One-parameter models for sentence-level post-editing effort estimation

Mikel L. Forcada mlf@ua.es
Miquel Esplà-Gomis mespla@dlsi.ua.es
Felipe Sánchez-Martínez fsanchez@dlsi.ua.es
Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant,
E-03690 Sant Vicent del Raspeig, Spain.

Lucia Specia l.specia@sheffield.ac.uk
Department of Computer Science, the University of Sheffield,
Sheffield S1 4DP, United Kingdom

Abstract

Methods to predict the effort needed to post-edit a given machine translation (MT) output are seen as a promising direction to making MT more useful in the translation industry. Despite the wide variety of approaches that have been proposed, with increasing complexity as regards their number of features and parameters, the problem is far from solved. Focusing on post-editing time as effort indicator, this paper takes a step back and analyses the performance of very simple, easy to interpret one-parameter estimators that are based on general properties of the data: (a) a weighted average of measured post-editing times in a training set, where weights are an exponential function of edit distances between the new segment and those in training data; (b) post-editing time as a linear function of the length of the segment; and (c) source and target statistical language models. These simple estimators outperform strong baselines and are surprisingly competitive compared to more complex estimators, which have many more parameters and combine rich features. These results suggest that before blindly attempting sophisticated machine learning approaches to build post-editing effort predictors, one should first consider simple, intuitive and interpretable models, and only then incrementally improve them by adding new features and gradually increasing their complexity. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) do not seem to be able to improve the performance of the single best estimator, which suggests that more complex, non-linear models could indeed be beneficial when multiple indicators are used.

1 Introduction

Over the last decade, the interest of the industry in machine translation (MT) has grown, mainly as a consequence of high demand and improvements in translation quality. Modern MT systems have proven to lead to productivity gains (Plitt and Masselot, 2010; Guerberof Arenas, 2009) when used to generate draft translations that are then post-edited (corrected) before publishing (Krings and Koby, 2001; O'Brien and Simard, 2014). However, not all the translations produced by MT systems are worth post-editing. In some cases, it would be faster to translate them from scratch. As a result, a strong focus has been put into developing methods for estimating the quality of machine-translated sentences (Blatz et al., 2004; Specia et al., 2009) to identify those translations that may harm productivity if provided to post-editors. Several methods are

proposed every year and compared in the framework of the WMT series of Workshops on Machine Translation.[1]

Most approaches to MT quality estimation (QE) work at the sentence level, although there are also approaches that try to estimate the quality at the word or document levels. Sentence-level QE models predict translation quality in terms of post-editing (PE) time, number of edits needed, and other related metrics (Specia, 2011; Bojar et al., 2014). This paper focuses on sentence-level MT QE and measures quality in terms of PE time. This setting has the important advantage that the time predicted for each machine-translated sentence can be directly used to budget a translation job.

As will be discussed below, existing PE time estimators use many parameters and combine rich features extracted from source sentences and their raw MT output, often with the help of one or more pseudo-references obtained using additional MT systems. They are, however, still far from producing human-like predictions (with Pearson correlations between predicted and human effort metrics plateauing around 0.65, (Bojar et al., 2013, 2014)). To try to understand the problem better, we explore the use of three types of very simple, one-parameter, black-box PE time estimators: (a) a weighted average of PE times in the training set, where weights are an exponential function of edit distances computed between the current sentence (source or raw MT) and training sentences (source or raw MT), so that the contribution of nearest examples is more important; (b) a simple model that learns a unit PE time, either per character or per word, and multiplies it by the length of the current sentence (source or raw MT); and (c) logarithmic probabilities obtained by applying a statistical language model of the source or the target language respectively to the source or raw MT.[2]

The results show that some of these very simple models outperform not only rather strong baselines, but also some complex, multi-parameter estimators participating in the WMT13 (Bojar et al., 2013) and WMT14 (Bojar et al., 2014) PE time estimation contests. Results can be taken as an indication that one should take a step back and first analyse simple models with intuitive interpretations, to only then carefully and gradually increase their complexity, before blindly attempting sophisticated machine learning approaches. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) above does not seem to be able to improve the performance of the single best estimator, which may be taken as an indication that more complex, non-linear models should be considered when multiple indicators are used.

2 Settings and models

2.1 Corpora

We have conducted experiments using the data sets for English-to-Spanish (en→es) translation, which are publicly available as part of the quality estimation shared Task 1.3 of WMT13[3] (Bojar et al., 2013) and WMT14[4] (Bojar et al., 2014); Table 1 describes these data sets. For the experiments in this paper the corpora were pre-processed using the vanilla word tokenizer available in the Python NLTK package (Bird et al., 2009).

2.2 Notation and evaluation

The training data consists of a set of N triplets $\{(s_i, \mathrm{MT}(s_i), t_i)\}_{i=1}^{N}$ where s_i is a source sentence, $\mathrm{MT}(s_i)$ its raw MT output, and t_i the time taken to post-edit $\mathrm{MT}(s_i)$ into an adequate

[1] Last edition: `http://www.statmt.org/wmt17/quality-estimation-task.html`

[2] Language model features have already been proven to be the single best predictors in previous work, see e.g. (Felice and Specia, 2012; Shah et al., 2015).

[3] `http://www.statmt.org/wmt13/quality-estimation-task.html`

[4] `http://www.statmt.org/wmt14/quality-estimation-task.html`

	Translation direction	No. of segments Training	Test
WMT13	en→es	803	284
WMT14	en→es	650	208

Table 1: Translation direction and number of training and test instances for the corpora used in the experiments.

translation of s_i. The goal is to predict the PE time for a new set of M source sentences and their translations, $\{(s_j, \mathrm{MT}(s_j))\}_{j=1}^{M}$.

As in the WMT13 and WMT14 contests, performance will be measured over the test set as the *mean absolute error* (MAE) of the prediction $\hat{t}_j$, that is,

$$\mathrm{MAE} = \frac{1}{M}\sum_{j=1}^{M}|\hat{t}_j - t_j|.$$

In addition to this, Pearson's correlation r between the predicted and measured times will also be reported as a secondary comparison metric.

The best parameter for each model will be determined through minimization of the MAE over the training set, as will be explained in the next section.

2.3 Models

In what follows we describe the three one-parameter models we experimented with in order to predict PE time.

2.3.1 Weighted-average model (*Avg*)

This model estimates the PE time needed to turn $\mathrm{MT}(s_j)$ into an adequate translation of s_j as the weighted average

$$\mathrm{Avg}_u(\alpha, x_j) = \sum_{i=1}^{N} w(\alpha, x_i, x_j)\, t_i,$$

controlled by a single parameter α, whose weights $w(\alpha, x_i, x_j)$ depend on edit distances through

$$w(\alpha, x_i, x_j) = e^{-\alpha \mathrm{ED}_u(x_i, x_j)} / \sum_{i=1}^{N} e^{-\alpha \mathrm{ED}_u(x_i, x_j)},$$

where $\mathrm{ED}_u(x_i, x_j)$ is the edit distance between x_i and x_j, u is the unit used to compute it, either characters ($u = \mathrm{c}$) or words ($u = \mathrm{w}$), and x_i (resp. x_j) is either the source sentence s_i (resp. s_j) or its machine translation $\mathrm{MT}(s_i)$ (resp. $\mathrm{MT}(s_j)$). For positive values of α, the contribution $w(\alpha, x_i, x_j)$ of t_i diminishes with the distance between either the source sentences or between their raw machine translations. In particular:

- When $\alpha = 0$, $\mathrm{Avg}_u(0, x_j) = \frac{1}{N}\sum_{i=1}^{N} t_i$ for all j, that is, the arithmetic average of measured PE times; we will refer to this as the *naïve zero-parameter average*;
- When $\alpha \to +\infty$, the t_i corresponding to the minimum $\mathrm{ED}(x_i, x_j)$, that is, the nearest neighbour, is selected. In what follows, this predictor will be referred to as $\mathrm{NN}_u(x_j)$.

It is expected that a careful choice of α in $[0, +\infty)$ will give a better estimate by assigning a higher weight to closer examples. The weighted average effectively acts as a "soft nearest-neighbour" predictor.

To find the optimum value of α, the training corpus is randomly split in two sets: 80% of the samples are used to compute the edit distances and the remaining 20% are used as a development set.

The idea behind the weighted-average model bears some resemblance to the work by Béchara et al. (2016), where a *semantic textual similarity* (between the source sentences) is used to select a close example: instead of predicting time, Béchara et al. (2016) predict the BLEU score for sentences that do not have a reference translation available, using as reference that for the close example. Note the weighted-average model is clearly a *black-box* model, as it does have access to the inner workings of the MT system whose quality is being predicted. It is also an *example-based* model that computes a prediction for the current segment by looking up measured times for existing segments in a training set.

2.3.2 Models based on the PE time per segment length unit (*TLen*)

These very simple, one-parameter estimators predict the PE time t_j as

$$\mathrm{TLen}_u(a, x_j) = a\,\mathrm{len}_u(x_j),$$

where x_j is a source sentence s_j, or its machine translation $\mathrm{MT}(s_j)$, and $\mathrm{len}_u(x_j)$ is the length of x_j in characters ($u = \mathrm{c}$) or words ($u = \mathrm{w}$). Note that the coefficient a, which is obtained by directly minimizing the MAE over the whole training corpus, has an easy interpretation in seconds per character or seconds per word, respectively. Again, this is a *black-box* model, which, in addition, only looks at one property of the source or machine-translated segment: its length. When $x_j = s_j$, it simply predicts that PE time grows linearly with the source sentence. When $x_j = \mathrm{MT}(s_j)$, the estimate is similar if one assumes that target-segment length grows linearly with source-segment length. Note, however, that this predictor pays very little attention to the actual post-editability of the translation:

- *Any* MT output having the same length would have the same post-editing time, regardless of the actual target words.
- Truncated or abnormally short MT outputs would be consistently —and often incorrectly— estimated to be easier to post-edit.

These models are therefore expected to be very limited predictors of PE time.

2.3.3 Statistical language models

Source-language (SLM) and target-language models (TLM), trained on a subset of the WMT13 translation task data[5] (an interpolated combination of Europarl and News Commentary data) were used to compute the logarithm of the probability of s_j and $\mathrm{MT}(s_j)$, respectively. This is then multiplied by a coefficient a which is also optimized to minimize MAE on the whole training set. Language models are common indicators used in QE but also have important limitations as PE time predictors:

- A TLM basically measures the *fluency* of the translation (Specia et al., 2013, p. 80), and would estimate more fluent translations as easier to post-edit, regardless of their actual semantic relationship to the source sentence.
- A SLM would in contrast measure the *complexity* of the translation (Specia et al., 2013, p. 80), or, if the language model was trained on texts similar to those on which the MT system was trained, its *expectedness*. Nevertheless, its predictive power may be limited when applied to a system that was not trained on similar data (or to a rule-based system).

[5] `http://www.statmt.org/wmt13/translation-task.html`

It is however worth mentioning that language models are amongst the best performing features for sentence-level MT QE (Felice and Specia, 2012; Shah et al., 2015) and are therefore included in most models submitted to the WMT QE shared tasks.

3 Results and discussion

3.1 Performance of one-parameter predictors

Tables 2 and 3 summarize the results for the one-parameter models, placing them in the context of the results obtained by other WMT13 and WMT14 participants. The performance of the zero-parameter naïve average, that is, the one obtained using for all test segments the average time in the training set as a fixed estimate, and the four nearest-neighbour estimates $\mathrm{NN}_u(x_j)$ (see Section 2.3.1) are also provided for completeness. The main metric used in the discussion is MAE, the official metric in WMT13 and WMT14. Pearson correlations, also provided, roughly follow the same trend, and their comparison would lead to similar conclusions (but see Section 3.1.3 for a more detailed discussion).

3.1.1 WMT13 results

When ordering results by MAE, as in (Bojar et al., 2013), the one-parameter models (*Avg*, *TLen*, SLM and TLM) outperform at least 2 of the 14 participants, with *TLen* models actually outperforming 8 of them and the TLM outperforming 12 of them. The baseline system (Baseline bb17 SVR), using support vector regression and a well-known set of 17 black-box features (Specia et al., 2013) also outperforms 8 of the 14 participant models. It is worth mentioning that language models are also included as features in this baseline set; that is, the baseline system is a superset of the single-parameter models using LM features. Nevertheless, the TLM outperforms the baseline by a rather large margin. This result in particular may reveal problems not only present in the baseline but also in other participating submissions such as (a) additional features adding noise that the learning algorithm could not adequately handle, (b) the regression architecture used (for instance, support vector regression in the case of the baseline) not being adequate, (c) optimization not being good enough (for instance, due to an incorrect choice of hyperparameters or to incomplete convergence), or (d) over-fitting to a rather small training set. All these reasons are in principle possible and worth a closer examination. One of the participating systems is even outperformed by the naïve-average zero-parameter estimate, and two of them by one of the (also parameterless) nearest-neighbour estimates.

In general terms, computationally simpler (linear) *TLen* models perform better than the more complex (sum of exponentials containing edit distances) *Avg* models, while the outstanding performance of TLM and SLM may be explained by the fact that they were trained on the same data as the system whose quality was estimated — therefore, in this last case, the black-box assumption would not hold entirely.

3.1.2 WMT14 results

When ordering results by MAE, as in (Bojar et al., 2014), one-parameter models have a more modest performance in this dataset, beating only 3 out of the 10 submissions: one of them (FBK-UPV-UEDIN/NOWP), which uses hundreds of features obtained from the best 100,000 translations produced by a purposely-trained statistical MT system; another one, the baseline, a rather strong model (17 features), equivalent to the WMT13 baseline. Contrary to what happened for WMT13, character-level *Avg* models seem to perform slightly better than the *TLen* model and the SLM and TLM models; these language models were trained on the same data as for WMT13, whereas the MT systems evaluated in WMT14 were not. All zero-parameter models (naïve average, nearest-neighbour) rank below all participants.

We note that the performance of the official baseline system (Baseline bb17 SVR) is

particularly poor on this data set. The reason for that were the ranges used for the grid search to optimize the hyperparameters of the support vector machine model, which were different from those used in the WMT13 model. If the same ranges are used, the baseline reaches a MAE of 17.65, which would place it above all of the one-parameter models and above two of the participating systems. This issue shows further evidence that more complex models need to be carefully crafted, with special attention dedicated to their hyperparameters.

3.1.3 Analysis

How can length be such a reasonable estimator? In both datasets, length-based $\mathrm{TLen}_u(x_j)$ estimators show a rather competitive performance, in spite of the obvious limitations discussed in Section 2.3.2. This may be due to the fact that the output of a single MT system was post-edited and, therefore MT quality and, consequently, the post-editing effort across the segments produced by the MT systems is quite stable, effectively yielding a roughly constant per-word or per-character post-editing time and therefore making length a reasonable estimator in this case.[6] It would therefore be reasonable to expect performance to have been clearly worse if output from at least two MT systems with very different levels of quality had been post-edited.

Pearson correlations between predictions. In addition to the Pearson correlation with the test set, we have computed the Pearson correlation coefficient between the predictions of participating submissions —which are available at the WMT13[7] and WMT14[8] websites— and our best one-parameter *TLen*, *Avg*, and TLM models. In general terms, systems showing a good correlation with the one-parameter models happen to perform similarly, an indication that their predictions are very similar for test sentences. There are, however, interesting exceptions. An example of moderate correlation among predictors but similar performance is the SHEF FS submission to WMT13, which has correlation coefficient with $\mathrm{Avg}_{\mathrm{c}}(\alpha = 0.256, \mathrm{MT}(s_j))$ of 0.67 and an absolute difference in MAE of only 0.70. This may point at a certain complementarity between the two predictors, which seem to predict differently for many test sentences in spite of similar MAE performance. Another remarkable exception is the LIMSI elastic submission to WMT13, which correlates reasonably well with $\mathrm{TLen}_{\mathrm{w}}(a = 3.226, \mathrm{MT}(s_j))$ $(r = 0.74)$, $\mathrm{Avg}_{\mathrm{c}}(\alpha = 0.256, \mathrm{MT}(s_j))$ $(r = 0.75)$, and specially $\mathrm{TLM}(a = -1.421, \mathrm{MT}(s_j))$ $(r = 0.85)$ but performs considerably worse (the absolute differences in MAE are 18.6, 14.0, and 21.8 respectively). As we discuss in what follows, this could be an issue related to inadequate scaling of predictions.

Correlation and MAE leading to different system rankings: A scaling study. There are cases in which using the Pearson correlation obtained by participants does not lead to the same system ranking as that obtained by the official MAE-based ranking. One such a case is the LIMSI elastic submission, which has a Pearson correlation in the range of participants getting much lower MAE. Another interesting case is that of FBK-UPV-UEDIN/WP, FBK-UPV UEDIN/NOWP and RTM-DCU/RTM-RR in WMT14. Again, their Pearson correlation coefficients are clearly higher than those of other participants having similar MAE.

Discrepancies between MAE and correlation coefficients may easily be explained in terms of scaling; in fact, by simply scaling the outputs of all participating predictors one can obtain better MAE results, as shown in tables 2 and 3.

[6] The actual time per unit, both in the training set and the test set, indeed shows a rather peaked distribution density around the average values used by the length predictors.

[7] `http://www.statmt.org/wmt13/quality_estimation_data/QE_WMT13_submissions_task1.3_sentence.zip`

[8] `http://www.statmt.org/wmt14/quality_estimation_data/QE_WMT14_submissions_task1.3_sentence.zip`

System ID	MAE	r	Scaling	Scaled MAE	ΔMAE
FBK-UEDIN Extra	47.5	0.65	0.909	46.5	1.0
FBK-UEDIN Rand-SVR	47.9	0.66	1.062	47.6	0.3
$\mathrm{TLM}(a=-1.421, \mathrm{MT}(s_j))$	48.8	0.65			
CNGL SVR	49.2	0.67	1.164	47.6	1.6
CNGL SVRPLS	49.6	0.68	1.104	48.9	0.7
$\mathrm{SLM}(a=-1.249, s_j))$	49.7	0.64			
CMU slim	51.6	0.63	0.902	50.6	1.0
Baseline bb17 SVR	51.9	0.61	1.103	51.4	0.5
$\mathrm{TLen_w}(a=3.226, \mathrm{MT}(s_j))$	52.0	0.57			
$\mathrm{TLen_w}(a=3.468, s_j)$	52.3	0.59			
DFKI linear6	52.4	0.64	0.857	50.7	1.7
$\mathrm{TLen_c}(a=0.664, s_j)$	52.4	0.57			
$\mathrm{TLen_c}(a=0.601, \mathrm{MT}(s_j))$	52.5	0.57			
CMU full	53.6	0.58	1.006	53.6	0.0
DFKI pls8	53.6	0.59	0.874	52.1	1.5
TCD-DCU-CNGL SVM2	55.8	0.47	1.082	55.4	0.4
TCD-DCU-CNGL SVM1	55.9	0.48	1.083	55.5	0.4
SHEF FS	55.9	0.42	0.870	54.7	1.2
$\mathrm{Avg_c}(\alpha=0.256, \mathrm{MT}(s_j))$	56.6	0.53			
$\mathrm{Avg_c}(\alpha=0.386, s_j)$	57.2	0.56			
$\mathrm{Avg_w}(\alpha=1.079, \mathrm{MT}(s_j))$	61.1	0.52			
$\mathrm{Avg_w}(\alpha=0.612, s_j)$	61.7	0.59			
$\mathrm{NN}_c(s_j)$	62.5	0.41			
SHEF FS-AL	64.6	0.57	1.054	64.4	0.2
$\mathrm{NN}_c(\mathrm{MT}(s_j))$	67.8	0.35			
Naïve zero-parameter average	68.1	—			
$\mathrm{NN}_w(s_j)$	70.1	0.37			
LIMSI elastic	70.6	0.58	1.804	54.4	26.2
$\mathrm{NN}_w(\mathrm{MT}(s_j))$	71.3	0.30			

Table 2: Mean absolute error (MAE) and Pearson correlation coefficient (r) for one-parameter ($\mathrm{Avg}(\alpha, x_j)$, $\mathrm{TLen}(a, x_j)$, $\mathrm{SLM}(a, s_j)$ and $\mathrm{TLM}(a, \mathrm{MT}(s_j))$) and zero-parameter (naïve average, $\mathrm{NN}_u(x_j)$) quality estimators (all shaded) in the context of WMT13 submissions. For WMT13 participants, the results of *oracle scaling* (see text) are also given: scaling factor, new MAE and variation of MAE.

System ID	MAE	r	Scaling	Scaled MAE	ΔMAE
RTM-DCU/RTM-SVR	16.77	0.63	0.863	16.29	0.48
MULTILIZER/MLZ2	17.07	0.64	0.851	16.22	0.75
SHEFF-lite	17.13	0.61	0.949	17.05	0.08
MULTILIZER/MLZ1	17.31	0.65	0.835	16.43	0.88
SHEFF-lite/sparse	17.42	0.61	0.963	17.38	0.04
FBK-UPV-UEDIN/WP	17.48	0.66	0.812	15.76	1.72
RTM-DCU/RTM-RR	17.50	0.64	0.814	16.16	1.34
$\mathrm{Avg_c}(\alpha = 0.217, s_j)$	17.69	0.58			
$\mathrm{Avg_c}(\alpha = 0.202, \mathrm{MT}(s_j))$	17.94	0.57			
$\mathrm{TLM}(a = -0.538, \mathrm{MT}(s_j))$	18.38	0.57			
$\mathrm{TLen_c}(a = 0.281, \mathrm{MT}(s_j))$	18.55	0.58			
$\mathrm{SLM}(a = -0.521, s_j)$	18.59	0.55			
$\mathrm{TLen_w}(a = 1.519, MT(s_j))$	18.66	0.55			
FBK-UPV-UEDIN/NOWP	18.69	0.62	0.758	16.72	1.97
$\mathrm{Avg_w}(\alpha = 0.794, \mathrm{MT}(s_j))$	18.75	0.54			
$\mathrm{TLen_c}(a = 0.327, s_j)$	18.80	0.59			
$\mathrm{TLen_w}(a = 1.616, s_j)$	18.84	0.56			
$\mathrm{Avg_w}(\alpha = 0.61, s_j)$	18.86	0.56			
USHEFF	21.48	0.57	0.907	21.25	0.23
Baseline bb17 SVR	21.49	0.54	0.906	21.25	0.24
$\mathrm{NN}_c(s_j)$	21.53	0.37			
$\mathrm{NN}_w(\mathrm{MT}(s_j))$	21.80	0.36			
Naïve zero-parameter average	21.93	—			
$\mathrm{NN}_w(s_j)$	22.14	0.31			
$\mathrm{NN}_c(\mathrm{MT}(s_j))$	22.65	0.32			

Table 3: Mean absolute error (MAE) and Pearson correlation coefficient (r) for one-parameter ($\mathrm{Avg}(\alpha, x_j)$, $\mathrm{TLen}(a, x_j)$, $\mathrm{SLM}(a, s_j)$ and $\mathrm{TLM}(a, \mathrm{MT}(s_j))$) and zero-parameter (naïve average, $\mathrm{NN}_u(x_j)$) black-box quality estimators (all shaded) in the context of WMT14 submissions. For WMT14 participants, the results of *oracle scaling* (see text) are also given: scaling factor, new MAE and variation of MAE.

In the case of the LIMSI elastic submission to WMT13, the scaling factor of 1.804 leads to the best possible test-set MAE of 54.4, which is much better and closer to that of other systems having similar Pearson's coefficients. Note that this is an *oracle scaling*, since the gold-standard time measurements for the test set are used to obtain the best scaling factor; however, it is reasonable to expect that linear scaling on the training set would also have improved this predictor. For the remaining participants in WMT13, *oracle scaling* factors in the range $[0.857, 1.164]$ lead to small changes in MAE between 0.3 and 1.7 seconds, that is, around 0.6% to 3.4%. These changes would be expected to be even smaller or even negligible if scaling had been learned on the training set.

The scaling picture for WMT14 is also interesting (see Table 3). Oracle scaling factors in the range $[0.758, 0.949]$ lead to improvements in MAE in the range $[0.24\ s, 1.97\ s]$, which are sometimes as large as 12%. The improvements are particularly substantial for FBK-UPV-UEDIN/NOWP (–1.97 s, scaling 0.758), FBK-UPV-UEDIN/WP (–1.72 s, scaling 0.812) and RTM-DCU/RTM-RR (–1.34 s, scaling 0.814), which would explain the discrepancies between Pearson correlation and MAE mentioned above. It is reasonable to expect that a scaling factor obtained using the training set would have also made a difference in the test-set MAE in these three cases.

Approximating complex predictors with just one parameter: Finally, it is worth noting that some systems showing a good Pearson correlation with the models presented in this paper use very many features and parameters. In particular, the Pearson correlation coefficient of the RTM-DCU/RTM-SVR submission to WMT14 with $\mathrm{TLen_c}(a = 0.281, \mathrm{MT}(s_j))$ is 0.90 (the absolute difference in MAE is 1.78) and, while the latter has one feature and a single parameter, the former uses hundreds of features and several other sources of information. Oracle scaling of RTM-DCU/RTM-SVR slightly improves its test-set MAE to 16.23 s.

3.2 Performance of few-parameter predictors

In view of the surprisingly competitive results obtained with some of the single-parameter models presented here, one would immediately ask the following question: would performance improve further by using linear combinations of them?

We take the following six linear predicting features: the length-based $\mathrm{TLen_c}(s_j)$, $\mathrm{TLen_w}(s_j)$, $\mathrm{TLen_c}(\mathrm{MT}(s_j))$, and $\mathrm{TLen_c}(\mathrm{MT}(s_j))$, and the two statistical-language models SLM and TLM. For the study, we leave aside the weighted-average features as they are computationally more intensive to use and to train, do not have a linear form, and need a separate development set to be trained.

All $2^6 - 1 = 63$ possible subsets of these 6 features are studied.[9] We take linear combinations of each subset and use the multidimensional downhill simplex algorithm of Nelder and Mead (1965) as implemented in the Python library `scipy` to search the coefficients that minimize the training set MAE. For more than two parameters, the result of the minimization heavily depends on the starting point (this is expected in view of the strong collinearity, for instance, between length features). Therefore, and to ensure the best possible training set MAE, for each subset, 50 searches are performed with starting parameters randomly sampled from the zero-average, unit-variance normal distribution $\mathcal{N}(0, 1)$. The results are shown in Table 4.

As expected, the lowest training set MAE is found when all six features are used; however, the resulting test set MAE does not improve the results obtained with the best single-parameter predictor: 48.8 s for WMT13 (same as TLM alone) and 18.39 s for WMT14 (almost the same as TLM alone). Conversely, some combinations having worse training-set MAE get better test set MAE results, such as 48.22 s for a mixture of just $\mathrm{TLen_w}(s_j)$ and $\mathrm{TLM}(\mathrm{MT}(s_j))$ in WMT13,

[9] Exhaustive search in feature spaces is sometimes performed in QE, e.g. (Scarton et al., 2015).

Dataset	Features	Best combination	Train MAE	Test MAE
WMT13	1	TLM	41.3	48.8
	2	$\mathrm{TLen_w}(s_i) + \mathrm{TLM}$	41.0	48.2
	3	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLM}$	40.7	49.1
	4	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM} + \mathrm{TLM}$	40.6	48.6
	5	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_w}(\mathrm{MT}(s_i)) + \mathrm{SLM} + \mathrm{TLM}$	40.5	48.8
	6	All 6	40.5	48.8
WMT14	1	SLM	15.92	18.59
	2	$\mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.60	18.44
	3	$\mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.57	18.51
	4	$\mathrm{TLen_c}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.53	18.40
	5	$\mathrm{TLen_c}(s_i) + \mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.53	18.40
	6	All 6	15.53	18.39

Table 4: Post-editing time prediction using a small number of linear features: number of features, best combination, training-set MAE, and test-set MAE.

or 18.2 s for a mixture of $\mathrm{TLen_w}(s_j)$, $\mathrm{TLen_c}(\mathrm{MT}(s_j))$ and $\mathrm{TLM}(\mathrm{MT}(s_j))$ for WMT14. These results may be a possible indication of over-fitting or a limitation of a simple linear regressor.

3.3 Budgeting translation jobs

An interesting use of PE time predictors is *budgeting* a PE job, when post-editors are paid by the hour. Given a new translation job, an estimate of time to complete that job may easily be obtained by summing up the predicted PE time over all segments. This is a very practical application of QE.

Disregarding the actual hourly rate (a constant factor), a good estimate of the usefulness for budgeting may be given by studying the Pearson correlation between the total time predicted for a job by a certain estimator and the actual total time for that job.

To simulate that, we repeatedly and randomly extract PE jobs $\{(s_j, \mathrm{MT}(s_j), t_j)\}_{j=1}^{n}$ of $n = 100$ sentences from each of the test sets without replacement. Over each one of these sets, we compute the Pearson correlation between the predicted total time and the actual total measured time. The actual regression coefficients obtained vary with the number of random jobs, but their values for job sizes of 0.4, 0.8, 1.0, and 2.0 times the size of the test set and for a fixed number of 1000 jobs show consistent relative trends. The results for a number of jobs equal to the number of segments in the test set are shown in Table 5.

As can be seen, the Pearson correlation reported for the best single-parameter predictors is almost the same as that for the winning system in WMT13, and slightly worse in WMT14. This would suggest that, at least for these datasets, simple predictors could be used instead of very complex predictors having a large number of features and parameters with a very small loss in budgeting accuracy.

Dataset	Predictor	r
WMT13	FBK-UEDIN Extra (winner)	0.867
	$\mathrm{TLM}(\mathrm{MT}(s_j))$	0.856
	$\mathrm{SLM}(s_j)$	0.854
	$\mathrm{TLen_w}(\mathrm{MT}(s_j))$	0.856
	$\mathrm{Avg_c}(\mathrm{MT}(s_j))$	0.806
	Baseline	0.849
WMT14	RTM-DCU/RTM-SVR (winner)	0.860
	$\mathrm{Avg_c}(s_j)$	0.821
	$\mathrm{TLM}(\mathrm{MT}(s_j))$	0.828
	$\mathrm{TLen_c}(\mathrm{MT}(s_j))$	0.827
	$\mathrm{SLM}(s_j)$	0.819
	Baseline	0.730

Table 5: *Budgeting* Pearson correlation coefficients for selected PE time predictors, computed for a number of random jobs equal to the number of segments in the test set.

4 Concluding remarks

The results obtained by very simple, one-parameter MT QE models happen to be surprisingly competitive with those obtained by complex QE models using strong learning algorithms, tens, hundreds or thousands of features, and, sometimes, additional resources such as existing, custom-trained, or external MT systems. The findings in this study lead us to make the following recommendations for researchers in MT QE:

- First, look at what can be done with very simple models before *using a sledgehammer to crack nuts*, in order to get an idea of the performance one could obtain and hopefully improve. As some of the features used in the simple models proposed here are usually part of participants' complex models, the modest performance they obtain may be due to noise introduced by new features that could not be filtered out by the regressors (probably as a result of a non-optimal training process), to learning problems such as over-fitting to the training set, to non-optimal hyper-parameter choice, to incomplete convergence, or to the shortcomings of the regressors used (as revealed by the oracle scaling described in Section 3.1.3); the actual reasons are probably worth a closer analysis.
- Then, incrementally explore more complex models; linear combinations of a few carefully selected features do not seem to help much; therefore, one should probably consider simple non-linear models. The results of this analysis may be expected to shed some light on the problem.

Finally, a better understanding of the contribution of each feature to the QE models using them could open the door to using, in real-life QE scenarios, feasible and computationally simpler predictors.

Acknowledgements: Work supported by the Spanish government through the EFFORTUNE (TIN2015-69632-R) project and through grant PRX16/00043 for Mikel L. Forcada, and by the European Commission through the QT21 project (H2020 No. 645452).

References

Béchara, H., Parra-Escartín, C., Orăsan, C., and Specia, L. (2016). Semantic textual similarity in quality estimation. *Baltic J. Modern Computing*, 4(2):256–268.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 315–321.

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 *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, MD, USA.

Felice, M. and Specia, L. (2012). Linguistic features for quality estimation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 96–103, Montreal, Quebec, Canada. Association for Computational Linguistics.

Guerberof Arenas, A. (2009). Productivity and quality in the post-editing of outputs from translation memories and machine translation. *The International Journal of Localisation*, 7(1):11–21.

Krings, H. P. and Koby, G. S. (2001). *Repairing texts: empirical investigations of machine translation post-editing processes*, volume 5. Kent State University Press.

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

O'Brien, S. and Simard, M. (2014). Introduction to special issue on post-editing. *Machine Translation*, 28(3-4):159–164.

Plitt, M. and Masselot, F. (2010). A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, (93):7–16.

Scarton, C., Tan, L., and Specia, L. (2015). USHEF and USAAR-USHEF participation in the WMT15 QE shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 336–341.

Shah, K., Cohn, T., and Specia, L. (2015). A Bayesian non-linear method for feature selection in machine translation quality estimation. *Machine Translation*, 29(2):101–125.

Specia, L. (2011). Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.

Specia, L., Shah, K., De Souza, J. G., and Cohn, T. (2013). QuEst - a translation quality estimation framework. In *Proceedings of the Conference of the Association of Computational Linguistics (Conference System Demonstrations)*, pages 79–84.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Conference of the European Association for Machine Translation*, pages 28–37, Barcelona, Spain.