Machine Translation Quality Estimation: Applications and Future Perspectives

Lucia Specia and Kashif Shah

Abstract Predicting the quality of machine translation (MT) output is a topic that has been attracting significant attention. By automatically distinguishing bad from good quality translations, it has the potential to make MT more useful in a number of applications. In this chapter we review various practical applications where quality estimation (QE) at sentence level has shown positive results: filtering low quality cases from post-editing, selecting the best MT system when multiple options are available, improving MT performance by selecting additional parallel data, and sampling for quality assurance by humans. Finally, we discuss QE at other levels (word and document) and general challenges in the field, as well as perspectives for novel directions and applications.

Keywords Translation quality assessment · Principles to practice · Translation errors · Translation models · Post-editing effort · Statistical machine translation · Machine translation system ranking · Machine translation system selection · Quality estimation

1 Introduction

Machine Translation (MT) systems are becoming widely adopted both for gisting purposes and to produce professional quality translations. However, the quality of automatic translation is still below an acceptable level in many cases. This makes evident the need for automatic metrics for predicting the quality of a translated segment. These metrics are referred to as Quality Estimation (QE). The goal of QE is to provide an estimate on how good or reliable a translated text is without access to

L. Specia (✉)
Department of Computer Science, University of Sheffield, Sheffield, UK
e-mail: l.specia@sheffield.ac.uk

K. Shah
eBay Research, San Jose, CA, USA
e-mail: skshah@ebay.com

J. Moorkens et al. (eds.), *Translation Quality Assessment*, Machine Translation: Technologies and Applications 1, https://doi.org/10.1007/978-3-319-91241-7_10

reference (human) translations. This is, therefore, different from standard evaluation methods where the task is to compare system translations with their reference counterparts, which are generally created by linguistic experts with knowledge of the languages involved. While MT systems can be evaluated using reference datasets and their average quality can be measured on those data points, it is known that the quality on individual inputs can vary considerably depending on a number of factors. QE is not aimed at estimating overall MT system performance, but rather performance on individual translations. The main motivation is to make applications more useful in real world settings, where information on the quality of each output is needed and reference outputs are not available. QE is aimed at MT systems in use. As such, QE metrics have several applications in the context of MT, which we discuss in this chapter. QE approaches also have the advantage of allowing for a flexible modelling of the concept of quality, depending, among other things, on the user or intended use of the MT system's output.

Work in QE for MT started in the early 2000s. Inspired by the confidence scores used in Speech Recognition, initial research explored information coming from the statistical MT models, such as word translation probabilities, language model scores and other statistical indicators. Back then it was called *confidence estimation*, a narrower term that reflects the fact that the statistical indicators used are related to the confidence of the MT system in the translation produced. A 6-week workshop on the topic at Johns Hopkins University in 2003 (Blatz et al. 2004) set as its goal the estimation of automatic metrics such as BLEU (Papineni et al. 2002) and WER (Word Error Rate) (Levenshtein 1966). These metrics are difficult to interpret, particularly at the sentence level. Given the metrics used and the fact that the overall quality of MT was considerably lower at the time, pinpointing the very few good quality MT segments was a much harder problem. As a consequence, results of multiple experiments proved unsuccessful. Also, no software or datasets were made available after the workshop.

A new surge of interest in the field started around 2010, motivated by the widespread use of MT systems in the translation industry, as a consequence of better translation quality, more user-friendly tools, and higher demand for translation. In order to improve the utility of MT in this scenario, a quantification of the quality of translated segments is needed. In a way, this quantification can be thought of as similar to "fuzzy match scores" from translation memory (TM) systems. However, QE work addresses this problem using more complex metrics that go beyond matching the source segment against previously translated data. In addition, QE can be useful for users other than professional translators, such as end-users reading translations for gisting, particularly those who cannot read the source language. Recent work focuses on estimating more interpretable metrics where "quality" is defined according to the task at hand, such as post-editing, gisting, sampling, etc. (see also Sect. 3.3 of Way in this volume).

A number of positive results have been reported. Examples include improving post-editing efficiency by filtering out low-quality segments which would require more effort or time to be corrected than translating from scratch (Specia et al. 2009; Specia 2011), selecting high-quality segments to be published as they are, without

post-editing (Soricut and Echihabi 2010), selecting a translation from either an MT system or a TM for post-editing (He et al. 2010), selecting the best translation from multiple MT systems (Specia et al. 2010; Avramidis 2013), and highlighting sub-segments that need revision (Bach et al. 2011; Quang et al. 2014).

QE is generally addressed as a supervised machine learning task using a variety of algorithms to induce models from examples of translations described through a number of features and annotated for quality. For an overview of various algorithms and features we refer the reader to the WMT12–16[1] shared task on QE (Callison-Burch et al. 2012b; Bojar et al. 2013, 2014, 2015, 2016). Most of the research work lies on deciding which aspects of quality are more relevant for a given task and designing feature extractors for them. These can go from simple, language-independent features, to advanced, linguistically-motivated features. They can include features that rely on information from the MT system that generated the translations, as well as features that are independent of the way translations were produced. While simple features such as counts of tokens and language model scores can be easily extracted, feature engineering for more advanced and useful information can be very labour- and resource-intensive. Different feature sets are necessary for different language pairs or for optimisation against specific quality scores, where translations are created with different applications in mind (e.g. post-editing time vs translation adequacy).

In this chapter we focus on sentence-level experiments and results for what we believe are some of the most promising and practical applications of QE to date. Each of these applications has been developed around a specific objective:

- Estimate how much effort will be needed to post-edit a segment.
- Select among alternative translations produced by different MT systems.
- Decide whether the translation can be used for self-learning of MT systems.
- Select samples of translations for manual inspection.

In what follows, we first explain the general experimental settings, including features and learning algorithms, for the various QE applications to be covered (Sect. 2). For consistency purposes, across all datasets and applications we use the same feature sets and learning algorithms where possible. In the remainder of the chapter (Sects. 3, 4, 5, and 6), we present our work on the various above-mentioned applications and benchmark the results on freely available datasets.

2 Experimental Settings

Our experiments with all applications of QE are performed using QuEst++ (Specia et al. 2013, 2015a)- an open source framework for quality estimation containing

[1]The Workshop (now Conference) on Machine Translation runs annual competitive MT system evaluations for a range of tasks. See http://www.statmt.org/wmt17/ for the latest in the series.

a number of features, covering complexity, adequacy and, fluency of segments using a machine-learning algorithm. Amongst the learning algorithms available in QuEst++, we choose the Support Vector Regression algorithm given its promising performance in previous work.

2.1 Support Vector Regression (SVR)

SVR (Chang and Lin 2011) is the most commonly used algorithm for sentence-level QE. This is a very popular and powerful machine-learning algorithm used when the score to predict is numeric and distributed over an ordinal or continuous range, for example, post-editing time or Likert scores in {1,5}. To make our results comparable with most previous work, we use a kernel version of this algorithm with a radial basis function (RBF) kernel, which has been shown to perform very well in this task (Callison-Burch et al. 2012a). Kernel parameters are optimised using grid search with five-fold cross-validation.

2.1.1 Feature Sets

As feature sets, we consider the following for the sentence-level tasks:

- **BL**: 17 simple but effective baseline features that perform well across languages and were used as baseline in the WMT12–16 shared tasks on QE.
- **AF**: All features available in QuEst++ across the datasets, for example, 80 language and MT system-independent features for sentence-level prediction.

2.1.2 Evaluation Metrics

We use two main error metrics to evaluate our sentence level regression models: Mean Absolute Error (**MAE**), shown in Eq. 1 and Root Mean Squared Error (**RMSE**), shown in Eq. 2.

$$\text{MAE} = \frac{\sum_{i=1}^{N} |H(s_i) - V(s_i)|}{N} \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (H(s_i) - V(s_i))^2}{N}} \tag{2}$$

where:

$N = |S|$ is the number of test instances
$H(s_i)$ is the predicted score for s_i
$V(s_i)$ is the human score for s_i

For the classification results, we use the standard **Accuracy** metric.

In addition, we use application-specific metrics, such as BLEU for MT system evaluation based on data selected through QE in Sect. 5.

2.1.3 Baselines

We compared our regression results with the **Mean** score, i.e., the score obtained by assigning the mean value of the training set labels to all test set instances. For classification experiments, we compared our results to the **Majority Class** score, i.e., the score obtained by assigning the most frequent label of the training set to all test set instances.

3 QE for Predicting Post Editing Effort

In this section we focus on QE for outbound purposes, i.e. a dissemination scenario. In this scenario, a judgement on the quality of translations has to take into account both the fluency and adequacy of such translations, and in some cases, it has to conform to style guides. MT is followed by manual post-editing and/or revision by human translators to achieve publishable quality. Our objective is to support human translators by designing QE methods to distinguish translations that are good enough for post-editing from those that are too bad, and so should be translated from scratch. A common distinction includes at least three levels of "effort": (i) translations that are good enough to be left untouched by human post-editors (but possibly still revised); (ii) translations which require further effort (post-editing) to be published; and (iii) translations that should better be discarded, as they require more effort from human translators to correct them than what is involved in manual translation from scratch. In the following we benchmark QE on various datasets annotated for post-editing effort in different ways.

3.1 Datasets

All datasets used in the experiments are available for download.[2] Statistics about these datasets are shown in Table 1. They differ in size, language pair and label for post-editing effort.

- **WMT14 (Task-1.1)** English-Spanish news sentence translations. The dataset contains news source sentences and their human translations, as well as three

[2] http://www.dcs.shef.ac.uk/~lucia/resources.html

Data	Languages	Training	Test	Label
WMT14	en-es	3,816	600	PEE 1–3
WMT12	en-es	1,832	422	PEE 1–5
EAMT11	en-es	900	64	PEE 1–4
EAMT11	fr-en	2,300	225	PEE 1–4
EAMT09-s_1	en-es	3,095	906	PEE 1–4
EAMT09-s_2	en-es	3,095	906	PEE 1–4
EAMT09-s_3	en-es	3,095	906	PEE 1–4
EAMT09-s_4	en-es	3,095	906	PEE 1–4

Table 1 Language pairs, number of training and test sentences and type of label in the datasets for the post-editing effort prediction

versions of MT output: by a statistical MT (SMT) system, a rule-based MT (RBMT) system and a hybrid system. Each translation was labelled by professional translators with 1–3 (lowest-highest) scores for perceived post-editing effort.

- **WMT12** English-Spanish sentence translations produced by a phrase-based (PB) Moses "baseline" system (Koehn et al. 2007),[3] and judged for post-editing effort in 1–5 (highest-lowest), taking a weighted average of three annotators.
- **EAMT11** English-Spanish (EAMT11 (en-es)) and French-English (EAMT11 (fr-en)) sentence translations produced by a PBSMT "baseline" Moses system and judged for post-editing effort in 1–4 (highest-lowest).
- **EAMT09** English sentences translated by four SMT systems into Spanish and scored for post-editing effort in 1–4 (highest-lowest). Systems are denoted by s_1–s_4.

3.2 Feature Selection

Given the large number of features available, it is often beneficial to select only the most relevant for the dataset at hand. We performed feature selection using Gaussian Processes, which has proved very effective in previous work (Shah et al. 2015). Gaussian Processes (GPs) (Rasmussen and Williams 2006) are a Bayesian non-parametric machine learning framework considered the state-of-the-art for regression. GPs have been used successfully for MT quality prediction (Shah et al. 2013), among other tasks. We use GPs with radial basis function (RBF) with automatic relevance determination, as in (3).

[3] http://www.statmt.org/moses/?n=Moses.Baseline

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_i^D \frac{x_i - x_i'}{l_i}\right) \quad (3)$$

where the $k(\mathbf{x}, \mathbf{x}')$ is the kernel function between two data points $\mathbf{x}$ and $\mathbf{x}'$, and D is the number of features; σ_f and $l_i \geq 0$ are the kernel hyper-parameters, which control the covariance magnitude and the *length scales* of variation in each dimension, respectively. This is closely related to the RBF kernel used with SVR, except that each feature is scaled independently of the others, i.e. $l_i = l$ for SVR, while we allow for a vector of independent values. Following standard practice we also include an additive white-noise term in the kernel with variance σ_s^2. The kernel hyper-parameters $(\sigma_f, \sigma_n, \mathbf{l})$ are learned via gradient descent with a maximum of 100 iterations and cross-validation on the training set.

Feature selection is done by fitting per-feature RBF widths (also known as the *automatic relevance determination* kernel). The learned length scale hyper-parameters can be interpreted as the per-feature RBF widths which encode the importance of a feature: the narrower the RBF (the smaller the l_i), the more important a change in the feature value is to the model prediction. Therefore, the outcome of a model trained using GPs can be viewed as a list of features ranked by relevance, and this information can be used for feature selection by discarding the lowest-ranked (least useful) features. GPs on their own do not provide a cut-off point on this ranked list of features; instead this needs to be determined by evaluating loss on a separate dataset to determine the optimal number of features.

3.3 Results

The error scores for all datasets using SVR as the learning algorithm are reported in Table 2. It can be seen that adding more features (systems **AF**) improves the results in most cases as compared to the baseline system with 17 features **BL**. However, in most cases the improvements are not significant. This behaviour is to be expected as adding more features may bring more relevant information, but at the same time it makes the representation more sparse and the learning prone to overfitting.

Our experiments with feature selection using GPs led to significant further improvements in all cases. The **FS(GP)** figures are produced from selecting the fixed 17 top-ranked features (i.e. the same number as that of the baseline features). **FS(GP)** outperforms other systems despite using considerably fewer features (17 in all datasets). These are very promising results, as they show that it is possible to reduce the resources and overall computational complexity for training the models, while achieving similar or better performance.

Dataset	System	# Features	MAE	RMSE
EAMT11(en-es)	Mean	–	0.6027	0.7314
	BL	17	0.4867	0.6288
	AF	80	**0.4696**	0.5438
	FS(GP)	17	**0.4397**	**0.5224**
EAMT11(fr-en)	Mean	–	0.5411	0.6927
	BL	17	0.4387	0.6357
	AF	80	0.4275	0.6211
	FS(GP)	17	**0.4166**	**0.6176**
WMT12	Mean	–	0.8278	0.9898
	BL	17	0.6802	0.8192
	AF	80	0.6703	0.8373
	FS(GP)	17	**0.6224**	**0.7645**
WMT14	Mean	–	0.4585	0.6678
	BL	17	0.5241	0.6591
	AF	80	0.4896	0.6349
	FS(GP)	17	**0.4850**	**0.6331**
EAMT09-s_1	Mean	–	0.5382	0.7092
	BL	17	0.5294	0.6643
	AF	80	0.5235	0.6558
	FS(GP)	17	0.5045	**0.6392**
EAMT09-s_2	Mean	–	0.6854	0.7926
	BL	17	0.4604	0.5856
	AF	80	0.4734	0.5973
	FS(GP)	17	0.4514	**0.5735**
EAMT09-s_3	Mean	–	0.6753	0.7751
	BL	17	0.5321	0.6643
	AF	80	0.5437	0.6827
	FS(GP)	17	0.5130	0.6572
EAMT09-s_4	Mean	–	0.4990	0.6112
	BL	17	0.3583	0.4953
	AF	80	0.3569	0.5000
	FS(GP)	17	0.3383	0.4811

Table 2 Results with black-box features and SVR as learning algorithm. For each dataset, bold-faced figures are significantly better than all others (paired t-test with $p \leq 0.05$)

4 QE for System Selection

In this section the goal is to model quality estimation by contrasting the output of several translation sources for the same input sentence. The outcome of this process is a ranking of alternative translations based on their predicted quality. For the system selection application, we are more interested in correctly ranking the best

translation at the top, as opposed to obtaining a complete ranking of all alternative translations. This top-ranked translation could either be provided to a human post-editor for revision, or used as is.

For all experiments, we use the features and settings for these experiments as those described in Sect. 2. We treat the problem as a machine-learning regression task, where SVR models are trained to estimate a continuous score within {1,3}. In the first round of experiments (Sect. 4.2) we evaluate different settings of these models following a standard regression setting, while in the second round of experiments we apply the models to select a given translation option for each segment and evaluate the outcome in terms of document-level translation quality (Sect. 4.3).

4.1 Datasets

The datasets used here are a superset of the **WMT14** dataset described in the previous section. They consist of news domain texts in four language pairs (Table 3): English-Spanish (**en-es**), Spanish-English (**es-en**), English-German (**en-de**), and German-English (**de-en**). For each language pair, the data contains a different number of source sentences and their human translations, as well as 2–3 versions of MT outputs: by an SMT system, an RBMT system and, for en-es/de only, a hybrid system. The translations were produced by top MT systems of each type (SMT, RBMT, and hybrid; hereafter **system2**, **system3**, **system4**) which participated in the translation shared task, plus the professional translation given as reference (**system1**).

Each translation in this dataset has been labelled by a professional translator with {1,3} scores for "perceived" post-editing effort, where:

- **1** = perfect translation, no editing needed.
- **2** = near miss translation: maximum of 2–3 errors, and possibly additional errors that can be easily fixed (capitalisation, punctuation).
- **3** = very low quality translation, cannot be easily fixed.

The distribution of true scores in both training and test sets is given in Figs. 1 and 2, for each language pair, and for each language pair and translation source (MT system or human), respectively.

Languages	# Training Src/Tgt	# Test Src/Tgt
en-es	954/3,816	150/600
en-de	350/1,400	150/600
de-en	350/1,050	150/450
es-en	350/1,050	150/450

Table 3 Number of training and test source (Src) and target (Tgt) sentences in each dataset for the system selection experiments

Fig. 1 Distribution of true scores by language pair

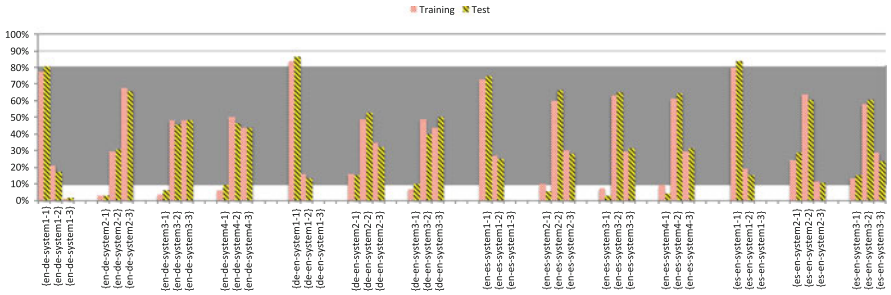


Fig. 2 Distribution of true scores for each MT system and language pair

4.2 Regression Results

For the standard regression evaluation, we compare prediction error for models trained (and tested) on pooled translations from all MT systems (and humans) together (Table 4), versus models trained on each dataset individually, considering two settings at test time:

- The system used to produce the translation is unknown (Table 5 blind setting), and so all models are applied, one by one, to predict the quality of this translation and the average prediction is used as output.
- The system is known and thus the model for the same translation system/human is used for prediction (Table 5 non-blind setting).

These two variants may be relevant depending on the application scenario. We consider a scenario where system identifiers are known by developers at model building time, but unknown at test time, to be very realistic, e.g. if QE is provided as a web service with pre-trained models. In all tables, **Mean** – assigning the mean

	System	# Features	MAE	RMSE
en-de	Mean	–	0.6831	0.7911
	BL	17	**0.6416**	**0.7620**
	AF	80	**0.6303**	**0.7616**
de-en	Mean	–	0.6705	0.7979
	BL	17	**0.6524**	**0.7791**
	AF	80	**0.6518**	**0.7682**
en-es	Mean	–	0.4585	0.6678
	BL	17	0.5240	0.6590
	AF	80	0.5092	0.6442
es-en	Mean	–	0.5825	0.6718
	BL	17	0.5736	0.6788
	AF	80	**0.5662**	**0.6663**

Table 4 SVR to build prediction models for each language pair combination, with all translation sources (including human) pooled together

of the training set labels to all test set instances – represents a strong baseline, given the large variation in scores across MT systems and human translators.

Comparing the two variants of the blind setting (Table 4 – blind training and test – and Table 5, blind test only), we see that pooling the data from multiple translation systems for blind model training leads to significantly better results than training models for individual translation sources but testing them in blind settings. This is likely to be due to the larger quantities of data available in the pooled models. In fact, the best results are observed with **en-es**, the largest dataset overall.

Comparing scores between blind versus non-blind test settings in Table 5, we observe a substantial difference in the scores for each of the individual translation system. This shows that the task is much more challenging when QE models are trained independently but the identifiers of the systems producing the test instances are not known.

There is also a considerable difference in the performance of individual models for different translation systems, which can be explained by the different distribution of scores (and also indicated by the performance of the **Mean** baseline). However, in general the prediction performance of the individual models seems less stable, and even worse than the baseline in several cases. Interestingly, the individual models trained on human translations only (system1) do even worse than individual models for MT systems. This can be an indication that the features used for quality prediction are not sufficient to model human translations.

In all cases, the use of all features (**AF**) instead of baseline features (**BL**) comparable or better results.

	System	# Features	Blind		Non-blind	
			MAE	RMSE	MAE	RMSE
en-de-system1	Mean	–	1.0351	1.2133	0.3552	0.4562
	BL	17	1.0487	1.2348	**0.3350**	0.4540
	AF	80	1.0510	1.2375	**0.3325**	0.4545
en-de-system2	Mean	–	0.7780	0.9339	0.4857	0.5487
	BL	17	**0.7006**	0.9499	**0.3615**	**0.4634**
	AF	80	**0.6924**	**0.9124**	**0.3570**	**0.4644**
en-de-system3	Mean	–	0.7369	0.8426	0.5577	0.6034
	BL	17	**0.6354**	**0.7950**	**0.4535**	**0.5363**
	AF	80	**0.6572**	0.8127	**0.4482**	**0.5245**
en-de-system4	Mean	–	0.7231	0.8215	0.5782	0.6433
	BL	17	**0.6438**	**0.7842**	**0.4912**	**0.5834**
	AF	80	**0.6386**	**0.7905**	**0.4818**	**0.5741**
de-en-system1	Mean	–	0.8594	1.0882	0.2506	0.3409
	BL	17	0.8747	1.1299	**0.2123**	0.3421
	AF	80	0.8747	1.1299	**0.2065**	0.3415
de-en-system2	Mean	–	0.7321	0.8484	0.5412	0.6678
	BL	17	**0.6897**	0.8330	**0.4745**	**0.5931**
	AF	80	0.7122	0.8509	**0.4604**	**0.5850**
de-en-system3	Mean	–	0.8137	0.9253	0.6000	0.6640
	BL	17	**0.7472**	**0.8903**	**0.4965**	**0.6011**
	AF	80	**0.7629**	0.9300	**0.4828**	**0.5901**
en-es-system1	Mean	–	0.8542	0.9923	0.3883	0.4353
	BL	17	0.8956	1.0480	**0.3633**	0.4390
	AF	80	0.8957	1.0480	**0.3519**	0.4381
en-es-system2	Mean	–	0.5567	0.6952	0.4232	0.5314
	BL	17	**0.5275**	0.6827	**0.3812**	**0.4951**
	AF	80	**0.5302**	0.6884	**0.3730**	**0.4893**
en-es-system3	Mean	–	0.5653	0.6998	0.4288	0.5213
	BL	17	**0.5155**	**0.6711**	**0.3821**	**0.4844**
	AF	80	**0.5184**	**0.6704**	**0.3714**	**0.4761**
en-es-system4	Mean	–	0.5573	0.6955	0.4300	0.5321
	BL	17	**0.5103**	**0.6680**	**0.4022**	**0.5162**
	AF	80	**0.5206**	**0.6727**	**0.3902**	**0.5016**
es-en-system1	Mean	–	0.6617	0.8307	0.3026	0.3916
	BL	17	0.6617	0.8307	0.3022	0.3917
	AF	80	0.6617	0.8308	0.3023	0.3915
es-en-system2	Mean	–	0.5637	0.6931	0.4494	0.6027
	BL	17	0.5588	0.7023	**0.4384**	0.6061
	AF	80	0.5567	0.7026	**0.4309**	0.6053
es-en-system3	Mean	–	0.6602	0.8129	0.4720	0.6245
	BL	17	0.7233	0.8621	0.4993	0.6220
	AF	80	0.6973	0.8435	0.4974	0.6198

Table 5 SVR to build individual prediction models for each language pair and translation source

4.3 System Selection Results

In what follows we turn to using the predictions from SVR models we have just described for system selection. The task consists of selecting, for each source segment, the best *machine* translation among all available: two or three depending on the language pair. For these experiments, we disregarded the human translations, as they do not tend to be present in settings for system selection, and would normally be better than the MT outputs in all cases. Another reason to rule out human translations from the selection is that they are used as references to compute BLEU scores of the selected sets of sentences, as explained below.

To provide an indication of the average quality of each MT system, Table 6 presents the BLEU scores on the QE test and training sets for individual MT systems. The bold-face figures for each language test set indicate the (BLEU) quality that would be achieved for that test set if the "best" system were selected on the basis of the average (BLEU) quality of the training set (i.e., no system selection). There is a significant variance in terms of quality scores, as measured by BLEU, among the outputs of 2–3 MT systems for each language pair, with training set quality being a good predictor of test set quality for all but **en-es**, once again, the largest dataset.

We measure the performance of the selected sets in two ways: (i) by computing the BLEU scores of the entire sets containing the supposedly best translations, using the human translation available in the datasets as reference, and (ii) by computing the accuracy of the selection process against the human labels, i.e., by computing the proportion of times both system selection and human agree (based on the predefined 1–3 human labels) that the sentence selected is the best among the 2–3 options (2–3 MT systems). We compare the results obtained from building pooled (all MT systems) against individual prediction models (one per MT system).

Table 7 shows the selection results with various models trained on MT translations only:

- **Best-SVR(I):** Best translation selected with regression model trained on data from individual MT systems, where prediction models are trained per MT system, and the translation selected for each source segment is the one with the

	system2		system3		system4	
WMT14	Test	Training	Test	Training	Test	Training
en-de	15.39	12.79	13.75	13.83	**17.04**	16.19
de-en	**27.96**	24.03	22.66	20.19	–	–
en-es	**25.89**	34.13	32.68	28.42	29.25	31.97
es-en	**37.83**	40.01	23.55	25.07	–	–

Table 6 BLEU scores of individual MT systems, without system selection. Bold-faced figures indicate scores obtained when selecting the best system on average (using BLEU scores for the training set)

	System	# Features	Best-SVR(I)	Best-SVR(P)
en-de	MC	–	16.14	15.55
	BL	17	17.20	17.05
	AF	80	**18.10**	17.55
de-en	MC	–	25.81	25.17
	BL	17	28.39	28.13
	AF	80	**28.75**	28.43
en-es	MC	–	30.88	30.29
	BL	17	32.92	32.81
	AF	80	**33.45**	33.25
es-en	MC	–	30.13	29.70
	BL	17	38.10	38.11
	AF	80	**38.73**	38.41

Table 7 BLEU scores on best selected translations (I = Individual, P = Pooled)

highest predicted score among these independent models. This requires knowing the source of the translations for training, but not for testing (blind test).

- **Best-SVR(P):** Best translation selected with regression model trained on pooled data from all MT systems. This assumes a *blind* setting where the source of the translations for both training and test sets is unknown, and thus pooling data is the only option for system selection.

Table 7 shows that the regression models trained on individual systems – **Best-SVR(I)** – with **AF** as feature set yield the best results, despite the fact that error scores (MAE and RMSE) for these individual systems are worse than for systems trained on pooled data. This is somewhat expected as knowing the system that produced the translation (i.e., training models for each MT system) adds a strong bias to the prediction problem towards the average quality of such a system, which is generally a decent quality predictor. We note, however, that the **Best-SVR(P)** models are not far behind in terms of performance. More important, we note the gains in BLEU scores as compared to the bold-face test set figures in Table 6, showing that our system selection approach leads to best-translated test sets rather than simply picking the MT system with best average quality (BLEU).

5 QE for Self-Training

One of the most efficient ways to improve the quality of an MT system is to supplement it with additional parallel training data. In some scenarios, monolingual data on either source or target languages (or both) can be abundant. However, parallel data has to be created by having humans translate monolingual content, which is an expensive process. Clever selection techniques to choose a subset with only the most useful sentences to translate from monolingual data can result in

systems with higher quality using less training data. These techniques are usually referred to as Active Learning (AL) (Settles 2010).

The majority of AL methods for MT are based on sentence (dis)similarity with the training data, with particular focus on domain adaptation. Eck et al. (2005) suggest a TF-IDF metric to choose sentences with words absent in the training corpus. Ambati et al. (2010) propose a metric of informativeness relying on unseen n-grams.

Similar to the work described here, Banerjee et al. (2013) proposed a data selection guided by automatic QE to identify poorly-translated sentences in the target domain. They restrict the reference set to the sentences that were poorly translated by the baseline model instead of using the entire target-domain data as reference for data selection.

An alternative approach is to select source sentences based on their estimated translation quality by a baseline MT system before the addition of new data. It is assumed that if a sentence has been translated well with the existing data, it will not contribute to improving the translation quality. If, however, a sentence has been translated poorly, it might have words or phrases that are absent or incorrectly represented. Haffari et al. (2009) use features including n-grams and phrase frequency, MT model score, etc. to decide which sentences to select. Ananthakrishnan et al. (2010) build a pairwise classifier that ranks sentences according to the proportion of n-grams they contain that can cause errors. For quality estimation, Banerjee et al. (2013) train language models of well- and badly-translated sentences. The usefulness of a sentence is measured as the difference of its perplexities in these two language models.

Logacheva and Specia (2014) proposed a new quality-based AL technique which is based on a more complex and therefore potentially more reliable QE framework. It employs a wider range of features, which go beyond those used in previous work, covering information from both source and target sentences. The approach adds post-edited or reference translations for MT outputs predicted to have low quality.

In this section we describe a similar quality-informed strategy, but focus on the addition of new data that has been translated by MT, rather than human references. Machine-translated segments predicted to have high enough quality are added to the training corpus of an SMT system. Therefore, we rely only on monolingual data. The assumption is that the MT segments added to the training corpus can help by reinforcing statistics on existing data.

Another direction we investigate here is the potential of using translations from an RBMT system to supplement the training data of an existing (iteratively improved) SMT system. In this case, in addition to reinforcing statistics on existing data, translations can also provide new information to the SMT system, helping, for example, to deal with out-of-vocabulary words. We compare the improvements obtained by an SMT system enhanced with either SMT or RBMT data, as well as against the improvements obtained by an SMT system enhanced with additional reference translations instead of MT outputs as in Logacheva and Specia (2014).

5.1 Active Learning Strategy

Four sets of data are necessary in our experiments: (i) parallel sentences to train an initial, baseline SMT system (including a subset for tuning), (ii) an additional pool of parallel sentences to select from (or monolingual sentences only, in the case of adding machine-translated segments to the SMT training corpus), (iii) source-MT segment pairs labelled for quality to train a QE model, and (iv) a held-out parallel test set to evaluate the performance of the baseline and improved SMT systems. We describe these datasets in Sect. 5.4.

Once a QE model is trained, the active learning pipeline includes the following steps:

1. Train a baseline SMT system.
2. Translate the pool of active learning data.
3. Predict the quality for the pool of AL data.
4. Select top *n* sentences based on QE scores and a given selection criterion to add to the SMT training data.
5. Remove top *n* sentences from the pool of AL data.
6. Retrain the SMT models including the additional selected data.
7. Go to step 2 until the AL pool is empty.

The SMT models are retrained incrementally with the additional QE selected data. The selection criteria are explained in the next section.

5.2 Selection Criteria

One of the aims of this work is to compare the use of MT against the use of reference translations, i.e. translations produced by humans. We therefore consider two scenarios as bases for the type of data to be added to the SMT training corpus: (i) reference translation sentences, which simulate a real AL setting where we would resort to humans to provide a translation for poorly-translated segments, (ii) machine-translated sentences (by an SMT or an RBMT system), where we assume human intervention is not possible or too costly, and so resort to the **self-training** of the SMT systems with their own or third party MT outputs.

In the second scenario, machine translations can be noisy and lead to degradation in MT performance. However, our hypothesis is that by filtering the candidates with a QE-based AL selection, we will select higher quality data to be added to the SMT training data, leading to improvements in overall performance.

More specifically, we experiment with the following settings to select data from the AL pool:

- SMT translations (source sentences and their machine translations) for the translations predicted as having highest QE scores (scenario 2, above).

- References (source sentences and their references) for the translations predicted as having the lowest QE scores (scenario 1, above).
- RBMT translations (source sentences and their machine translations) for the translations predicted as having highest QE scores (scenario 2, above).

5.3 SMT Models

We use the Moses toolkit to train our SMT system with phrase-based models using 14 standard features.[4] These feature functions include phrase and lexical translation probabilities in both directions, seven features for a lexicalised distortion model, word and phrase penalties, and a target language model. MERT (Och 2003) is used to tune the weights of these feature functions. For simplicity, our experiments use only the QuEst 17 baseline features, i.e., the **BL** set.

5.4 Datasets

We assume a common real-world scenario that explores two types of data: a relatively small parallel dataset and an additional (often larger) pool of source language only sentences. We use the former to train a baseline SMT system, translate the latter using this baseline SMT system and then inject a subset of sentences selected as outlined in Sect. 5.2 (either a human translation or the automatic translation produced by the MT system) to the initial parallel corpus and retrain the SMT system. The following datasets were used in the experiments. Their statistics are given in Table 8:

- **SMT training:** To train the initial SMT models we randomly selected 70% of the News Commentary training data for two language pairs: en-de and de-en. We set aside 30% of the corpus as AL pool.
- **SMT tuning and test:** We used the official WMT14 (translation task) tuning and test sets.
- **QE training:** To train our QE models we used data provided for WMT14 QE Task 1.1 (both training and test sets pooled together). The QE dataset and its labels are explained in Sect. 4.1.

Before we turn to the AL experiments, we look at the quality of the different versions of the AL pool data (reference, SMT and RBMT translations) in two ways. We first measured the BLEU score obtained for the SMT translations in the entire $\sim$ 60k AL pool (produced by the baseline version of the SMT system with $\sim$140k parallel sentences) versus the BLEU score obtained for the RBMT translations in

[4] http://www.statmt.org/moses/?n=moses.baseline

Corpora	de-en	en-de
Initial data (baseline SMT system)		
Training – 70% of News Commentary corpus	140,900	140,900
Tuning – WMT newstest-2013	3,000	3,000
Test – WMT newstest-2014	3,000	3,000
Additional data (AL data)		
AL pool data – remaining 30% of News Commentary corpus	60,388	60,388
QE data		
Training QE models – WMT14 QE task	1,500	2,000

Table 8 Statistics of the datasets used for the self-learning experiments

	de-en	en-de
SMT translations	13.71	11.29
RBMT translations	13.30	11.09

Table 9 BLEU score for sentences in the AL pool against the reference translations. SMT translations are generated by the baseline SMT system (before any incremental learning)

	de-en	en-de
Source-reference	16.83	12.12
Source-SMT translations	14.99	10.84
Source-RBMT translations	14.66	10.70

Table 10 BLEU score for test sentences from models built with variants of the AL pool data only (∼60 K parallel sentences)

the entire ∼60k AL pool, both against the reference translations. These are shown in Table 9. The quality, in terms of BLEU, of both datasets is very similar, with the SMT translations achieving slightly higher figures for both languages.

Second, we built an SMT system using only the ∼60k AL pool data for training, without incremental training with either the baseline SMT translations, the RBMT translations, or the reference translations. These models were tested on the official test set (WMT newstest-2014) and the results in terms of BLEU scores are shown in Table 10. Surprisingly, the SMT and RBMT translations seem equally useful as SMT parallel training corpora. We had hypothesised that SMT translations tend to be closer to the source segments than the latter in word order and style, leading to better word-alignment performance, which in turn leads to better translation models. This, however, does not seem to be the case with this dataset. Reference translations are clearly more helpful in building better SMT systems.

5.5 Results

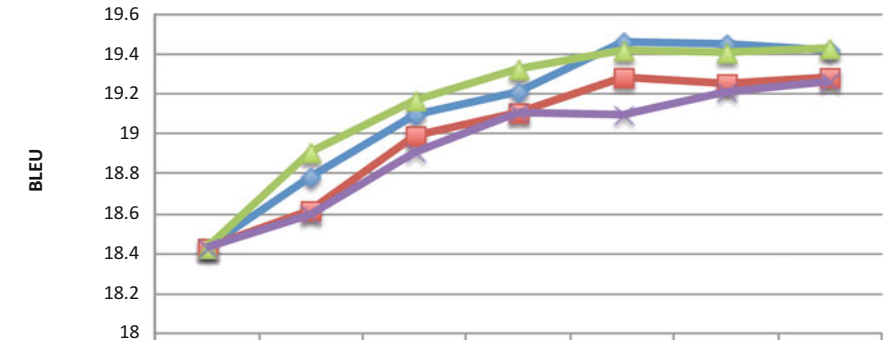
We conducted a set of experiments to show the improvement rate of our main selection strategy (adding MT data) compared to reference/random data selection. With the **SMT translations**, at every iteration, based on quality predictions for translations in the AL pool produced by the current SMT system, batches of 10 K sentences from the pool with the predicted highest/lowest scores (depending upon MT or reference translation as selection criterion) were selected. These were added to the training data of the SMT system, which was then retrained using the **incremental training** option in Moses[5] to skip some of its initial, time-consuming steps. The selected sentences were removed from the AL pool. The new SMT system was applied to translate the held-out test set, and the performance was measured using BLEU. The process was repeated until the pool was empty. We note that the updated SMT system is also used to translate the remaining sentences in the AL pool.

RBMT translations were generated by the Lucy system, which is known to achieve the state-of-the-art performance for English↔German. The process here was slightly different: since the RBMT system cannot be easily updated based on quality predictions, the SMT system was updated with RBMT translations. Also different from the experiments with SMT translations, the AL pool was translated once, and each translation had its quality predicted also only once. The AL was then sorted by highest predicted quality and batches of 10 K (best-worst) were taken for every step and added to the SMT training corpus. As in the remaining experiments, the SMT system was retrained to translate the test set at every step.

In order to compare the impact of **SMT**, **RBMT** and (**Reference**) translations on SMT quality, we added these variants of translations to baseline SMT systems (10 K sentences at each iteration), starting with the baseline, using 70% of the News Commentary corpus. To evaluate the effectiveness of the QE predictions for the SMT translations, we also add a baseline that selects the 10 K batches of SMT translations randomly (**Random**). This allows us to contrast simply adding more data against adding more supposedly good quality data. Results are shown in Figs. 3 and 4, for each language pair. All BLEU figures are reported based on the test set. The BLEU scores show that adding more data significantly improves the results using all variants of the selection strategy.

Overall, as expected, the use of reference translations leads to better performance than using machine-translated segments. However, the performance obtained with the use of SMT translations follows closely behind. The use of SMT translations is even better than References for one particular step (iteration 3) with English-

[5] As detailed in http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc37, instead of producing a phrase table with pre-calculated scores for all translations, the entire source and target corpora are stored in memory as a suffix array along with their alignments, and translation scores are calculated on the fly. When new training data is available, the word alignments are simply updated.



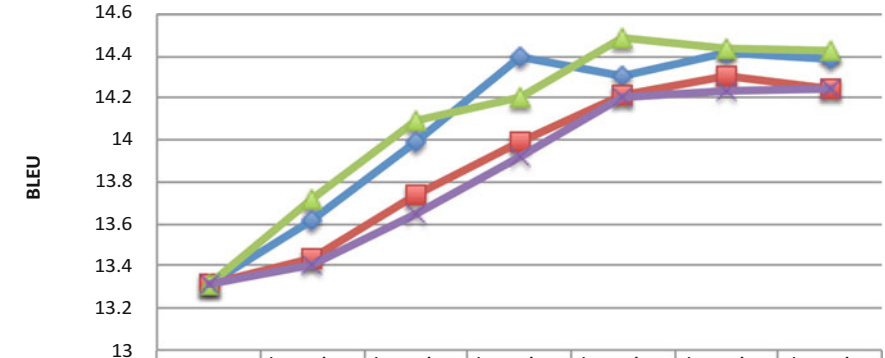
	Baseline	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6
SMT	18.43	18.78	19.1	19.21	19.46	19.45	19.42
RBMT	18.43	18.62	18.99	19.11	19.29	19.25	19.29
References	18.43	18.91	19.17	19.33	19.42	19.41	19.43
Random	18.43	18.59	18.91	19.11	19.1	19.21	19.27

Fig. 3 Performance of de-en enhanced with data selected by different AL strategies

German translation. More importantly, the differences in the final scores (iteration 6) for SMT and References are virtually non-existent for de-en, and very marginal for en-de. This is a very positive result, as it shows that the same level of improvements can be obtained with machine-translated segments instead of reference translations. Another very interesting observation was that we observed that the performance for both language pairs is higher by using smaller amounts of data selected with QE rather than using the entire dataset. In particular, for de-en, at iteration 4 the BLEU score achieved is slightly superior to the score achieved when the entire AL pool is used (both with references and machine translations). This could indicate that some references may be noisy or difficult to align to their corresponding source segments, proving less helpful to the SMT system. The use of RBMT translations, on the other hand, does not seem very helpful, as its performance is close to or worse than that of randomly selecting SMT translations.

To highlight some important differences in terms of impact on SMT systems' performance with various settings, we look more closely at the following comparisons: The impact of each system can be observed in Figs. 3 and 4.

- **SMT vs. Reference:** In a first comparison between SMT translations and reference translations on SMT quality, it seems very encouraging that we get similar final scores (or very close) with both additional references and additional SMT data. SMT translations are much cheaper to obtain than reference translations. While our AL pool was relatively small, one could rely on much larger collections of monolingual data for this approach.
- **RBMT vs. Reference:** This comparison inspects the impact of RBMT systems versus reference translations on SMT quality. RBMT translations seem to



	Baseline	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6
SMT	13.31	13.62	13.99	14.4	14.31	14.42	14.39
RMBT	13.31	13.43	13.74	13.99	14.21	14.31	14.25
References	13.31	13.72	14.09	14.2	14.49	14.44	14.43
Random	13.31	13.4	13.65	13.92	14.2	14.23	14.25

Fig. 4 Performance of en-de enhanced with data selected by different AL strategies

perform substantially worse than reference translations. We note that the RBMT system has not been customised in any manner to translate the type of data used in the experiments (news).

- **SMT vs. RBMT:** The improvements with SMT data are consistently higher than RBMT for both directions. One reason for that is the incremental versus static version of the experiments with both types of translations. As previously mentioned, in the RBMT setting, the translations in the remaining AL pool cannot be updated as the SMT system is updated, since they are generated from an RBMT system. Additionally, intuitively translations produced by RBMT systems are less close to the source segments than translations produced by SMT systems. The latter can thus be potentially more easily word-aligned by automatic tools, rendering them more useful to SMT retraining.
- **SMT vs. Random:** Here we compare our selection technique from SMT data against randomly selecting SMT data. From Figs. 3 and 4 we can see that our selection strategy with SMT data consistently outperforms random selection.
- **RBMT vs. Random:** Finally we compare our selection technique from RBMT data against randomly selecting data. RBMT data and random selection perform very similarly.

One final aspect investigated was the effects of incremental training, as opposed to batch training, on the final translation quality. We tested the performance of an SMT system built from the entire parallel corpus of source and reference translations, by simply concatenating the original ∼140k and the additional ∼60k segment pairs and training a batch model for it. For the batch mode, the scores

obtained are 19.63 (de-en) and 14.65 (en-de). We recall that the BLEU scores obtained by these systems with the iterative AL setting (at the final iteration 6) are lower: 19.43 (de-en) and 14.43 (en-de). This shows that incremental learning leads to some performance degradation. If time is not an issue, one solution to this problem is to retrain the SMT models from scratch at every AL iteration, instead of using incremental training.

6 Sampling QE for Quality Assurance

Human assessment of translations for quality assurance purposes is a cognitively intensive and time-consuming task. While various assessment methods exist (e.g. the LISA QA model; see also the chapters by Popović and Lommel in this volume) that provide insight into translation quality, they cannot be implemented within the rapid development cycles that characterise the use of MT. Even with HT, quality assessment is often done on small samples of translations.

Traditionally, samples for quality assessment are selected at random. Random selection is a valid choice if enough data can be sampled for analysis, as this would reflect the natural distribution of errors across the entire set. However, more often than not, very small samples of translations are selected, potentially leaving out certain issues. In addition, for different purposes, it may be desirable to focus the assessments on the lower/higher quality cases translations. In this section we propose a quality-informed sampling method where translations estimated to have a certain level of quality (e.g. average, top or lower levels) are selected for human inspection. We contrast this method against random selection in terms of the number of selected translations that can be effectively assessed and the distribution of issues found.

We compare the task of quality assessment on data selected at random against data selected according to quality predictions for four language pairs. The two samples are given to human translators for error annotation using the Multidimensional Quality Metrics (MQM) error typology (see Lommel et al. (2014) and Lommel in this volume). Translations with quality predicted to be around average for the set are selected. This decision was based on the fact that translations with high quality do not require human inspection, and translations with very low quality are too hard – if not impossible – to have errors identified. One of our hypotheses was that translators could find more errors in samples of translations selected using QE, as many samples selected at random are too bad to be annotated. However, our analysis showed that this is not the case: the absolute number of errors found with randomly selected cases is still higher. Nevertheless, the error distributions in both types of samples were very similar. This indicates that samples with average quality, which are potentially easier and less time-consuming to annotate, still offer an advantage for human quality assessment over random samples.

6.1 Datasets

The datasets used for **training** the models were taken from official WMT14 task 1.1 on QE, which was described in Sect. 4.1. However, here we only use translations produced by the statistical phrase-based system. We train four QE models, one per language pair, with 500 instances for all but the en-es data, which has 1104 instances.

We apply the models to generate predictions for the WMT10–11 translation task **test sets**, taking only those segments whose source is originally in the language of interest (∼600 segments).

6.2 Sampling and Error Annotation

After training QE models for each of the datasets, we took a sample of 100 sentences whose quality predictions are the closest possible to 2 (good enough). The hypothesis here is that QE is helpful to select near-miss segments for manual inspection in order to perform systematic QE: perfect cases do not need to be inspected, worst cases are too bad to be inspected manually. It is worth mentioning that other selection criteria could be defined, such as selecting sentences with the lowest predicted quality. For comparison purposes, we selected a non-overlapping random sample of another 100 sentences.

For each language pair, we generated a combination of 100 QE-based and 100 random samples consisting of source segments and their translation. We gave these segments for annotation without disclosing the source of the sample. translate5 was used as annotation tool.[6] Each segment was annotated by four professional translators who received training on the annotation task and on translate5. Annotators were requested to annotate only cases with errors and mark segments that were too bad to be annotated as "fully unintelligible". In total, annotations were performed on 3,200 segments, i.e., 200 segments for four language pairs, with four annotators for each segment.

For human annotation we used a subset of MQM. This set of issues provides a reasonably comprehensive set of analytic issues that can be applied to spans within segments to identify specific issues at a fairly granular level. MQM issues are arranged in a hierarchy with more and less general types. A selection of core MQM issues which was designed specifically to analyse MT output is used here:

- **Accuracy**. Issues related to the relationship of the target and source content.
  - **Omission**. Content present in the source is improperly omitted in the target.

[6] http://test.translate5.net/

  - **Mistranslation**. Content is translated with a different meaning from the source.
  - **Untranslated**. Content present in the source remains in the source language.
  - **Addition**. Content not present in the source has been added to the target text.
- **Fluency**. Issues related to the linguistic properties of the target language itself without regard to the fact that it is a translation.
  - **Spelling**. The text is misspelled (including capitalisation problems).
  - **Typography**. The text does not follow typographic conventions (other than spelling).
  - **Grammar**. There is a grammatical problem with the text.
    - **Word Form**. The text uses an incorrect word form.
      - **Part of speech**. The text uses the wrong part of speech.
      - **Agreement**. The text shows problems with number, gender, or case agreement.
      - **Tense/aspect/mood**. Verbs show incorrect tense, aspect, or mood.
    - **Word Order**. Portions of the text appear in the wrong order.
    - **Function word**. Function words (e.g. articles, prepositions) are used incorrectly
      - **Extraneous**. The text contains unneeded function words.
      - **Missing**. The text is missing needed function words.
      - **Incorrect**. The text uses function words incorrectly.
  - **Unintelligible**. The meaning of the text cannot be recovered. Used for cases in which a serious break-down of fluency has occurred.

As these issues are hierarchical in nature, if none of the subtypes for a given category apply, then the parent may be chosen. In addition to these categories, annotators were given an additional option to select: *fully unintelligible*. This annotation was used for cases where the annotators found the fluency or accuracy of the target segment so bad that they would not be able to identify individual errors in the translation.

6.3 Results and Analysis

In what follows we summarise the most important findings when comparing the annotation of QE-based samples versus random samples.

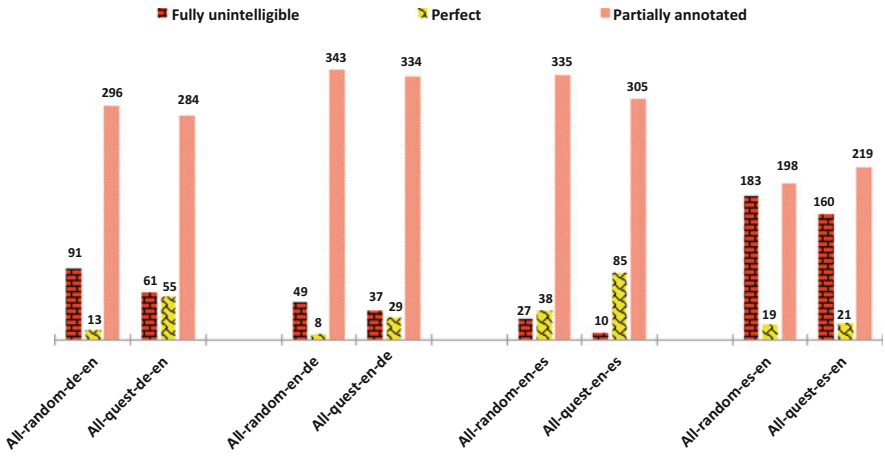


Fig. 5 Number of sentences (out of 400 per language pair) which are fully unintelligible, perfect or annotated for errors

6.3.1 Fully Unintelligible, Perfect and Annotated Segments

Figure 5 shows the number of three types of segments in each dataset, where "QuEst" represents the QE-based selection:

- **Perfect:** Segments that are not annotated at all as they are perfectly good translations.
- **Fully unintelligible:** Segments that are so bad that they cannot be annotated.
- **Annotated:** The remaining segments which are neither perfect nor fully unintelligible and are good enough for annotation.

Fewer fully unintelligible cases were found across all datasets with the QE-based sampler than the random sampler for en-es, de-en, and en-de. This finding is in line with our hypothesis that systematic quality evaluation can help in discarding segments which are too bad for annotation. However, in the case of es-en, we did not find the expected difference.

Although we are not certain why es-en results were different, annotators for this pair seem to have been much more critical of the output, annotating almost half of all segments as *fully unintelligible*. In previous annotation work,[7] we found that es-en translations were particularly prone to grammar problems, especially with incorrect subject and object pronouns, when compared to the other language pairs. Because Spanish is a "pro-drop" language with considerable verbal syncretism that relies on context for disambiguation, segments often lack sufficient syntactic and morphological information for proper translation without a consideration of their context. Since pronouns and verbal forms are particularly important for

[7]See QTLaunchPad Deliverable D1.3.1, "Barriers for High-Quality Machine Translation", p 15–20, at http://www.qt21.eu/launchpad/system/files/deliverables/QTLP-Deliverable-1_3_1-v2.0.pdf

understanding the meaning of sentences, it may be that annotators found many sentences unintelligible at first glance, which would have been intelligible in other language pairs.

As expected, the number of perfect segments is low in all datasets. This finding is true even for the QE sampler, given that segments were selected to have average rather than good quality. Nevertheless, more perfect segments were selected by QE than by the random sampler. As long as this number is still much lower in comparison to the remaining selected segments, it should not be a problem, as perfect segments can be easily skipped by annotators.

6.3.2 Total of Errors Annotated

The number of errors for each of the datasets, on a per-annotator basis, is shown in Fig. 6. The number of errors found in random samples is clearly larger than in QE samples, except for es-en, where the figures are very close, probably for the same reasons as noted above: the annotators were more critical in rejecting sentences outright. While different annotators annotated different numbers of errors, the relative differences in error counts between QE-based and random samples are maintained across annotators. For this analysis a fully unintelligible segment is counted as one error. However, those segments would most likely contain multiple errors had they been annotated. This may also explain the difference between annotators, as some annotators chose to mark more entire segments as unintelligible than others. Finally, it could also explain the case of es-en, where many of the segments were marked as fully unintelligible by all of the annotators. The fact that QE led to a higher proportion of "perfect" segments being sampled will naturally decrease the number of errors found in its samples.

For a more detailed analysis, we excluded from the counts the segments marked as fully unintelligible. The total number of errors per language pair (all annotators together) can be seen in Fig. 7. There is a clear drop in the number of errors for all language pairs, but the trend between random sampling and QE-based sampling is maintained: a higher number of errors is found with the randomly selected samples, except for es-en, where the number of errors in both samples is virtually the same: 648 (random) and 652 (QE). This is most likely a consequence of the fact that annotators discarded nearly 50% of the cases that are too complex to annotate in both samples, and thus the remaining sets in both cases will contain translations of similar levels of quality.

One important finding that stands out from Fig. 7 is the fact that, with the random sample, there are fewer segments annotated (more were rejected), but in absolute terms they contain more errors than the larger sets of segments selected by QE. Therefore, the proportion of errors per segment is much higher with random samples. Since we could not log annotation time, it is unclear whether annotating fewer segments with more errors is more time-consuming than annotating more segments with fewer errors. We can however hypothesise that samples with fewer errors per segment may lead to more consistent annotations, as multiple errors are

often interrelated, making annotation harder and therefore more prone to mistakes and inconsistencies, particularly across annotators.

6.3.3 Distribution of Error Types

For a closer look at the overall distribution of errors, in Fig. 8 we combine annotations from all translators for each language pair and plot the proportion of each type of error, i.e., we normalise the counts of each error type by the total number of errors for that language pair (all annotators). The figures were obtained after excluding all segments marked as fully unintelligible. Across all datasets, mistranslation is the most common error type, followed by word order issues. A significant proportion of errors fall under the "unintelligible" category, particularly for es-en. This category covers unintelligible parts of a segment, as opposed to representing cases where the entire segment is too bad to be annotated. Given the small number of samples, particularly after excluding fully unintelligible cases, it is to be expected that certain types of errors will not be observed at all. Surprisingly however, this is only the case for very few error types, and these are mostly general error types, which work as fall back options when the exact error cannot be identified, such as the accuracy and fluency categories.

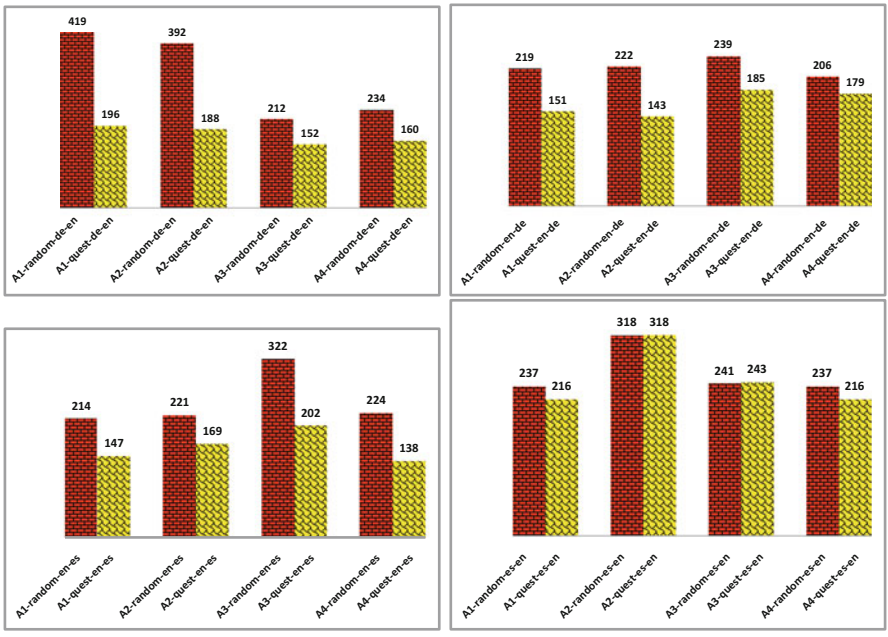


Fig. 6 Errors per dataset and annotator. Fully unintelligible segments count as one error

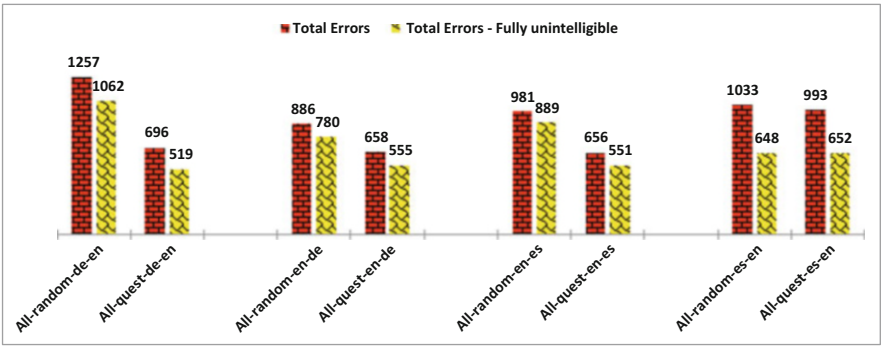


Fig. 7 Number of errors within all segments against number of errors within segments excluding those fully unintelligible, per language pair

Interestingly, the distribution of errors are very similar between random and QE-based samples. This shows that both sampling techniques will lead to spotting the same types of errors, in the same proportions. However, as was mentioned before, different annotators chose to annotate different segments, as they considered a different number of (potentially non-overlapping) fully unintelligible or perfect segments. In Fig. 9 we further analyse the error distributions by excluding all segments which at least one of the four annotators judged to be either perfect or fully unintelligible. In other words, given a segment and its four annotations, if one or more of these annotations was set as "perfect" or "fully unintelligible", the remaining 1–3 annotations were also set as "perfect" or "fully unintelligible" and removed from the analysis. The counts of each error type were thus normalised by the total number of errors for that language pair (all annotators) that remained after the exclusion. This was an attempt to isolate any disagreements between annotators.

Mistranslation and word order are still the most common error types across all datasets. The distribution of errors is still similar between random and QE-based samples. The effect of removing potentially conflicting segments is very visible for all language pairs, and particularly for es-en: the proportion of partially unintelligible cases became virtually zero. These were probably cases which some annotators had chosen to mark as fully unintelligible, while others had gone to the effort of marking parts of the segment as unintelligible.

7 Discussion and Future Directions

We have presented a number of applications of QE. While a number of evaluation campaigns and other benchmarking efforts have been made in recent years to measure progress in QE (we refer the reader to Callison-Burch et al. (2012b) and Bojar et al. (2013, 2014, 2015, 2016) for comprehensive experiments), our intention

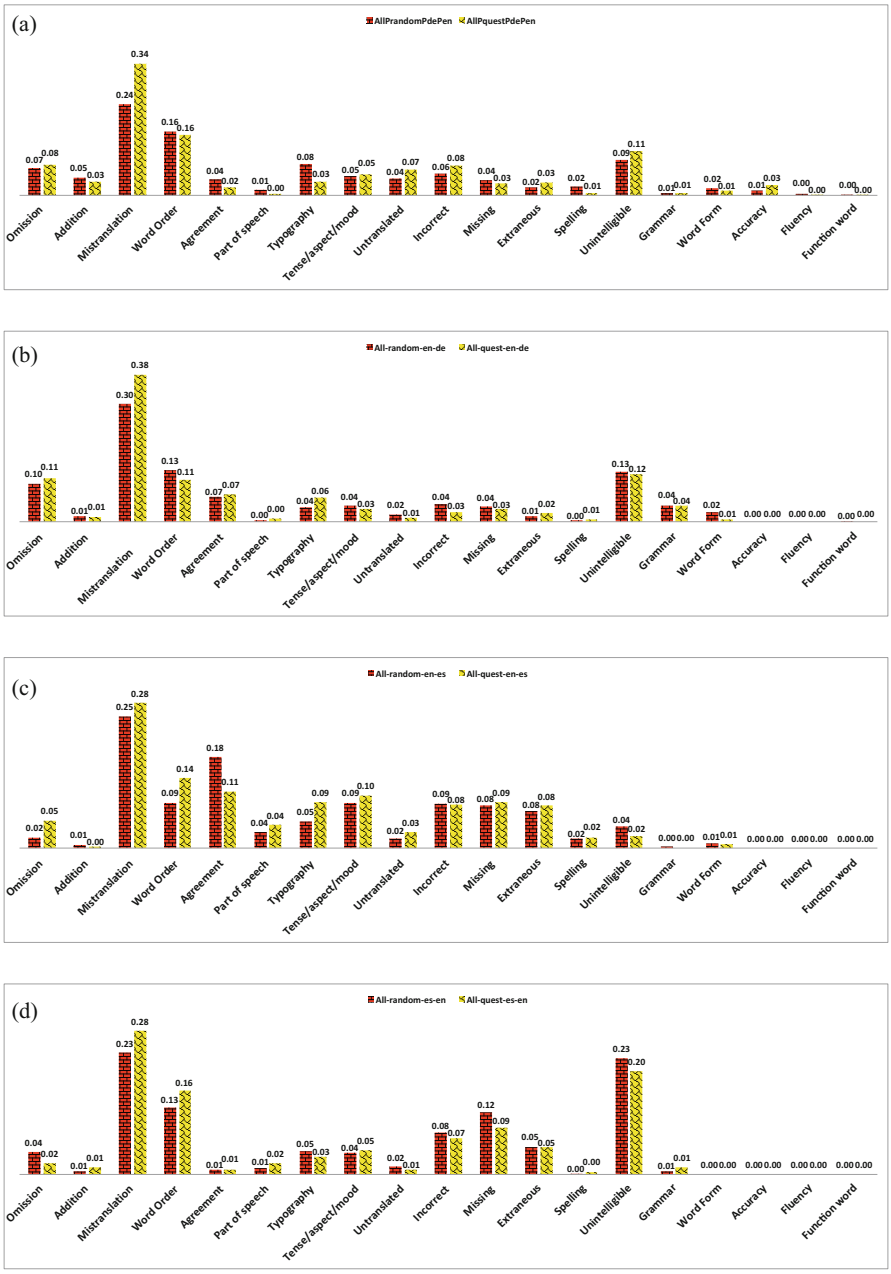


Fig. 8 Proportion of error types for all annotators per language pair, after excluding segments judged by at least one of the four annotators as perfect or fully unintelligible. (**a**) de-en. (**b**) en-de. (**c**) en-es. (**d**) es-en

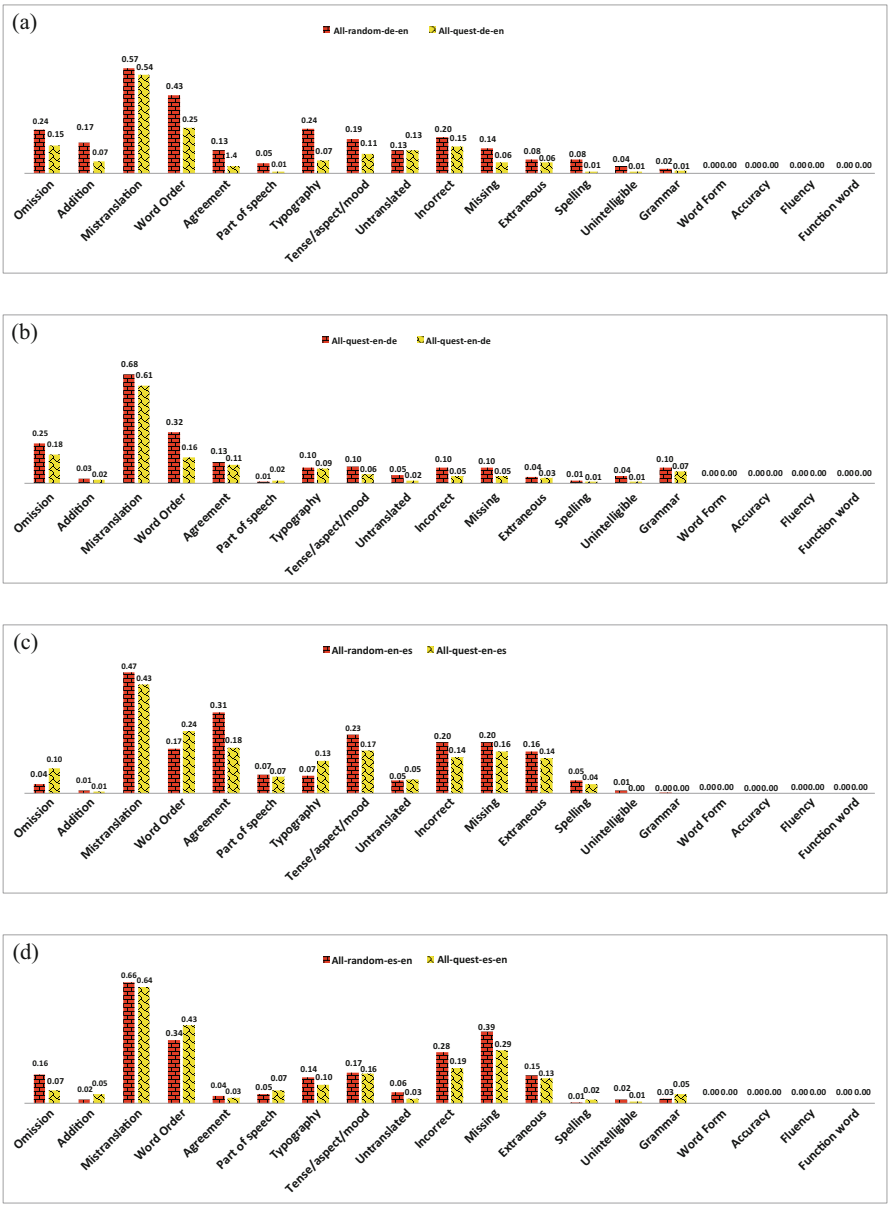


Fig. 9 Proportion of error types for all annotators per language pair, after excluding all fully unintelligible annotations as set by at least one annotator. (**a**) de-en. (**b**) en-de. (**c**) en-es. (**d**) es-en

was to shed some light on promising practical uses of QE and on more intuitive evaluation approaches for these applications. Our focus was on sentence-level QE.

QE for predicting post-editing effort as described in Sect. 3 is perhaps the most widely studied variant of the task, with very clear application in the translation industry: translators are often required to post-edit the output of MT systems, but for many segments the effort required to fix the MT output is greater than that of translating the source segment from scratch. Filtering out these cases is very desirable to improve productivity and user experience. In addition, this information could be used to customise pricing of MT post-editing, as well as to estimate the time a post-editing job would require to be completed. Work done in this direction has showed promising results, but an important topic that is still to be researched is the investigation of the reliability and utility of quality labels in translation workflows. Preliminary experiments have been done in Turchi et al. (2015) and Specia (2011). The former focused on the usefulness of showing the translator a binary (good/bad) quality prediction for the sentence during post-editing, without performing any filtering on the MT output. The latter compared the time taken to post-edit sentences predicted to have high quality according to QE against sentences selected at random. While it showed that the QE-selected sentences can be post-edited in much shorter time, it did not factor in the translation from scratch of sentences predicted to have low quality.

The utility of QE for MT system selection can be more easily validated by checking the final quality of the selected dataset in terms of automatic metric scores such as BLEU, as was done in Sect. 4. As long as a reference set is provided to compute such metrics, this can be done automatically, without further human intervention. The results of the experiments presented in this chapter are very promising, showing that QE-selected sets are able to demonstrate improvements of up to 7.56 BLEU points over individual MT systems.

The same evaluation criterion can be used in the employment of QE for MT "self-learning" (Sect. 5). Our results using QE to select sentences predicted to have high enough quality to add to the SMT training corpus showed consistent improvements of around 1 BLEU point, which is virtually the same level of improvement obtained from adding the corresponding reference (human) translations to the SMT training corpus.

Work presented in Sect. 6 is a clear attempt to validate QE in a real-world application where the purpose is effective error annotation by human translators for quality assurance. The results of our experiments were, however, somewhat inconclusive, potentially due to the criterion used for the QE-based sampling: average quality translations. Future experiments should include selecting sets with different levels of quality, leading to a more general sample for quality assurance. This criterion is highly dependent on the objective of the error annotation process: finding the largest number of errors, annotating the largest number of segments, etc.

It is also important to mention that although we have focused on quality prediction for sentences, QE can be performed at other textual levels. QE has been gaining increasing attention at word and document levels. Word-level QE is useful

for pinpointing specific errors in the words of a translated segment. It has various interesting applications, among others:

- Highlight words that need editing in post-editing tasks.
- Inform readers of portions of the sentence that are not reliable.
- Select the best words/phrases among options from multiple MT systems for system combination.
- Guide automatic post-editing.

Most of the work on word-level QE has focused on prediction of automatically derived labels. These are obtained mainly by aligning the MT output to its post-edited version, as has been done in most of the WMT shared tasks on QE (Bojar et al. 2013, 2015, 2016). To minimise the amount of annotated data that is needed and reduce data sparsity, errors are often conflated into one category, resulting in a binary classification task: correct versus incorrect target words. In 2014, the word-level QE shared task at WMT instead provided specific errors manually annotated according to 21 error categories from MQM. However, this introduced significant sparsity in the data, which made learning from it virtually impossible. Existing work exploits classification and sequence-labelling algorithms with a range of word-level and contextual features. Overall, this looks likely to remain a more challenging task. The context plays an important role in deciding whether a target word is an incorrect translation, but often words in context are also incorrect. A much larger number of examples is necessary to represent occurrences of target words in various contexts, and often the modelling is hindered by skewed class distributions: most words in a sentence tend to be correct.

Document-level QE focuses on more coarse-grained assessments to judge the overall quality of entire documents. While certain sentences are perfect in isolation, their combination in context may lead to an incoherent document. Conversely, while a sentence can be poor in isolation, when put in context it may benefit from information in surrounding sentences, leading to a good quality document. Document-level QE is needed particularly for gisting purposes where post-editing is not an option. An example application is quality prediction for translations of product reviews in order for readers to decide whether or not they are understandable and to select a subset of reviews for a given product that are good enough to be published. This level of prediction has been included in recent years as part of the WMT shared task on QE (Bojar et al. 2015, 2016), but has attracted very few participants.

Document-level QE is also very challenging as it requires annotations for quality at document level and modelling of discourse features (Scarton et al. 2015). No standard quality labels exist that capture all potential issues at document level. As for discourse features, very few processing tools are available to extract discourse-wide information. Moreover, the performance of existing tools (mostly for English) is negatively impacted by the various types of errors in the MT output.

Sentence-level QE, despite its popularity, is far from a solved problem. While extensive work has been done on feature engineering, this continues to be an active topic, with recent research showing the value of combinations of shallow

and linguistically-motivated features (Bojar et al. 2016). Various approaches also explore neural models, including using them to generate features (Shah and Specia 2016) and to train prediction models (Kim and Lee 2016; Kim et al. 2017). Larger datasets have been produced in recent years (15K samples in WMT16 instead of 2.2K in WMT12), with additional post-editing data also used, where the edit distance between the original and revised MT output is taken as the quality label.

In recent years, a number of software toolkits have been made available to facilitate research and use of QE approaches. These include QuEst++ (Specia et al. 2015b), Marmot (Logacheva et al. 2016), WCE LIG (Servan et al. 2015), and Qualitative (Avramidis 2016). These tools generally differ in the feature set they extract and the type of machine-learning algorithms they provide, since they focus on different levels and types of prediction.

Overall, we believe that successful approaches to QE have immense potential to make MT more useful to end-users of various types. As a research area, many aspects of the problem require further investigation. Therefore, it is likely that QE at all levels will continue to be an active area of research, with continuing efforts by the community to push the field forward, ideally in collaboration with end-users to validate the proposed solutions.

Acknowledgements We thank Arle Lommel for his help with setting up and running the error annotation for the experiments in Sect. 6.

References

Ambati V, Vogel S, Carbonell J (2010) Active learning and crowd-sourcing for machine translation. In: Proceedings of the seventh international conference on language resources and evaluation (LREC 2010), 17–23 May 2010, Valletta, pp 2169–2174

Ananthakrishnan S, Prasad R, Stallard D, Natarajan P (2010) Discriminative sample selection for statistical machine translation. In: Proceedings of the 2010 conference on empirical methods in natural language processing (EMNLP-2010), MIT, Massachusetts, 9–11 Oct 2010, pp 626–635

Avramidis E (2013) Rankeval: open tool for evaluation of machine-learned ranking. Prague Bull Math Linguist (PBML) 100:63–72

Avramidis E (2016) Qualitative: python tool for MT quality estimation supporting server mode and hybrid MT. Prague Bull Math Linguist 106:147–158

Bach N, Huang F, Al-Onaizan Y (2011) Goodness: a method for measuring machine translation confidence. In: ACL11, Portland, pp 211–219

Banerjee P, Rubino R, Roturier J, van Genabith J (2013) Quality estimation-guided data selection for domain adaptation of SMT. In: MT summit XIV: Proceedings of the fourteenth machine translation summit, Nice, 2–6 Sept 2013. EAMT, pp 101–108

Blatz J, Fitzgerald E, Foster G, Gandrabur S, Goutte C, Kulesza A, Sanchis A, Ueffing N (2004) Confidence estimation for machine translation. In: Coling04, Geneva, pp 315–321

Bojar O, Buck C, Callison-Burch C, Federmann C, Haddow B, Koehn P, Monz C, Post M, Soricut R, Specia L (2013) Findings of the 2013 WMT. In: 8th WMT, Sofia, pp 1–44

Bojar O, Buck C, Federmann C, Haddow B, Koehn P, Monz C, Post M, Specia L (eds) (2014) Proceedings of the ninth workshop on statistical machine translation, Baltimore

Bojar O, Chatterjee R, Federmann C, Haddow B, Hokamp C, Huck M, Logacheva V, Pecina P (eds) (2015) Proceedings of the tenth workshop on statistical machine translation, Lisbon

Bojar O, Chatterjee R, Federmann C, Graham Y, Haddow B, Huck M, Jimeno Yepes A, Koehn P, Logacheva V, Monz C, Negri M, Neveol A, Neves M, Popel M, Post M, Rubino R, Scarton C, Specia L, Turchi, M, Verspoor K, Zampieri M (2016) Findings of the 2016 conference on machine translation. In: First conference on machine translation, volume 2: shared task papers, WMT, Berlin, pp 131–198

Callison-Burch C, Koehn P, Monz C, Post M, Soricut R, Specia L (2012a) Findings of the 2012 WMT. In: WMT12, Montréal, pp 10–51

Callison-Burch C, Koehn P, Monz C, Post M, Soricut R, Specia L (eds) (2012b) Proceedings of the seventh workshop on statistical machine translation, Montréal

Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27

Eck M, Vogel S, Waibel A (2005) Low cost portability for statistical machine translation based on N-gram frequency and TF-IDF. In: IWSLT 2005: proceedings of the international workshop on spoken language translation, Pittsburgh, 24–25 Oct 2005

Haffari G, Roy M, Sarkar A (2009) Active learning for statistical phrase-based machine translation. In: The 2009 annual conference of the North American chapter of the Association for Computational Linguistics. https://doi.org/10.3115/1620754.1620815

He Y, Ma Y, van Genabith J, Way A (2010) Bridging SMT and TM with translation recommendation. In: ACL2010, Uppsala, pp 622–630

Kim H, Lee JH (2016) Recurrent neural network based translation quality estimation. In: Proceedings of the 1st conference on MT, pp 787–792

Kim H, Lee JH, Na SH (2017) Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In: Proceedings of the 2nd conference on MT, pp 562–568

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: 45th annual meeting of the Association for Computational Linguistics: demo and poster sessions, Prague, pp 177–180

Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions and reversals. Sov Phys Dokl 10(8):707–710. https://www.bibsonomy.org/bibtex/220546d80ce76f58c6ef6ece9dd5f5056/jimregan

Logacheva V, Specia L (2014) A quality-based active sample selection strategy for statistical machine translation. In: Chair NCC, Choukri K, Declerck T, Loftsson H, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (eds) Proceedings of the ninth international conference on language resources and evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik

Logacheva V, Hokamp C, Specia L (2016) Marmot: a toolkit for translation quality estimation at the word level. In: Tenth international conference on language resources and evaluation, Portoroz, pp 3671–3674

Lommel A, Popovic M, Burchardt A (2014) Assessing inter-annotator agreement for translation error annotation. In: Automatic and manual metrics for operational translation evaluation workshop programme, p 5

Och FJ (2003) Minimum error rate training in statistical machine translation. In: Proceedings of the 41st annual meeting on Association for Computational Linguistics, ACL '03, Sapporo, vol 1. Association for Computational Linguistics, Stroudsburg, pp 160–167. https://doi.org/10.3115/1075096.1075117

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: ACL02, Philadelphia, pp 311–318

Quang LN, Laurent B, Benjamin L (2014) LIG system for word level QE task at WMT14. In: Workshop on machine translation (WMT)

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge, MA

Scarton C, Zampieri M, Vela M, van Genabith J, Specia L (2015) Searching for context: a study on document-level labels for translation quality estimation. In: 18th annual conference of the European Association for machine translation, Antalya, pp 121–128

Servan C, Le NT, Luong NQ, Lecouteux B, Besacier L (2015) An open source toolkit for word-level confidence estimation in machine translation. In: 12th international workshop on spoken language translation, Da Nang

Settles B (2010) Active learning literature survey. Computer sciences technical report 1648, University of Wisconsin, Madison

Shah K, Specia L (2016) Large-scale multitask learning for machine translation quality estimation. In: Conference of the North American chapter of the association for computational linguistics: human language technologies, San Diego, pp 558–567. http://www.aclweb.org/anthology/N16-1069

Shah K, Cohn T, Specia L (2013) An investigation on the effectiveness of features for translation quality estimation. In: Machine translation summit XIV, Nice, pp 167–174

Shah K, Cohn T, Specia L (2015) A Bayesian non-linear method for feature selection in machine translation quality estimation. Mach Translat 125. https://doi.org/10.1007/s10590-014-9164-x

Soricut R, Echihabi A (2010) TrustRank: inducing trust in automatic translations via ranking. In: ACL11, Uppsala, pp 612–621

Specia L (2011) Exploiting objective annotations for measuring translation post-editing effort. In: EAMT11, Leuven, pp 73–80

Specia L, Turchi M, Cancedda N, Dymetman M, Cristianini N (2009) Estimating the sentence-level quality of machine translation systems. In: EAMT09, Barcelona, pp 28–37

Specia L, Raj D, Turchi M (2010) Machine translation evaluation versus quality estimation. Mach Translat 24:39–50

Specia L, Shah K, Souza JGCD, Cohn T (2013, to appear) QuEst – a translation quality estimation framework. In: Proceedings of ACL demo session

Specia L, Paetzold G, Scarton C (2015a) Multi-level translation quality prediction with quest++. In: ACL-IJCNLP 2015 system demonstrations, Beijing, pp 115–120

Specia L, Paetzold GH, Scarton C (2015b) Multi-level translation quality prediction with quest++. In: Proceedings of the 53rd ACL

Turchi M, Negri M, Federico M (2015) MT quality estimation for computer-assisted translation: does it really help? In: 53rd annual meeting of the association for computational linguistics, Beijing, pp 530–535