Empirical Study of a Two-Step Approach to Estimate Translation Quality

Jesús González-Rubio[†], *J. Ramón Navarro-Cerdán*[‡], *Francisco Casacuberta*[†]

[†]D. de sistemas informáticos y computación [‡]Inst. Tecnológico de Informática
Universitat Politècnica de València, camino de Vera s/n, 46022 Valencia, Spain
jegonzalez@dsic.upv.es, jonacer@iti.upv.es, fcn@dsic.upv.es

Abstract

We present a method to estimate the quality of automatic translations when reference translations are not available. Quality estimation is addressed as a two-step regression problem where multiple features are combined to predict a quality score. Given a set of features, we aim at automatically extracting the variables that better explain translation quality, and use them to predict the quality score. The soundness of our approach is assessed by the encouraging results obtained in an exhaustive experimentation with several feature sets. Moreover, the studied approach is highly-scalable allowing us to employ hundreds of features to predict translation quality.

1. Introduction

Despite an intensive research in the last fifty years, machine translation (MT) systems are still far from perfect [1]. Hence, a desirable feature to improve their practical deployment is the capability of predicting at run-time[1] the reliability of the generated translations. This task, referred to as quality estimation [2] (QE), is becoming a crucial component in practical MT systems [3, 1]. For instance, to decide if an automatic translation is worth being supervised by a translator or it should be translated from scratch. Quality can be estimated at the word, sentence, or document level. Here, we focus on the estimation of sentence-level quality.

Sentence-level QE is typically addressed as a regression problem [4, 2]. Given a translation (and other sources of information), a set of features is extracted and used to build a model that predicts a quality score. This point of view provides a solid framework within which accurate predictors can be derived. However, several problems arise when applying this approach to predict the quality of natural language sentences. For example, while the concept of translation quality is quite intuitive, the definition of features that reliably account for it has proven to be elusive [4, 1]. Thus, in practice, feature sets contain a large number of noisy, collinear and ambiguous features that hinder the learning process of the regression models, e.g., due to the "curse of dimensionality" [5].

An interesting approach to overcome these problems is to conceive QE as a two-step problem. In a first step, a dimensionality reduction (DR) process strips out the noise present in the original features returning a reduced set of (potentially new) features. Then, the actual quality prediction is made from this reduced set. Typically, QE systems reduce the dimensionality by simply selecting a subset of the original features according to some relevance measure [2, 6, 7]. However, a recent study [8] have shown that DR methods based on a projection of the original features may be more effective. The intuition for this is clear, the new features extracted by a projection-based DR method summarize the "information" contained in the all the original features, in contrast, the information contained in the features discarded by a feature selection method is inevitably lost.

We work on the foundations of [8] and provide an exhaustive empirical study of the most successful QE approach described there. This approach (§2) involves a DR method based on a partial least squares [9] (PLS) projection of the data and a support vector machine [10] (SVM) as prediction model. We test this two-step QE approach in a wide variety of conditions (§3) where we compare the performance of PLS to the most widely-used projection-based DR approach, namely principal component analysis [11] (PCA). Empirical results (§4) show that PLS consistently outperformed PCA in prediction accuracy and feature reduction ratio. This latter result is particularly interesting because it allows us to apply QE in scenarios with strict temporal restrictions, for instance interactive machine translation tasks.

[1] That is, in the absence of reference translations.

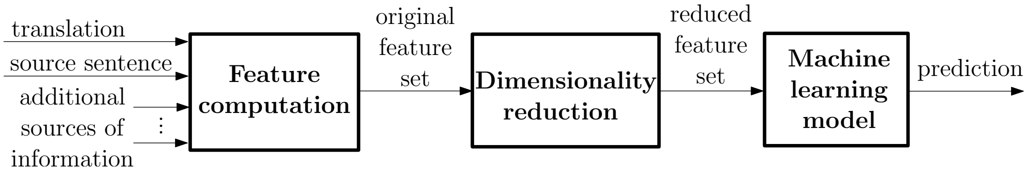

Figure 1: Dataflow of the studied two-step QE approach.

2. A Two-Step QE Approach

The method proposed in [8] divide QE ($\mathbb{R}^m \to \mathbb{R}$) into two sub-problems. First, the original m-dimensional set of features is projected into a new r-dimensional set of features ($\mathbb{R}^m \to \mathbb{R}^r, r < m$). Then, this reduced feature set is used to build a regression model that predicts the actual quality scores ($\mathbb{R}^r \to \mathbb{R}$). Figure 1 shows a diagram of this two-step training methodology. Next sections describe how to solve these two sub-problems.

2.1. Dimensionality Reduction

Typical approaches to reduce a set of noisy features involve the use of principal components analysis [11] (PCA). PCA projects the set of features into a set of principal components (PCs) where each PC explains the variability of the features in one principal direction. As a result, these PCs contain almost no redundancy but, since the PCA transformation ignore the quality scores to be predicted, they do not necessarily have to be the best features to perform the prediction.

Instead, we implement a feature reduction technique based on partial least squares [9] (PLS). PLS extracts a ordered set of latent variables (LVs) such that each of them accounts for the maximum possible co-variability between the features and the scores to be predicted under the constraint of being uncorrelated with previous LVs. That is, LVs are uncorrelated as PCs do, and additionally, they explain as much as the variability in the quality scores as possible. As a result, usually few LVs than PCs are required to reach a certain accuracy.

Let $\{\mathrm{x}_i, y_i\}_{i=0}^{n}$ be a corpus with n samples where x_i are m-dimensional feature vectors, and y_i are quality scores. This corpus can be written in matrix form where symbol $^\intercal$ indicates the transpose of a matrix or vector:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\intercal \\ \vdots \\ \mathbf{x}_n^\intercal \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{1}$$

Then, PLS constructs the following linear model where $\mathbf{b}$ is a vector of regressor coefficients, and $\mathbf{f}$ is a vector of zero-centered Gaussian errors:

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{f} \tag{2}$$

PLS also defines two PCA-like transformations ($\mathbf{P}$ for $\mathbf{X}$, and $\mathbf{q}$ for $\mathbf{y}$) with $\mathbf{E}$ and $\mathbf{f}$ being the corresponding errors, and a linear relation $\mathbf{R}$ linking both blocks:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\intercal + \mathbf{E} \quad \mathbf{y} = \mathbf{U}\mathbf{q}^\intercal + \mathbf{f} \qquad \mathbf{U} = \mathbf{T}\mathbf{R} \tag{3}$$

where matrices $\mathbf{T}$ and $\mathbf{U}$ are the projections of $\mathbf{X}$ and $\mathbf{y}$ respectively. The value of the regression coefficients $\mathbf{b}$ are finally computed as [9]:

$$\mathbf{b} = \mathbf{R}\mathbf{q}^\intercal \quad \text{where} \quad \mathbf{R} = \mathbf{W}(\mathbf{P}^\intercal\mathbf{W})^{-1} \tag{4}$$

where $\mathbf{W}$ is a weight matrix that accounts for the correlation between $\mathbf{X}$ and $\mathbf{U}$.

The columns in matrix $\mathbf{T}$ are the LVs of $\mathbf{X}$. Each of these LVs accounts for the maximum co-variability between $\mathbf{X}$ and $\mathbf{y}$ not explained by previous LVs. Therefore, similarly as it is usually done with PCA, we can collect the first r LVs and use them to represent the original m-dimensional feature set. Given that $r < m$, and that the LVs are orthogonal by definition, we are simultaneously addressing the "curse of dimensionality" and reducing the noise present in the original features. Moreover, the reduced set also explains most of the variability in the quality scores to be predicted.

In the experiments, we used the `pls` library [12] of the R toolkit. The dimension of the reduced set r is one of the meta-parameters of the studied QE approach.

PLS can be directly used as a predictor model (see Equation (2)). However, its simple linear model is not adequate to model the nonlinear relation that may exist between the features and the quality scores. Preliminary experiments confirmed this intuition.

2.2. Prediction Model

Once the reduced feature set is extracted, a support vector machine (SVM) is used predict the quality scores ($\mathbb{R}^r \to \mathbb{R}$). We choose SVMs because they have shown good prediction accuracy and robustness when dealing with noisy data in a number of tasks.

SVMs, first proposed for classification problems by Cortes and Vapnik [10], are a class of machine learning models that are able to model nonlinear relations between the features and the values to be predicted. Prior to any calculation, SVMs project the data into an alternative space. This projection, defined by a kernel function, may be nonlinear; thus, though a linear relationship is learned in the projected feature space, this relationship may be nonlinear in the original space. Following previous works on QE [2], we use SVMs with a radial basis kernel as implemented in the `LibSVM` package [13]. Values γ, ϵ, and C are additional meta-parameters to be optimized.

3. Experimental Setup

3.1. Corpus

We computed quality scores for the English-Spanish news evaluation data used in the QE task of the 2012 workshop on statistical MT [1] (WMT12). The Spanish translations were generated by a phrase-based MT system trained on the Europarl and News Commentaries corpora as provided for the WMT12 translation task. Evaluation data contains 1832 translations for training, and 422 translations for test. The quality score of each translation $\{y \in \mathbb{R} \mid 1 \leq y \leq 5\}$ is computed as the average of the scores given manually by three different experts in terms of post-editing effort:

5: The translation requires little editing to be publishable
4: 10%–25% of the translation needs to be edited
3: 25%–50% of the translation needs to be edited
2: 50%–70% of the translation needs to be edited
1: The translation must be translated from scratch

3.2. Feature Sets

We conducted QE experiments with several feature sets submitted to the WMT12 QE task[2]. These sets allow us to test our approach under a wide variety of conditions. Table 1 displays, for each set, the number of features, whether or not the features are result of a feature selection process, the percentage of features in the training partition that are collinear with the rest of features (redundancy), and the percentage of features in the training partition that are constant, and hence, irrelevant to perform the prediction. We estimated the degree of collinearity of each feature by its condition number considering a value above 100 to denote collinearity [14]

Name	#features	feature selection?	collinear features	constant features
DCU-SYMC	308	no	34.6%	0.7%
LORIA	49	yes	12.2%	0.0%
SDLLW	15	yes	0.0%	0.0%
TCD	43	no	18.6%	0.0%
UEDIN	56	no	5.5%	1.8%
UPV	497	no	54.3%	6.8%
UU	82	no	7.5%	2.5%
WLV-SHEF	147	no	21.0%	2.7%

Table 1: Main properties of the feature sets. We estimated the collinearity with the condition number [14].

We consider the feature sets as independent corpora provided by an external agent. Hence, and due to space limitations, we only provide a brief description of each set; an exhaustive description can be found in the corresponding citation. Many of the sets include the 17 baseline features provided by the organizers [1].

DCU-SYMC: [15] 308 features including features based on latent Dirichlet allocation; source grammatical features from the TreeTagger part-of-speech tagger, an English grammar, the XLE parser, and the Brown re-ranking parser; and target TreeTagger features.

LORIA: [6] 66 features including the baseline features, and features based on cross-lingual triggers.

SDLLW: [7] 15 features exhaustively selected from an original set of 45 features: the 17 baseline features, 8 features based on decoder information, and 20 features based on n-gram precisions and word alignments.

TCD: [16] 43 features including the baseline features, and features based on similarity measures with respect to the Google n-grams data set.

UEDIN: [17] 56 features including the baseline features and features based on named entities, morphological information, lexicon probabilities, word-alignments, and sentence and n-grams similarities.

UPV: [18] 497 features including the baseline features and features based on word-level quality scores.

UU: [19] 82 features computed from syntactic, constituency, and dependency trees.

WLV-SHEF: [20] 147 features based on part-of-speech information, subject-verb agreement, phrase constituency and target lexicon analysis.

3.3. Experimental Methodology

For each feature set, a QE system was built following the two-step methodology described in §2 and depicted

[2] These are available in https://github.com/lspecia/QualityEstimation.

Feature set	Baseline		PCA			Our approach		
	RMSE	#features	RMSE	#features		RMSE	#features	
DCU-SYMC	0.71±0.02	308	0.70±0.02	82	(26.6%)	**0.62±0.02***	**28**	**(9.1%)**
LORIA	**0.72±0.03**	49	0.75±0.01	43	(87.7%)	**0.72±0.02**	**10**	**(20.4%)**
SDLLW	**0.67±0.02**	15	**0.67±0.02**	15	(100.0%)	**0.67±0.02**	**10**	**(66.7%)**
TCD	0.76±0.01	43	0.74±0.02	24	(55.8%)	**0.72±0.02**	**15**	**(38.9%)**
UEDIN	0.72±0.03	56	0.71±0.02	43	(76.8%)	**0.69±0.02**	**8**	**(14.3%)**
UPV	0.74±0.02	497	0.69±0.02	99	(19.9%)	**0.62±0.02***	**58**	**(11.7%)**
UU	0.72±0.02	82	0.68±0.02	74	(90.2%)	**0.67±0.02**	**29**	**(35.4%)**
WLV-SHEF	0.71±0.02	147	0.71±0.02	91	(61.9%)	**0.65±0.02***	**25**	**(17.0%)**

Table 2: RMSE and number of LVs obtained by cross-validation for the different feature sets. In parenthesis, we show the number of LVs as a percentage of the original features. Baseline denotes a system trained with the whole feature set. PCA denotes a system built using PCA instead of PLS. Best mean RMSE values and lowest number of features are displayed boldface. Asterisks denote a statistically better result than *both* the other two systems (95% confidence).

in Figure 1. All features were standardized by subtracting the feature mean from the raw values, and dividing the difference by the corresponding standard deviation.

The number of LVs (r) was optimized by ten-fold cross-validation using the training partitions (1832 samples). Each cross-validation experiment took eight folds for training (dev-train), one held-out fold for development and the other held-out fold for test (dev-test). We used the dev-train folds to estimate a PLS model. Then, this model was used to extract the r LVs of dev-train, and of the separated development fold and the dev-test fold. Next, we used the reduced dev-train folds to estimate an SVM model, the reduced development fold to optimize the SVM meta-parameters (γ, ϵ, and C), and the reduced dev-test fold to test the optimized SVM model. The result of each complete cross-validation experiment was the averaged prediction accuracy on the ten held-out dev-test folds. The number of LVs was selected to optimize this average accuracy.

Once the number of LVs was fixed, we built a new prediction model with the whole training partition optimizing the SVM meta-parameters by cross-validation. Finally, we used this optimized SVM model to predict the quality scores of the test partitions (422 samples).

3.4. Assessment Criteria

We measure the accuracy of a QE system by the deviation of its predictions $\hat{\mathbf{y}} = \{\hat{y}_1, \ldots, \hat{y}_n\}$ respect to the reference quality scores $\mathbf{y} = \{y_1, \ldots, y_n\}$. Following previous QE works [2, 1], we calculate the root-mean-squared error (RMSE) between them:

$$\text{RMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (5)$$

where n is the number of samples. RMSE quantifies the average error of the estimation with respect to the actual quality score. I.e. the lower the value, the better the performance of the QE system.

Additionally, we perform different significance tests for the reported RMSE results. On the one hand, we obtain confidence intervals for the averaged cross-validation test results with Student's t-tests [21]. On the other hand, we use paired bootstrap re-sampling [22] to measure the significance of the RMSE differences observed between the different methods in the test sets.

4. Results

We now present the results of the empirical evaluation of the studied QE approach. First, we predicted quality scores for each of the feature sets described in §3.2. Then, we took advantage of the scalability of the studied QE approach using jointly all the features in those sets to perform the prediction.

4.1. Results for the Individual Feature Sets

Table 2 shows the cross-validation results (RMSE and number of LVs) obtained for the different feature sets. As a comparison, we present results for SVMs trained with all the features in each set (Baseline), and for systems built using the widespread PCA instead of PLS in the studied two-step training methodology.

We can observe that the studied approach consistently obtained equal or better prediction accuracy (RMSE) than the baseline systems. Additionally, the number of LVs used to build the final SVMs was much lower than the number of original features. The size of the reduced sets varied between two thirds and one tenth

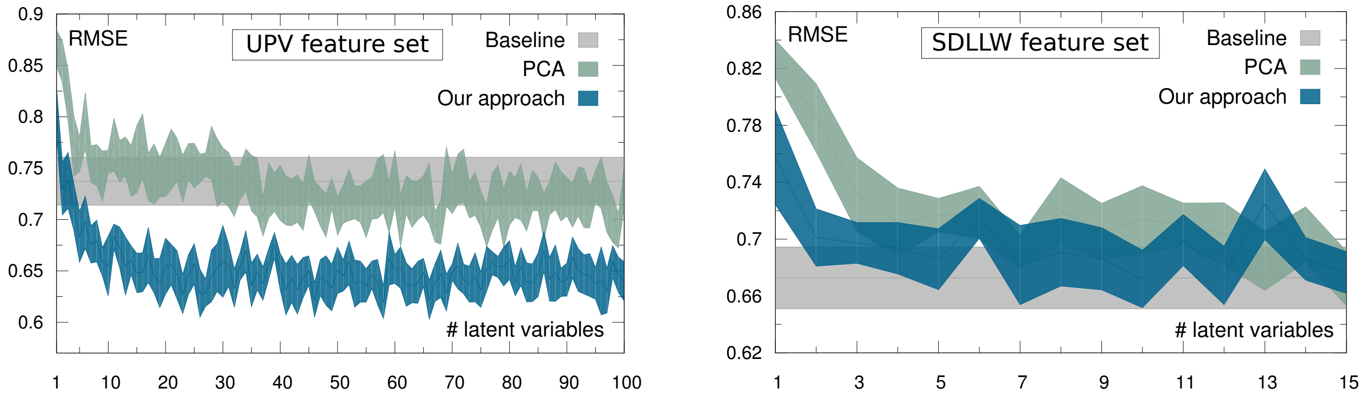

Figure 2: Cross-validation learning curves (RMSE and 95% confidence interval) for two representative feature sets: the highly-redundant UPV set (left), and the concise SDLLW set (right). Baseline denotes the RMSE of systems trained with the whole original feature sets: 497 features for UPV set, and 15 features for SDLLW set.

of the original features. This reductions are roughly related with the percentage of collinear and constant features in Table 1. In comparison to PCA, the studied DR technique, PLS, was able to obtain better prediction accuracy with less features. Usually, the number of LVs is less than half the number of PCs.

These result indicate that the studied QE approach was indeed able to strip out the noise present in the original features. Additionally, the DR technique based on PLS projections showed a better performance (both in prediction accuracy and reduction ratio) that the commonly-used PCA. As a result, even for highly-engineered features sets such as SDLLW [7] that contain no collinear or redundant features, our approach was able to obtain a more compact feature set (10 LVs) that still retained the prediction potential of the whole original set (15 features).

Next, to better understand the influence of the number of LVs in the results, Figure 2 displays the prediction accuracy as a function of the number of features for two prototypical feature sets: the highly noisy and collinear UPV set, and the low redundant SDLLW set.

The prediction accuracy of our method for the UPV feature set (left panel in Figure 2) rapidly improved as more LVs were considered. With only 5 LVs, prediction accuracy already statistically outperformed the baseline (497 features), and it reached its top performance for 58 LVs. As we considered more LVs (for simplicity the graph only shows up to 100 LVs), prediction error steadily increased which was indicative of over-training. Thus, we chose 58 as the optimum number of variables for the UPV set. The quite large RMSE reduction respect to the baseline can be explained by the ability of our approach to strip out the great amount of noise present in the original UPV set, see Table 1. Regarding PCA, it was consistently outperformed by our approach and only slightly improved the RMSE score of the baseline system.

For the concise SDLLW feature set (right panel in Figure 2), our system showed approximately the same behavior: prediction accuracy rapidly improved up to a point from where the performance remains approximately stable. In this case, 10 was the optimal number of LVs. In contrast to the UPV set, our approach could not improve Baseline performance which is reasonable since SDLLW is a very clean set with no redundant or irrelevant features (see Table 1) that could hinder the learning process. Nevertheless, our method was able to obtain the same prediction accuracy as Baseline with only two thirds of the original features.

In a following experiment, we built QE systems with the whole training partitions and the optimal number of LVs estimated in the previous cross-validation experiments. The SVM meta-parameters (γ, ϵ, and C) were optimized by standard cross-validation and the optimized models were used to predict the quality scores of the test partitions. Note that due to variations in the learning procedures, Baseline results may differ from those reported in the WMT12 QE task [1] .

Table 3 displays, for each feature set, the RMSE obtained by our approach in the test partition. We also show baseline results for SVMs built with all the features in each set, and for systems that used PCA instead of PLS to reduce the dimensionality. RMSE confidence intervals for Baseline, PCA and our approach always overlapped but the observed differences were still statistically significant for a number of sets: for DCU-SYMC, Baseline obtained a statistically bet-

Feature set	Baseline	PCA	Our approach
DCU-SYMC	**0.87±0.07***	1.01±0.07	0.96±0.08
LORIA	**0.84±0.06**	0.87±0.06	0.85±0.06
SDLLW	**0.76±0.05**	0.77±0.05	**0.76±0.05**
TCD	**0.82±0.06**	1.00±0.05	0.83±0.06
UEDIN	0.86±0.06	**0.85±0.05**	0.86±0.05
UPV	0.82±0.06	0.83±0.05	**0.78±0.05***
UU	**0.81±0.05**	**0.81±0.05**	0.82±0.06
WLV-SHEF	0.84±0.05	0.84±0.05	**0.82±0.05***

Table 3: RMSE and 95% confidence intervals of the predictions for the test partitions. Best mean results are displayed boldface. Asterisks denote a significant difference in performance (paired re-sampling, 95% confidence) respect to *both* the other two methods.

ter result than PCA and our approach; for LORIA and TCD, no statistically significant difference was observed between our approach and Baseline but both systems obtained a statistically better result than PCA; for UPV and WLV-SHEF, our approach statistically outperformed the other two methods; and for SDLLW, UEDIN and UU, no significant differences were found.

These were quite surprising results. Given the encouraging RMSE improvements observed in cross-validation (see Table 2), we expected to obtain similar differences over Baseline in test. We followed a careful cross-validation training process (see Section 3.3) where each experiment was evaluated in a held-out test fold used neither to reduce the dimensionality nor to estimate the prediction model. Therefore, we hypothesized that the explanation for the results in Table 3 was that the training partitions were not representative of the test partitions. We evaluated this hypothesis by means of a series of multivariate Hotelling's two-sample T^2 tests [23]. The objective of these tests is to determine if two samples (in our case the values of the features in the training and test partitions) have been sampled from the same population or not. The results of the tests indicated that, for all feature sets, the training and test partitions were indeed statistically different ($p < 0.01$). In contrast, no statistical difference was found, for any of the feature sets, between the dev-train and dev-test folds used in the cross-validation training process.

In a more fine-grained analysis, we study individually the features in each set. The results of a series of Student's two-sample t-tests [21] indicated that most of the features did exhibit statistically different values ($p < 0.01$) between training and test. E.g., the value one of these "mismatched" features in the UPV set was $\mu = 1.7$ ($\sigma = 1.4$) in training, and $\mu = 0.9$ ($\sigma = 0.8$) in test. In contrast, only about 1% of the features exhibit different values between the cross-validation dev-train and dev-test folds. Table 4 displays, for each set, the percentage of "mismatched" features between the training and test partitions.

DCU-SYMC	45.1%	UEDIN	48.1%
LORIA	24.5%	UPV	67.4%
SDLLW	73.3%	UU	38.8%
TCD	30.2%	WLV-SHEF	28.6%

Table 4: Ratio of the features in each set that have significantly different values in the training and test partitions. These ratios reduce to about 1% in the dev-train and dev-test cross-validation folds. Significance computed by Student's two-sample t-test (99% confidence).

This mismatch can be partially explained by the fact that the training and test partitions contain news texts of different years [1], but we still consider that the main issue is the size (only 1832 samples) of training partitions that did not adequately represent test partitions. However, both our approach and the baseline systems had to deal with this mismatch, so, why our method and PCA seemed to be more heavily penalized than Baseline?

The projection of the features is computed based on the training data. Thus, if the training partition is not representative of the test partition, the reduced feature sets will be projected in a "direction" that may penalize the prediction accuracy for the test set. That is, crucial information to predict the quality scores of the test partition may be stripped out. This drawback is common to any dimensionality reduction technique as exemplified by the also poor test results (Table 3) obtained by PCA.

The conclusion that can be extracted from these results is that the use of feature reduction implies a greater risk of over-training the prediction system. This effect particularly important if training data is scarce but it is mitigated as more training data is available. Thus, given the encouraging cross-validation results in Table 2, better prediction accuracy could be expected in test whenever an adequate training partition is provided.

Under the assumption that the original features can be computed in advance, a complimentary advantage of the studied two-step QE approach is that it allows us to build more time-efficient QE systems. Figure 3 displays the time required to build an SVM model (including meta-parameter optimization) and obtain the test predictions as a function of the number of features

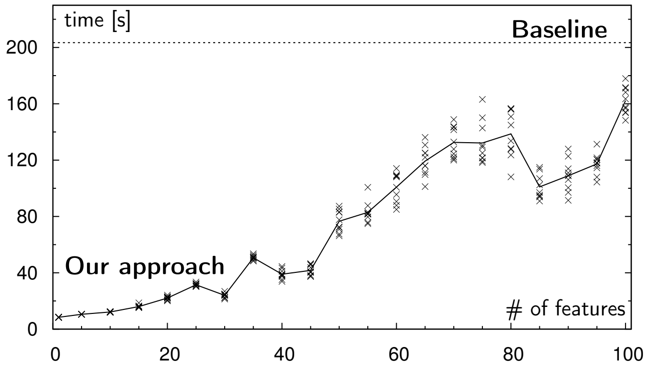

Figure 3: Operating time (training plus prediction) of the SVM model as a function of the number of features used to built the model. Baseline system was trained with the 147 original features of the WLV-SHEF set.

used to train the model. Specifically, we built QE systems with an increasing number of LVs extracted from the WLV-SHEF feature set. Each point in the figure is the average time of ten experiments. Results show how operating times increased with the number of LVs. For instance, the operating time of the baseline model trained with the original 147 features (0.84 RMSE) was $\sim$200 seconds, while the operating time of the system built with the 14 LVs extracted by PLS (0.82 RMSE) was only $\sim$15 seconds which represents one order of magnitude less operating time. Hence, our approach is well-suited to be applied to scenarios, such as interactive MT [3], with strict temporal restrictions.

4.2. Exploiting the scalability of our approach

Results in the previous section have shown that the studied QE approach was able to extract the relevant prediction information from different sets of noisy features. We now take a further step in this direction and present results where all the features used in the previous experiments are joined together to create an extremely high-dimensional feature set from which to predict quality scores. This aggregated set, denoted by ALL, contains 1197 features for each translation; approximately 55% of them being collinear with the rest.

Figure 4 shows cross-validation prediction accuracy (RMSE and 95% confidence interval) of the studied QE approach as a function of the number of LVs. Again, we also display results for a baseline SVM model built using all the features, and for a system built using PCA instead of PLS. Our approach obtained a score of 0.45 ± 0.01 RMSE with only 86 LVs. This result represents a 30% reduction relative to the baseline RMSE calculated with 1197 features. Regarding PCA, it barely reached Baseline performance. These results indicate that our approach was able to exploit the information contained in the ALL set to improve prediction accuracy. In contrast, both Baseline and PCA were unable to adequately manage the huge number of noisy and collinear features. Additionally, the operating time of the Baseline systems was $\sim$ 23 minutes, while it reduced to $\sim$2 minutes when we used the optimal 86 LVs.

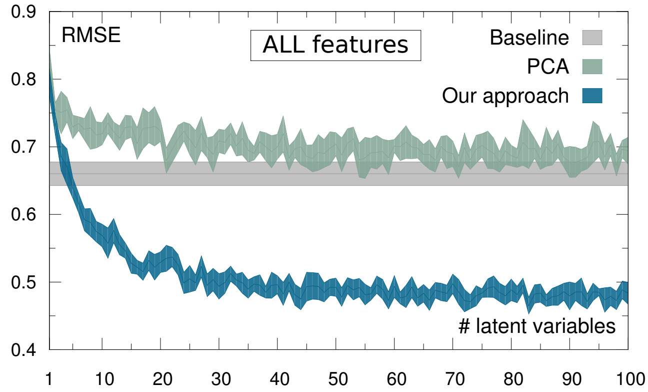

Figure 4: Cross-validation learning curve for the high-dimensional (1197 features) ALL set.

Test results were again quite disappointing: 1.4 ± 0.1 RMSE of our approach versus 0.78 ± 0.06 RMSE of Baseline and 0.81 ± 0.07 of PCA. We hypothesize that the clearly worse result of our approach in this case was due to the larger number features. As more features are available, our system can generate more "specialized" LVs. Given that the training data does not adequately represents the test data (see discussion in §4.1), this better projection (as shown in Figure 4) actually hinders prediction accuracy in the test set.

5. Summary

We have described an empirical study of a two-step QE approach specifically designed to manage the noisy features usually derived from natural language sentences. This approach, first described in [8] implements a method based on PLS to extract, from the original features, the LVs that actually govern translation quality, and an SVM model to actually predict the quality scores from these LVs.

Empirical cross-validation results showed that the studied QE approach was able to obtain very large feature reduction ratios, and at the same time, it usually outperformed systems built with all the original features and systems that use PCA instead of PLS to reduce the dimensionality. Unfortunately, results in the held-out test partitions were disappointing. The results of different statistical tests seem to indicate that this was due to the small size of the training partitions. Hence, larger RMSE improvements could be expected in test whenever a representative training partition is provided.

A complimentary advantage of the studied QE approach is its time-efficiency. This fact makes our approach well-suited to be deployed in scenarios with strict temporal restrictions, such as interactive MT systems. Alternatively, we could take advantage of this efficiency to predict translation quality from huge sets of features. Results in this direction show that our approach was able to efficiently manage more than a thousand features largely improving prediction accuracy.

6. Acknowledgments

Work supported by the European Union 7th Framework Program (FP7/2007-2013) under the CasMaCat project (grans agreement n^o 287576), by Spanish MICINN under grant TIN2012-31723, and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

7. References

[1] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proc. of the* 7th *Workshop on SMT*, 2012, pp. 10–51.

[2] L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *Proc. of the European Association for Machine Translation*, 2009, pp. 28–35.

[3] J. González-Rubio, D. Ortiz-Martínez, and F. Casacuberta, "Balancing user effort and translation error in interactive machine translation via confidence measures," in *Proc. of the Association for Computational Linguistics*, 2010, pp. 173–177.

[4] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *Proc. of the conference on Computational Linguistics*, 2004, pp. 315–321.

[5] R. Bellman, *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[6] D. Langlois, S. Raybaud, and K. Smaïli, "LORIA system for the WMT12 quality estimation shared task," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 114–119.

[7] R. Soricut, N. Bach, and Z. Wang, "The SDL Language Weaver systems in the WMT12 quality estimation shared task," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 145–151.

[8] J. González-Rubio, J. R. Navarro-Cerdán, and F. Casacuberta, "Dimensionality reduction methods for machine translation quality estimation," *Machine Translation*, pp. 1–21, 2013.

[9] H. Wold, *Estimation of Principal Components and Related Models by Iterative Least squares*. Academic Press, 1966, pp. 391–420.

[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[12] B. Mevik, R. Wehrens, and K. H. Liland, *PLS: Partial Least Squares and Principal Component regression*, 2011, R package version 2.3-0.

[13] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[14] E. W. Cheney and D. R. Kincaid, *Numerical Mathematics and Computing*. Brooks/Cole, 2012.

[15] R. Rubino, J. Foster, J. Wagner, J. Roturier, R. Samad Zadeh Kaljahi, and F. Hollowood, "Dcu-symantec submission for the WMT 2012 quality estimation task," in *Proc. of the* 7th *Workshop on SMT*, 2012, pp. 138–144.

[16] E. Moreau and C. Vogel, "Quality estimation: an experimental study using unsupervised similarity measures," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 120–126.

[17] C. Buck, "Black box features for the WMT 2012 quality estimation shared task," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 91–95.

[18] J. González-Rubio, A. Sanchís, and F. Casacuberta, "PRHLT submission to the WMT12 quality estimation task," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 104–108.

[19] C. Hardmeier, J. Nivre, and J. Tiedemann, "Tree kernels for machine translation quality estimation," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 109–113.

[20] M. Felice and L. Specia, "Linguistic features for quality estimation," in *Proceedings of the* 7th *Workshop on SMT*, 2012, pp. 96–103.

[21] W. Gosset, "The probable error of a mean," *Biometrika*, no. 1, pp. 1–25, 1908.

[22] Y. Zhang and S. Vogel, "Measuring confidence intervals for the machine translation evaluation metrics," in *Proc. of the Conference on Theoretical and Methodological Issues in Machine Translation*, 2004.

[23] T. Anderson, *An Introduction to Multivariate statistical Analysis*. New York: Wiley, 1958.