Δ HTER	μ MAE	Batch MAE	Adaptive MAE	Adaptive Alg.	Empty MAE	Empty Alg.
rad	cons	20.5	21.4	20.6	14.5	PA	**12.5**	OSVR
cons	rad	19.4	21.2	21.3	16.1	PA	**11.3**	OSVR
sim1	sim2	3.3	14.7	**12.2**	12.6*	OSVR	12.9*	OSVR
sim2	sim1	3.2	13.4	**13.3**	13.9*	OSVR	15.2*	OSVR
IT Domain								
Train	Test	Δ HTER	μ MAE	Batch MAE	Adaptive MAE	Adaptive Alg	Empty MAE	Empty Alg
cons	rad	12.8	19.2	19.8	17.5*	OSVR	**16.6**	OSVR
rad	cons	9.6	16.8	16.6	15.6	PA	**15.5**	OSVR
sim2	sim1	3.3	14.7	**14.4**	15*	OSVR	15.5*	OSVR
sim1	sim2	1.1	15	**13.9**	14.4*	OSVR	16.1*	OSVR

Table 2: MAE of the best performing *batch*, *adaptive* and *empty* models on CAT data collected from different users in the same domain.

within the same domain (§6.1), as well as user and domain changes at the same time (§6.2).

For each document *D* (L or IT), these two scenarios are obtained by dividing *D* into two parts of equal size (80 instances for L and 140 for IT). The result is one training set and one test set for each post-editor within the same domain. For the `user_change` experiments, training and test sets are selected from different post-editors within the same domain. For the `user+domain_change` experiments, training and test sets are selected from different post-editors in different domains.

On each combination of training and test sets, the *batch*, *adaptive*, and *empty* models are trained and evaluated in terms of global MAE scores on the test set.

6.1 Dealing with user changes

Among the possible combinations of training and test data from different post-editors in the same domain, Table 2 refers to two opposite scenarios. For each domain, these respectively involve the most dissimilar and the most similar post-editors according to the ΔHTER. Also in this case, for each model (*batch*, *adaptive* and *empty*) we only report the MAE of the best performing algorithm.

The first scenario defines a challenging situation where two post-editors (*rad* and *cons*) are characterized by opposite behaviour. As evidenced by the high ΔHTER values, one of them (*rad*) is the most "radical" post-editor (performing more corrections) while the other (*cons*) is the most "conservative" one. As shown in Table 2, global MAE scores for the online algorithms (both *adaptive* and *empty*) indicate their good adaptation capabilities. This is evident from the significant improvements both over the baseline (μ) and the batch models. Interestingly, the best results are always achieved by the *empty* models (with MAE reductions up to 10 points when tested on *rad* in the L domain, and 3.2 points when tested on *rad* in the IT domain). These results (MAE reductions are always statistically significant) suggest that, when dealing with datasets with very different label distributions, the evident limitations of batch methods are more easily overcome by learning from scratch from the feedback of a new post-editor. This also holds when the amount of test points to learn from is limited, as in the L domain where the test set contains only 80 instances. From the application-oriented perspective that motivates our work, considering the high costs of acquiring large and representative QE training data, this is an important finding.

The second scenario defines a less challenging situation where the two post-editors (*sim1* and *sim2*) are characterized by the most similar behaviour (small ΔHTER). This scenario is closer to the situation described in Section §5. Also in this case MAE results for the *adaptive* and *empty* models are slightly worse, but not significantly, than those of the batch models and the baseline. However, considering the very small amount of "uninformative" instances to learn from (especially for the *empty* models), these lower results are not surprising.

A closer look at the behaviour of the online algorithms in the two domains leads to other observations. First, OSVR always outperforms PA for the *empty* models and when post-editors have sim-

user+domain_change								
Train	Test	Δ HTER	μ MAE	Batch MAE	Adaptive MAE	Adaptive Alg	Empty MAE	Empty Alg
L cons	IT rad	24.5	26.4	27	18.2	OSVR	**16.6**	OSVR
IT rad	L cons	24.0	24.9	25.4	19.7	OSVR	**12.5**	OSVR
L rad	L cons	20.5	21.4	20.6	14.5	PA	**12.5**	OSVR
L cons	L rad	19.4	21.2	21.3	16.1	PA	**11.3**	OSVR
IT cons	L cons	13.5	17.3	17.5	15.7	OSVR	**12.5**	OSVR
IT cons	IT rad	12.8	19.2	19.8	17.5	OSVR	**16.6**	OSVR
L cons	IT cons	12.7	17.6	17.6	**15.1**	OSVR	15.5	OSVR
IT rad	IT cons	9.6	16.8	16.6	15.6	PA	**15.5**	OSVR
IT cons	L rad	8.3	12.3	13	**10.7**	OSVR	11.3	OSVR
L rad	IT rad	6.8	17	16.9	**16.2**	OSVR	16.6	OSVR
L rad	IT cons	5.0	15.4	16.2	**14.7**	OSVR	15.5	OSVR
IT rad	L rad	2.2	10.6	10.8	**10.5**	OSVR	11.3	OSVR

Table 3: MAE of the best performing *batch*, *adaptive* and *empty* models on CAT data collected from different users and domains.

ilar behaviour, which are situations where the algorithm does not have to quickly adapt or react to sudden changes.

Second, PA seems to perform better for the *adaptive* models when the post-editors have significantly different behaviour and a quick adaptation to the incoming points is required. This can be motivated by the fact that PA relies on a simpler and less robust learning strategy that does not keep track of all the information coming from the previously processed instances, and can easily modify its weights taking into consideration the last seen point (see Section §3). For OSVR the addition of new points to the support set may have a limited effect on the whole model, in particular if the number of points in the set is large. This also results in a different processing time for the two algorithms.[14] For instance, in the *empty* configurations on IT data, OSVR devotes $6.0\ ms$ per instance to update the model, while PA devotes $4.8\ ms$, which comes at the cost of lower performance.

6.2 Dealing with user and domain changes

In the last round of experiments we evaluate the reactivity of different online models to simultaneous user and domain changes. To this aim, our QE models are created using a training set coming from one domain (L or IT), and then used to predict the HTER labels for the test instances coming from the other domain (*e.g.* training on L, testing on IT).

Among the possible combinations of training and test data, Table 3 refers to scenarios involving the most conservative and radical post-editors in each domain (previously identified with *cons* and *rad*)[15]. In the table, results are ordered according to the ΔHTER computed between the selected post-editor in the training domain (*e.g. L cons*) and the selected post-editor in the test domain (*e.g. IT rad*). For the sake of comparison, we also report (grey rows) the results of the experiments within the same domain presented in §6.1. For each type of model (*batch*, *adaptive* and *empty*) we only show the MAE obtained by the best performing algorithm.

Intuitively, dealing with simultaneous user and domain changes represents a more challenging problem compared to the previous setting where only post-editors changes were considered. Such intuition is confirmed by the results of the *adaptive* models that outperform both the baseline (μ) and the *batch* models even for low ΔHTER values. Although in these cases the distance between training and test data is comparable to the experiments with similar post-editors working in the same domain (*sim1* and *sim2*), here the predictive power of the *batch* models seems in fact to be lower. The same holds also for the *empty* models except in two cases where the ΔHTER is the smallest (2.2 and 5.0). This is a strong evidence of the fact that, in case of domain changes, online models can still learn from new test instances even if they have a label distribution similar to the training set.

When the distance between training and test increases, our results confirm our previous findings

[14] Their complexity depends on the number of features (*f*) and the number of previously seen instances (*n*). While for PA it is linear in *f*, *i.e. O(f)*, for OSVR it is quadratic in *n*, *i.e.* $O(n^2 * f)$.

[15] For brevity, we omit the results for the other post-editors which, however, show similar trends with respect to the previous experiments.

about the potential of the *empty* models. The observed MAE reductions range in fact from 10.4 to 12.9 points for the two combinations with the highest ΔHTER.

From the algorithmic point of view, our results indicate that OSVR achieves the best performance for all the combinations involving user and domain changes. This contrasts with the results of most of the combinations involving only user changes with post-editors characterized by opposite behaviour (grey rows in Table 3). However, it has to be remarked that in the case of heterogeneous datasets the difference between the two algorithms is always very high. In our experiments, when PA outperforms OSVR, its MAE results are significantly lower and vice-versa (respectively up to 1.5 and 1.7 MAE points). This suggests that, although PA is potentially capable of achieving higher results and better adapt to the new test points, its instability makes it less reliable for practical use.

As a final analysis of our results, we investigated how the performance of the different types of models (*batch*, *adaptive*, *empty*) relates to the distance between training and test sets. To this aim, we computed the Pearson correlation between the ΔHTER (column 3 in Table 3) and the MAE of each model (columns 5, 6 and 8), which respectively resulted in 0.9 for the *batch*, 0.63 for the *adaptive* and -0.07 for the *empty* model. These values confirm that *batch* models are heavily affected by the dissimilarity between training and test data: large differences in the label distribution imply higher MAE results and vice-versa. This is in line with our previous findings about *batch* models that, learning only from the training set, cannot leverage possible dissimilarities of the test set. The lower correlation observed for the *adaptive* models also confirms our intuitions: adapting to the new test points, these models are in fact more robust to differences with the training data. As expected, the results of the *empty* models are completely uncorrelated with the ΔHTER since they only use the test set.

This analysis confirms that, even when dealing with different domains, the similarity between the training and test data is one of the main factors that should drive the choice of the QE model. When this distance is minimal, *batch* models can be a reasonable option, but when the gap between training and test data increases, *adaptive* or *empty* models are a preferable choice to achieve good results.

7 Conclusion

In the CAT scenario, each translation job can be seen as a complex situation where the user (his personal style and background), the source document (the language and the domain) and the underlying technology (the translation memory and the MT engine that generate translation suggestions) contribute to make the task unique. So far, the adaptability to such specificities (a major challenge for CAT technology) has been mainly supported by the evolution of translation memories, which incrementally store translated segments incorporating the user style. The wide adoption of translation memories demonstrates the importance of capitalizing on such information to increase translators productivity.

While this lesson recently motivated research on adaptive MT decoders that learn from user corrections, nothing has been done to develop adaptive QE components. In the first attempt to address this problem, we proposed the application of the online learning protocol to leverage users feedback and to tailor QE predictions to their quality standards. Besides highlighting the limitations of current batch methods to adapt to user and domain changes, we performed an application-oriented analysis of different online algorithms focusing on specific aspects relevant to the CAT scenario. Our results show that the wealth of dynamic knowledge brought by user corrections can be exploited to refine in a stepwise fashion the quality judgements in different testing conditions (user changes as well as simultaneous user and domain changes).

As an additional contribution, to spark further research on this facet of the QE problem, our adaptive QE infrastructure (integrating all the components and the algorithms described in this paper) has been released as open source. Its C++ implementation is available at `http://hlt.fbk.eu/technologies/aqet`.

Acknowledgements

This work has been partially supported by the EC-funded project MateCat (ICT-2011.4.2-287688).

References

Daniel Beck, Kashif Shah, Trevor Cohn, and Lucia Specia. 2013. SHEF-Lite: When less is more for translation quality estimation. In *Proceedings of the*

8^{th} *Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August.

Nicola Bertoldi, Mauro Cettolo, and Federico Marcello. 2013. Cache-based Online Adaptation for Machine Translation Enhanced Computer Assisted Translation. In *Proceedings of the XIV Machine Translation Summit*, pages 1147–1162, Nice, France.

Ergun Bicici. 2013. Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the 8^{th} Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence Estimation for Machine Translation. Summer workshop final report, JHU/CLSP.

Ondrej 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 *Proceedings of the 8^{th} Workshop on Statistical Machine Translation*, WMT-2013, pages 1–44, Sofia, Bulgaria.

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 7^{th} Workshop on Statistical Machine Translation (WMT'12)*, pages 10–51, Montréal, Canada.

Nicolò Cesa-Bianchi, Gabriel Reverberi, and Sandor Szedmak. 2008. Online Learning Algorithms for Computer-Assisted Translation. Deliverable D4.2, SMART: Statistical Multilingual Analysis for Retrieval and Translation.

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 51^{st} Annual Meeting of the Association for Computational Linguistics*, ACL-2013, pages 32–42, Sofia, Bulgaria.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *J. Mach. Learn. Res.*, 7:551–585, December.

José G.C. de Souza, Christian Buck, Marco Turchi, and Matteo Negri. 2013a. FBK-UEdin participation to the WMT13 quality estimation shared task. In *Proceedings of the 8^{th} Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August.

José G.C. de Souza, Miquel Esplà-Gomis, Marco Turchi, and Matteo Negri. 2013b. Exploiting Qualitative Information from Automatic Word Alignment for Cross-lingual NLP Tasks. In *Proceedings of the 51^{st} Annual Meeting of the Association for Computational Linguistics - Short Papers*, pages 771–776, Sofia, Bulgaria.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Tree Kernels for Machine Translation Quality Estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation (WMT'12)*, pages 109–113, Montréal, Canada.

Zhengyan He and Houfeng Wang. 2012. A Comparison and Improvement of Online Learning Algorithms for Sequence Labeling. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1147–1162, Mumbai, India.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45^{th} Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Maarit Koponen, Wilker Aziz, Luciana Ramos, and Lucia Specia. 2012. Post-editing Time as a Measure of Cognitive Effort. In *Proceedings of the AMTA 2012 Workshop on Post-editing Technology and Practice (WPTP 2012)*, San Diego, California.

Maarit Koponen. 2012. Comparing Human Perceptions of Post-editing Effort with Post-editing Operations. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 181–190, Montréal, Canada.

Nick Littlestone. 1988. Learning Quickly when Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. In *Machine Learning*, pages 285–318.

Junshui Ma, James Theiler, and Simon Perkins. 2003. Accurate Online Support Vector Regression. *Neural Computation*, 15:2683–2703.

Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. 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.

Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. 2012. Online adaptation strategies for statistical machine translation in post-editing scenarios. *Pattern Recognition*, 45(9):3193–3203, September.

Prashant Mathur, Mauro Cettolo, and Marcello Federico. 2013. Online Learning Approaches in Computer Assisted Translation. In *Proceedings of the 8^{th} Workshop on Statistical Machine Translation*, Sofia, Bulgaria.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012. Match without a Referee: Evaluating MT Adequacy without Reference Translations. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 171–180, Montréal, Canada.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. 2010. Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 546–554, Stroudsburg, PA, USA.

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, Jennifer Foster, and Lucia Specia. 2013a. Topic Models for Translation Quality Estimation for Gisting Purposes. In *Proceedings of the Machine Translation Summit XIV*, Nice, France.

Raphael Rubino, Antonio Toral, S Cortés Vaíllo, Jun Xie, Xiaofeng Wu, Stephen Doherty, and Qun Liu. 2013b. The CNGL-DCU-Prompsit translation systems for WMT13. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 211–216, Sofia, Bulgaria.

Kashif Shah, Marco Turchi, and Lucia Specia. 2014. An Efficient and User-friendly Tool for Machine Translation Quality Estimation. In *Proceedings of the 9^{t}h International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Radu Soricut, Nguyen Bach, and Ziyuan Wang. 2012. The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In *Proceedings of the 7th Workshop on Statistical Machine Translation (WMT'12)*, pages 145–151, Montréal, Canada.

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 European Association for Machine Translation (EAMT'09)*, pages 28–35, Barcelona, Spain.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine Translation Evaluation versus Quality Estimation. *Machine translation*, 24(1):39–50.

Lucia Specia, Kashif Shah, José G.C. de Souza, and Trevor Cohn. 2013. QuEst - A Translation Quality Estimation Framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, ACL-2013, pages 79–84, Sofia, Bulgaria.

Marco Turchi and Matteo Negri. 2014. Automatic Annotation of Machine Translation Datasets with Binary Quality Judgements. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the Subjectivity of Human Judgements in MT Quality Estimation. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 240–251, Sofia, Bulgaria.

Alexander Yeh. 2000. More Accurate Tests for the Statistical Significance of Result Differences. In *Proceedings of the 18th conference on Computational linguistics (COLING 2000) - Volume 2*, pages 947–953, Saarbrucken, Germany.