Towards a Combination of Online and Multitask Learning for MT Quality Estimation: a Preliminary Study

José G. C. de Souza desouza@fbk.eu
University of Trento, Italy
Fondazione Bruno Kessler, Italy

Marco Turchi turchi@fbk.eu
Matteo Negri negri@fbk.eu
Fondazione Bruno Kessler, Italy

Abstract

Quality estimation (QE) for machine translation has emerged as a promising way to provide real-world applications with methods to estimate at run-time the reliability of automatic translations. Real-world applications, however, pose challenges that go beyond those of current QE evaluation settings. For instance, the heterogeneity and the scarce availability of training data might contribute to significantly raise the bar. To address these issues we compare two alternative machine learning paradigms, namely *online* and *multi-task* learning, measuring their capability to overcome the limitations of current batch methods. The results of our experiments, which are carried out in the same experimental setting, demonstrate the effectiveness of the two methods and suggest their complementarity. This indicates, as a promising research avenue, the possibility to combine their strengths into an online multi-task approach to the problem.

1 Introduction

Quality estimation (QE) for machine translation (MT) is the task of estimating the quality of a translated sentence at run-time and without access to reference translations (Specia et al., 2009).

As a quality indicator, in a typical QE setting, automatic systems have to predict either the time or the number of editing operations (e.g. in terms of HTER[1]) required by a human to transform the machine-translated sentence into an adequate and fluent translation. In recent years, QE gained increasing interest in the MT community as a possible way to: decide whether a given translation is good enough for publishing as is, inform readers of the target language only whether or not they can rely on a translation, filter out sentences that are not good enough for post-editing by professional translators, or select the best translation among options from multiple MT or translation memory systems.

So far, despite its many possible applications, QE research has been mainly conducted in controlled laboratory testing scenarios that disregard some of the possible challenges posed by real working conditions. Indeed, the large body of research resulting from three editions of the shared QE task organized within the yearly Workshop on Machine Translation (WMT

[1]The HTER (Snover et al., 2006) measures the minimum edit distance between the MT output and its manually post-edited version. 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.

Proceedings of the Workshop on Interactive and Adaptive Machine Translation, pages 9–19
AMTA Workshop. Vancouver, Canada. September 22, 2014

(Callison-Burch et al., 2012; Bojar et al., 2013, 2014)) has relied on simplistic assumptions that do not always hold in real life. These assumptions include the idea that the data available to train QE models is: (i) large (WMT systems are usually trained over datasets of 800 or more instances for training) and (ii) training and test are sampled from the same distribution (WMT training and test sets are drawn from the same domain and are uniformly distributed).

In order to investigate the difficulties of training a QE model in realistic scenarios where such conditions might not hold, in this paper we approach the task in situations where: (i) scarce amounts of training data are available and (ii) training instances come from different domains. In these two particularly challenging contexts from the machine learning perspective, we investigate the potential of online and multitask learning methods, comparing them with the batch methods currently used. Our experiments are carried out over datasets of three different domains with 1,000 tuples of source, machine translated and post-edited sentences each.

To the best of our knowledge, this represents the first attempt to compare the two learning paradigms in the MT QE field and within the same experimental setting. The analysis of the results achieved with the two methods yields interesting findings that suggest, as a promising research avenue, the possibility to exploit their complementarity.

2 Related Work

State-of-the-art in QE explores different supervised linear or non-linear learning methods for regression or classification such as, among others, support vector machines (SVM), different types of decision trees, neural networks, elastic-net, gaussian processes, naive bayes (Specia et al., 2009; Buck, 2012; Beck et al., 2013; C. de Souza et al., 2014a). Another aspect related to the learning methods that has received attention is the optimal selection of features in order to overcome issues related with the high-dimensionality of the feature space (Soricut et al., 2012; C. de Souza et al., 2013; Beck et al., 2013).

Despite constant improvements, such learning methods have limitations. The main one is that they assume that both training and test data are independently and identically distributed. As a consequence, when they are applied to data from a different distribution or domain they show poor performance (C. de Souza et al., 2014b). This limitation harms the performance of QE systems for several real-world applications, such as computer-assisted translation (CAT) environments. Advanced CAT systems currently integrate suggestions obtained from MT engines with those derived from translation memories (TMs). In such framework, the compelling need to speed up the translation process and reduce its costs by presenting human translators with good-quality suggestions raises interesting research challenges for the QE community. In such environments, translation jobs come from different domains that might be translated by different MT systems and are routed to professional translators with different idiolect, background and quality standards (Turchi et al., 2013). Such variability calls for flexible and adaptive QE solutions by investigating two directions: (i) modeling translator behaviour (Cohn and Specia, 2013; Turchi et al., 2014) and (ii) maximize the learning capabilities from all the available data (C. de Souza et al., 2014b).

In this study we experiment with the approaches proposed to address directions (i) and (ii) under the same conditions and evaluate their performance. We use the best learning algorithm presented by C. de Souza et al. (2014b) and the online learning protocol for QE presented in Turchi et al. (2014) and compare their results. In our experiments we use more data than both studies to perform our experiments (1000 data points) for three different domains and compare both methods with each other as well as with competitive baselines.

3 Adaptive MT QE

Multitask Learning (MTL). In MTL different tasks (domains in our case) are correlated via a certain structure. Examples of such structures are the hidden layers in a neural network (Caruana, 1997), shared feature representations (Argyriou et al., 2007), among others. This common structure allows for knowledge transfer among tasks and has been demonstrated to improve model generalization over single task learning (STL) for different problems in different areas. Under this scenario, several assumptions can be made about the relatedness among the tasks, leading to different transfer structures.

In MTL there are T tasks and each task $t \in T$ has m training samples $\{(x_1^{(t)}, y_1^{(t)}), \ldots, (x_m^{(t)}, y_m^{(t)})\}$, with $x_i^{(t)} \in \mathbb{R}^d$ where d is the number of features and $y_i^{(t)} \in \mathbb{R}$ is the output (the response variable or label). The input features and labels are stacked together to form two different matrices $X^{(t)} = [x_1^{(t)}, \ldots, x_m^{(t)}]$ and $Y^{(t)} = [x_1^{(t)}, \ldots, x_m^{(t)}]$, respectively. The weights of the features for each task are represented by W, where each column corresponds to a task and each row corresponds to a feature.

$$\min_W \sum_{t=1}^{T} ||(W^{(t)}X^{(t)} - Y^{(t)})||_2^2 + \lambda_l ||L||_* + \lambda_s ||S||_{1,2} \text{ subject to: } W = L + S \qquad (1)$$

where $||S||_{1,2}$ is the group regularizer that induces sparsity on the tasks and $||L||_*$ is the trace norm.

The key assumption in MTL is that tasks are related in some way. However, this assumption might not hold for a series of real-world problems. In situations in which tasks are not related a negative transfer of information among tasks might occur, harming the generalization of the model. One way to deal with this problem is to: (i) group related tasks in one structure and share knowledge among them, and (ii) identify irrelevant tasks maintaining them in a different group that does not share information with the first group. This is the idea of robust MTL (RMTL henceforth). The algorithm approximates task relatedness via a low-rank structure and identifies outlier tasks using a group-sparse structure (column-sparse, at task level).

RMTL is described by Equation 1. It employs a non-negative linear combination of the trace norm (the task relatedness component L) and a column-sparse structure induced by the $l_{1,2}$-norm (the outlier task detection component S). If a task is an outlier it will have non-zero entries in S. Both L and S are matrices that represent T tasks in the columns and d features in the rows, like W. The trace norm is the sum of singular values computed over the feature weights and given by $||L||_* = \sum_{i=1}^{r} \sigma_i(L)$ where $\{\sigma_i\}_{i=1}^{r}$ is the set of non-zero singular values in non-increasing order and $r = rank(L)$. The $l_{1,2}$-norm is given by $||S||_{1,2} = \sum_{t=1}^{T} ||s_t||_2$ where s_t is the column representing task t and $||.||_2$ is the l_2-norm (also known as the Euclidean norm of a vector).

Online Learning. In the online framework, differently from the batch mode, the learning algorithm sequentially processes a sequence of n instances $X = x_1, x_2, \ldots, x_n$, returning a prediction $\hat{y}_t = w_t \cdot x_t$ as output at each step. A loss function between $\hat{y}_t$ and the true label y_t obtained as feedback is used by the algorithm to update the model. In our experiments we aim to predict the quality of the suggested translations in terms of HTER. To this aim we use online learning, in particular, the passive aggressive learning method, which is defined as follows (adapted from Crammer et al. (2006)):

- Receive X, the vector of features extracted from sentence (*source, target*) pairs;

- Predict $\hat{y}_t = w_t \cdot x_t$. The prediction $\hat{y}_t$ is the estimated HTER score for instance t and w_t is the incrementally learned weights feature vector;
- Receive label $y_t = [0, 1]$. The observed HTER score;
- Compute loss l_t for the current instance t. The loss is 0 if $|w \cdot x - y| < \epsilon$ and $|w \cdot x - y| - \epsilon$ otherwise. This is know as the ϵ-insensitive loss;
- Update w according to $w_{t+1} = w_t + \text{sign}(y_t - \hat{y}_t)\tau_t x_t$ where τ_t is given by $l_t / ||x_t||^2$.

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.

4 Experimental Setting

In this section we describe the data used for our experiments, the features extracted, the set up of the learning methods, the baselines used for comparison and the evaluation of the models. The goal of our experiments is to show that the methods presented in Section 3 outperform competitive baselines and standard QE learning methods that are not capable of adapting to different domains. We experiment with three different domains of comparable size and evaluate the performance of the adaptive methods and the standard techniques with different amounts of training data. The RMTL algorithm described in section 3 is trained with the Malsar toolkit implementation (Zhou et al., 2012). The online learning algorithm is trained using the AQET toolkit[2] (Turchi et al., 2014). The hyper-parameters for both RMTL and PA algorithms are optimized using 5-fold cross-validation in a grid search procedure over the training data.

Data. Our experiments focus on the English-French language pair and encompass three very different domains: newswire text (henceforth News), transcriptions of Technology Entertainment Design talks (TED) and Information Technology manuals (IT). Such domains are a challenging combination for adaptive systems since they come from very different sources spanning speech and written discourse (TED and News/IT, respectively) as well as a very well defined and controlled vocabulary in the case of IT.

Each domain is composed of 1000 tuples formed by the source sentence in English, the French translation produced by an MT system and a human post-edition of the translated sentence. For each pair (translation, post-edition) we use as labels the HTER score computed with TERCpp[3]. For the three domains we use 70% of the data for training (700 instances) and 30% of the data for testing (300 instances). The limited amount of instances for training contrasts with the 800 or more instances of the WMT evaluation campaigns and is closer to real-world applications where the availability of large and representative training sets is far from being guaranteed (e.g. the CAT scenario).

The TED talks domain is formed by subtitles of several talks in a range of topics presented in the TED conferences. The complete dataset has been used for MT and automatic speech recognition systems evaluation within the International Workshop on Spoken Language Translation (IWSLT). The News domain is formed by newswire text used in WMT translation campaigns and covers different topics. The sentence tuples for TED and News domains are taken from the Trace corpus[4]. The translations were generated by two different MT systems, a state-of-the-art phrase-based statistical MT system and a commercial rule-based system. Furthermore, the translations were post-edited by up to four different translators, as described in

[2] http://hlt.fbk.eu/technologies/aqet

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

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

(Wisniewski et al., 2013). The IT texts come from a software user manual translated by a statistical MT system based on the state-of-the-art phrase-based Moses toolkit (Koehn et al., 2007) trained on about 2M parallel sentences. The post-editions were collected from one professional translator operating on the Matecat[5] (Federico et al., 2014) CAT tool in real working conditions.

Features. For all the experiments we use the same feature set composed of 17 features proposed in Specia et al. (2009) and extracted with the QuEst feature extractor (Specia et al., 2013; Shah et al., 2014). The set is formed by features that model the complexity of translating the source sentence (e.g. the average source token length or the number of tokens in the source sentence), and the fluency of the translated sentence produced by the MT system (e.g. the language model probability of the translation). The decision to use this feature set is motivated by the fact that it demonstrated to be robust across language pairs, MT systems and text domains (Specia et al., 2009).

Baselines. As a term of comparison, in our experiments we consider two baselines. A simple to implement but difficult to beat baseline when dealing with regression on tasks with different distributions is to compute the mean of the training labels and use it as the prediction for each testing point (Rubino et al., 2013). In our experiments we compute the mean HTER of the training instances of each domain and use it as prediction for each instance of the in-domain test set. Hereafter we refer to this baseline as μ.

Since supervised domain adaptation techniques should outperform models that are trained only on the available in-domain data, we also use as baseline the regressor built only on the available in-domain data (SVR in-domain). The in-domain baseline system is trained on the feature set described earlier in Section 4 with an SVM regression (SVR) method using the implementation of Scikit-learn (Pedregosa et al., 2011). The radial basis function (RBF) kernel is used for all experiments. The hyper-parameters of the model are optimized using randomized search optimization process with 50 iterations as described in Bergstra and Bengio (2012) and used previously for QE in C. de Souza et al. (2013).

Evaluation. The accuracy of the models is evaluated with the mean absolute error (MAE), which was also used in previous work and in the WMT QE shared tasks (Bojar et al., 2013). MAE is the average of the absolute difference between the prediction $\hat{y}_i$ of a model and the gold standard response y_i (Equation 2). As it is an error measure, lower values indicate better performance.

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|\hat{y_i} - y_i| \tag{2}$$

In our experiments we compare multiple hypothesis among each other (μ, SVR in-domain, RMTL and PA) across different training sets sizes. Given these requirements we need to perform multiple hypothesis tests instead of paired tests. It has been shown in Demšar (2006) that for comparisons of multiple machine learning models, the recommended approach is to use a non-parametric multiple hypothesis test followed by a post-hoc analysis that compares each pair of hypothesis. For computing the statistical significance we use the Friedman test (Friedman, 1937, 1940) followed by a post-hoc analysis of the pairs of regressors using Holm's procedure (Holm, 1979) to perform the pairwise comparisons when the null hypothesis is rejected. All tests for both Friedman and post-hoc analysis are run with $\alpha = 0.05$. For more details about these methods, we refer the reader to Demšar (2006); Garcia and Herrera (2008) which provide a complete review about the application of multiple hypothesis testing to machine learning methods.

[5] www.matecat.com

5 Results and Discussion

In this section we describe the experiments made with the models described in Section 3 and discuss the results. As shown in previous work, using single task learning algorithms with in-domain training data on a cross-domain setting leads to poor results (C. de Souza et al., 2014b). In our experiments we run the baselines described in Section 4 and the methods described in Section 3 on different amounts of training data, ranging from 70 to 700 instances (10% and 100% of the training data, respectively). The motivation is to verify how much training data is required by the MTL and online methods to outperform the baselines for a target domain. It is important to remark that MTL approach use the training data of the multiple domains to jointly learn the models for each domain whereas the online learning protocol used here only uses in-domain data.

Algorithm	**20%**	**50%**	**100%**
TED			
μ	0.2088	0.2091	0.2066
SVR in-domain	0.2063	0.2083	0.2036
RMTL	**0.1962**	0.2019	0.1990
PA	0.2036	**0.1977**	**0.1943**
News			
μ	**0.1384**	**0.1386**	**0.1384**
SVR in-domain	0.1533	0.1484	0.1460
RMTL	0.1492	0.1446	0.1433
PA	0.2305	0.2218	0.2200
IT			
μ	0.2125	0.2128	0.2125
SVR in-domain	0.2114	0.1959	0.1863
RMTL	0.2082	0.2041	0.2023
PA	**0.1917**	**0.1877**	**0.1858**

Table 1: Average performance of 30 runs of the algorithms on different train and test splits with 20, 50 and 100 percent of training data. The average scores reported are the MAE.

Table 1 presents the results for the three domains with models trained on 20, 50 and 100% of the training data (140, 350 and 700 instances, respectively). Each method was run on 30 different train/test splits of the data in order to account for the variability of points in each split. Results for PA are statistically significant w.r.t both baselines for IT ($p \leq 0.016667$) and TED ($p \leq 0.025$) but not for News. Results for RMTL are statistically significant w.r.t both baselines for TED ($p \leq 0.025$) and they are not statistically significant for the other two domains.

Both the RMTL and PA algorithms outperform the SVR in-domain and μ baselines for the TED and IT domains with different amounts of training data. For TED, with as much as 20% of the training data, RMTL outperforms SVR in-domain (the best performing baseline) by around 4.89%. Training the models with 50 and 100% of the training data PA outperforms all other models and in particular the SVR in-domain by 5 and 4.5%, respectively. The learning curves of all algorithms for the TED domain are shown in Figure 1. The learning curves show that RMTL does very well with very little training data whereas PA performs better as we add more training data.

Similarly, for the IT domain, PA presents the best performance outperforming the best performing baselines when trained with 20, 50% of the training data by 9.13 and 4.15% and a very similar performance when trained with 100% of the training data. It is important to notice

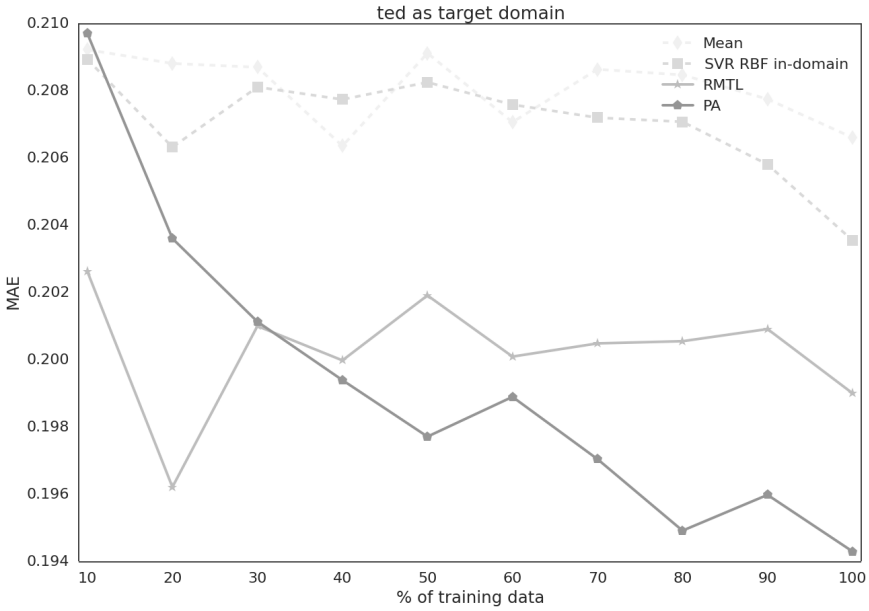

Figure 1: Learning curves for the TED domain.

that PA learns in an online fashion over the test data in addition to the training data, as opposed to the other algorithms presented here.

For the News domain, RMTL outperforms SVR in-domain but it is outperformed by the μ baseline. One indication that explains why the μ baseline is hard to beat are the distributions of the HTER scores for the News domain (Table 2). Whereas the three domains present similar means, the standard deviation of the HTER scores of News is smaller than for IT and TED. This indicates that every point in the News domain is closer to the mean than in the other two domains.

Domain	**Mean**	**Std**
IT	0.3620	0.2653
TED	0.3396	0.2446
News	0.3737	0.1859

Table 2: Mean and standard deviation of the distributions of HTER scores for TED, IT and News domains.

The distribution of data for News shows that different things might be happening in this data, such as: (i) the different MT systems that compose this domain produce translations of similar quality (around the mean of 0.3737); (ii) the difficulty of translating the sentences is homogeneous and (iii) the post-editors tend to agree more. The kernel density estimation of the labels for the three domains is shown in Figure 2. The News domain presents only one maxima and has a different shape than the other two domains that present at least two other maximum, indicating that TED and IT are more alike in terms of label distributions with respect to the News domain.

The results show that both RMTL and PA improve over in-domain single-task learning on different domains. The MTL method used in our experiments is capable of transferring knowledge from different domains whereas the online learning method is capable of training

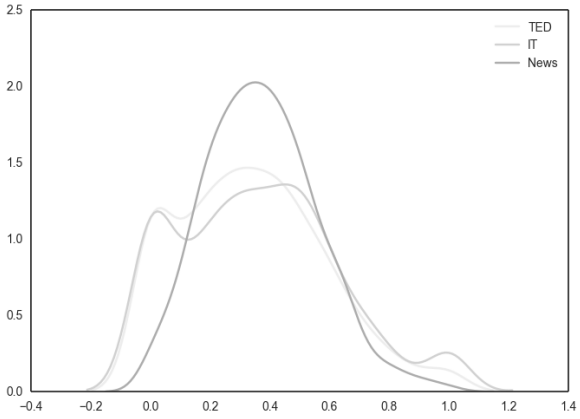

Figure 2: Kernel density estimation of HTER scores for TED, IT and News domains (1000 instances).

incremental models that can leverage also the test data. Interestingly, the results achieved with the two approaches suggest that they can complement each other if combined. Indeed, online MTL would make it possible to leverage the positive characteristics of both methodologies for both for batch and online learning applications of MT QE.

For example, in an application like MT QE for the CAT scenario, we can have an online MTL method that uses the MTL transfer capability to learn more robust models that can continuously evolve over time accounting for knowledge acquired from post-editors work (the same setting proposed by Turchi et al. (2014)). Likewise, online MTL can be used to adapt to new domains (different post-editors, MT systems and text genres) in scenarios in which only a very limited amount of training labels is available (the scenario described in C. de Souza et al. (2014b)). An interesting characteristic of the results presented in this work is that both online and MTL learning require fewer training points than single-task batch learning methods (as shown in Figure 1). A combination of both techniques might hence lead to further reduction on the amount of training data needed, depending on the data.

This motivates, as an interesting line of future work, the combination of the two methods. We believe that significant improvements towards the application of QE in real-world scenarios could be reached by leveraging the adaptation capability of MTL and the incremental learning capability of online methods.

6 Conclusion

In this work we presented an evaluation of multitask and online methods capable of learning models across different domains for MT QE. In our experiments we worked close to a real world scenario in which the training data is formed by translations generated by different MT systems, the translations are post-edited by different translators and the texts come from differente text genres. We compared one multitask (robust MTL) and one online learning method (passive agressive) with two different competitive baselines.

The results of our experiments show that both MTL and online learning methods produce better models than single task learning batch models under such difficult conditions. Furthermore, this comparison opens an interesting research direction for MT QE that is to explore online multitask learning methods. Such methods join the information transfer capability intrinsic to MTL methods with the incremental learning capabilities of online learning methods, enabling better adaptation capabilities in MT QE applications that require online or batch learning.

References

Argyriou, A., Evgeniou, T., and Pontil, M. (2007). Multi-task feature learning. In *Advances in neural information processing systems*, volume 19.

Beck, D., Shah, K., Cohn, T., and Specia, L. (2013). SHEF-Lite: When less is more for translation quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 337–342.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Eighth Workshop on Statistical Machine Translation*, pages 1–44.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58.

Buck, C. (2012). Black Box Features for the WMT 2012 Quality Estimation Shared Task. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 91–95.

C. de Souza, J. G., Buck, C., Turchi, M., and Negri, M. (2013). FBK-UEdin participation to the WMT13 Quality Estimation shared-task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 352–358.

C. de Souza, J. G., González-Rubio, J., Buck, C., Turchi, M., and Negri, M. (2014a). FBK-UPV-UEdin participation in the WMT14 Quality Estimation shared-task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 322–328.

C. de Souza, J. G., Turchi, M., and Negri, M. (2014b). Machine Translation Quality Estimation Across Domains. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers.*, pages 409–420.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 10–51, Montreal, Canada. Association for Computational Linguistics.

Caruana, R. (1997). Multitask Learning. *Machine learning*, 28(28):41–75.

Cohn, T. and Specia, L. (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.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, 7:1–30.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., Cattelan, A., Farina, A., Lupinetti, D., Martines, A., Massidda, A., Schwenk, H., Barrault, L., Blain, F., Koehn, P., Buck, C., and Germann, U. (2014). The Matecat Tool. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132.

Friedman, M. (1937). The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200):675–701.

Friedman, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

Garcia, S. and Herrera, F. (2008). An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9:2677–2694.

Holm, S. (1979). A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6(2):pp. 65–70.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zenz, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL 2007 Demo and Poster Sessions*, number June, pages 177–180.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rubino, R., de Souza, J. G. C., and Specia, L. (2013). Topic Models for Translation Quality Estimation for Gisting Purposes. In *Machine Translation Summit XIV*, pages 295–302.

Shah, K., Turchi, M., and Specia, L. (2014). An Efficient and User-friendly Tool for Machine Translation Quality Estimation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Association for Machine Translation in the Americas*.

Soricut, R., Bach, N., and Wang, Z. (2012). The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 145–151.

Specia, L., Cancedda, N., Dymetman, M., Turchi, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proceedings of the 13th Annual Conference of the EAMT*, number May, pages 28–35.

Specia, L., Shah, K., de Souza, J. G. C., and Cohn, T. (2013). QuEstA translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 79–84.

Turchi, M., Anastasopoulos, A., de Souza, J. G. C., and Negri, M. (2014). Adaptive Quality Estimation for Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Turchi, M., Negri, M., and Federico, M. (2013). Coping with the subjectivity of human judgements in mt quality estimation. In *Eighth Workshop on Statistical Machine Translation (WMT)*, pages 240–251.

Wisniewski, G., Singh, A. K., Segal, N., and Yvon, F. (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.

Zhou, J., Chen, J., and Ye, J. (2012). MALSAR: Multi-tAsk Learning via StructurAl Regularization.