Mach Translat (2013) 27:171–192
DOI 10.1007/s10590-013-9138-4

Predicting sentence translation quality using extrinsic and language independent features

Ergun Biçici · Declan Groves · Josef van Genabith

Received: 6 October 2012 / Accepted: 20 April 2013 / Published online: 30 August 2013

Abstract We develop a top performing model for automatic, accurate, and language independent prediction of sentence-level statistical machine translation (SMT) quality with or without looking at the translation outputs. We derive various feature functions measuring the closeness of a given test sentence to the training data and the difficulty of translating the sentence. We describe `mono` feature functions that are based on statistics of only one side of the parallel training corpora and `duo` feature functions that incorporate statistics involving both source and target sides of the training data. Overall, we describe novel, language independent, and SMT system extrinsic features for predicting the SMT performance, which also rank high during feature ranking evaluations. We experiment with different learning settings, with or without looking at the translations, which help differentiate the contribution of different feature sets. We apply partial least squares and feature subset selection, both of which improve the results and we present ranking of the top features selected for each learning setting, providing an exhaustive analysis of the extrinsic features used. We show that by just looking at the test source sentences and not using the translation outputs at all, we can achieve better performance than a baseline system using SMT model dependent features that generated the translations. Furthermore, our prediction system is able to achieve the 2 nd best performance overall according to the official results of the quality estimation task (QET) challenge when also looking at the translation outputs.

E. Biçici (✉) · D. Groves · J. van Genabith
Centre for Next Generation Localisation, Dublin City University, Dublin, Ireland
e-mail: ebicici@computing.dcu.ie

D. Groves
e-mail: dgroves@computing.dcu.ie

J. van Genabith
e-mail: josef@computing.dcu.ie

Our representation and features achieve the top performance in QET among the models using the SVR learning model.

Keywords Statistical machine translation · Quality estimation · Machine learning · Performance prediction

1 Predicting sentence translation quality

We investigate the prediction of sentence translation quality for machine translation (MT) without training a statistical MT (SMT) model or using the reference for the translations, relying only on the extrinsic features derived from the training set and the test set. We use feature functions measuring the *closeness* of test sentences to the training data with the goal of predicting MT performance in terms of translation quality given samples from the test and training sets. The feature functions we develop are novel, language and SMT system independent, and allow quality estimation to be performed extrinsically given an SMT model. SMT system independence is important due to possible modeling or search errors that the SMT system might encounter during decoding (Koehn 2010).

SMT output is often post-edited by human translators to create a correct translation. Predicting sentence translation quality is important because the expected translation performance can help in estimating how much post-editing effort (PEE) we would spend for correcting the translations obtained for a given set of test sentences, which is valuable in commercial MT workflows. Furthermore, prediction of MT performance can also be useful for domain adaptation in MT. If we can estimate the achievable performance before training an SMT model, we may incorporate additional training data to increase the expected performance. Another application area is in the evaluation and selection of better SMT models among different models for translating a particular test set. Prediction of sentence-level MT performance can help develop automatic evaluation metrics, for instance by using various target side features and SMT translations from different systems (Albrecht and Hwa 2008; Specia et al. 2010).

Confidence estimation is the task of estimating a score of confidence in the output generated by a natural language processing system without a reference using extrinsic (black-box) or intrinsic (glass-box) features (Blatz et al. 2004; Specia et al. 2009). He et al. (2010) develop a translation recommender system aiming to recommend either an SMT output or a match from a given translation memory (TM) to the user for minimizing PEE. Birch et al. (2008) examine translation performance differences between different language pairs based on the amount of reordering, the vocabulary size, and the historical relatedness of the language pairs as measured by the percentage of cognates on a set of meanings. Their experiments used a fixed test set to be translated between 11 different languages trained over the same Europarl corpus, the 3 variables as the input, and the BLEU (Papineni et al. 2002) scores as the output. In this work, we focus on the quality estimation task (QET) (Callison-Burch et al. 2012), which aims to develop quality indicators for translations at the sentence-level and predictors without access to the reference.

1.1 The extrinsic translation quality prediction problem

In this section, we define the extrinsic translation quality prediction problem where our goal is to estimate the sentence-level MT performance given a training set, a source test sentence, and its translation by the MT system, without relying on any information intrinsic to the SMT model used, which may be available only after the model has been trained. Let $\mathcal{M}$ represent an SMT model, $\mathcal{D} = (\mathcal{D}_S, \mathcal{D}_T)$ some parallel training data with $\mathcal{D}_S$ the source side and $\mathcal{D}_T$ the target side, and (S, T) a pair of sentences from a test set, $\mathcal{T}$, where S is the source sentence and T is its reference translation. We use $\mathcal{M}_\mathcal{D}$ to refer to the SMT model $\mathcal{M}$ trained on $\mathcal{D}$ and $T' = \mathcal{M}_\mathcal{D}(S)$ to refer to the translation obtained by $\mathcal{M}_\mathcal{D}$ on S.

The *extrinsic translation quality prediction problem* involves finding a function f approximating the target performance on the test sentence and its translation before training:

$$f(\mathcal{M}, \mathcal{D}, S, T') \approx f_T(\mathcal{M}_\mathcal{D}(S), T). \tag{1}$$

We approach f as a supervised learning problem and learn from training data representing the sentence-level translation performance. We describe our approach to modeling the learning problem in Sect. 3. In this paper, we try to estimate the PEE score required to correct the translations obtained by an SMT system.

The *blind translation quality prediction problem* involves finding a function f approximating the target performance on the test sentence without looking at its translation before training:

$$f(\mathcal{M}, \mathcal{D}, S) \approx f_T(\mathcal{M}_\mathcal{D}(S), T). \tag{2}$$

As we show in Sect. 5, our blind prediction model is able to achieve similar or better performance than the QET baseline system, which is not blind and intrinsic as it uses the translations and exploits features dependent on the SMT model used.

1.2 The machine translation performance prediction (MTPP) model

Unlike the traditional quality estimation in MT (Callison-Burch et al. 2012) which looks both at the source sentence and its translation in the target language, we show that using only the source sentences themselves can be enough to achieve better predictions with our MT performance prediction (MTPP) model than a competitive QET baseline system. SMT system performance is affected by the amount of training data used as well as the *closeness* of the test set to the training set. We measure how well the test set matches the training set using several features and metrics. One way to measure the similarity of test sentences to the training sentences is by looking at how close can we find grammatical structures or vocabulary in the training set. As an example, we obtain unsupervised Common Cover Link parses (Seginer 2007) for the sentences found in the test set and the training set and we measure the coverage of the syntactic structures observed in a test sentence, the features being the links found. Therefore, we can

use *grammar coverage*, which represents the percentage of the test sentence syntactic structures found in the training sentences. We define various novel, extrinsic, and language independent feature functions for measuring the closeness of a test sentence to a given training set in Sect. 2. Since we only rely on extrinsic features, we can train MTPP faster and we do not need to wait for training an SMT model.

Our MTPP model works without training an MT model, without any MT model dependent information, and can predict without even looking at the translations generated by the MT model after decoding. We show that by just looking at the test source sentences and without using their SMT translation outputs, we can achieve better performance than the QET baseline system, which uses SMT model dependent features that generate the translations. Our prediction system is able to achieve the 2nd best performance according to the official results of the QET challenge when also looking at the translation outputs.

We organize our work as follows. In Sect. 2, we define the feature functions we use to measure the closeness of test sentences to the training corpora. In Sect. 3, we present the translation quality estimation task, the baseline system used, the learning settings, and the performance evaluation metrics we use. In Sect. 4, we discuss the learning models we use for predicting the translation quality, feature subset selection and partial least squares (PLS) algorithms that are employed, and how we optimize the parameters of the learning models and settings. In Sect. 5, we present our results on the QET challenge comparing the performance in different learning settings, the performance of learning algorithms, and their performance after dimensionality reduction with PLS or feature selection. We also give a detailed ranking of the features selected in each learning setting. In the last section, we summarize our contributions and present out conclusions.

2 Feature functions for modeling $f(\mathcal{M}, \mathcal{D}, S, T')$ and $f(\mathcal{M}, \mathcal{D}, S)$

In this section we discuss the feature functions we use for modeling $f(\mathcal{M}, \mathcal{D}, S, T')$ and $f(\mathcal{M}, \mathcal{D}, S)$ and how we model the prediction problem. We categorize the feature functions used as either using a single side of the parallel corpus (i.e. the distribution of test features in the source language training corpus) or using both sides (i.e. features based on co-occurrence). We refer to these feature functions as `mono` or `duo`, respectively. Each feature function measures the closeness of the training data to the test sentences and is an indicator that can be used for predicting machine translation quality. Our set of feature functions are novel, extrinsic to any SMT model, and language independent. Our features introduce novel representations of distributional similarity, close retrieval, and translational equivalence for predicting the performance. We provide the number of features each feature function contributes in curly brackets after their name.

2.1 Features

We use n-gram features (up to 3-g) defined over text or *link structures* as the basic units of information over which similarity calculations are made. We perform unsupervised

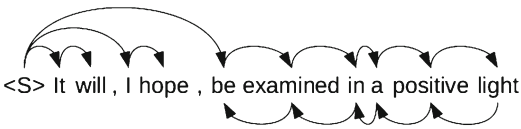

Fig. 1 Example CCL output with *arrows* representing the links and <S> representing the start of the sentence

parsing using the Common Cover Link (CCL) algorithm (Seginer 2007) [1] and extract links from base words to head words. CCL structures allow us to obtain structures representing the grammatical information instantiated in the training and test data. The CCL parser is trained using the source side of the training and test sets. The main differences between dependency structures and common cover (CC) link sets or CCL structures can be summarized as follows (Seginer 2007): (i) CCL allows exocentric constructions, constructions where multiple elements are combined with equal status, without one dominating the other; (ii) each CC link has a depth, which represents the depth of the source node of the link; (iii) long distance CC links require links to every element in between. Shortest CC links use the shortest possible CC links to represent the sentence structure. In exocentric constructions, back and forth links exist between every two adjacent words. CCL allows equivalent classes by double links between words whereas traditional dependency representation restricts words to be either a head or a modifier. For instance, a noun phrase such as [DT N], where DT is a determiner and N is a noun, may be represented as [DT $\rightleftarrows$ N], where CCL gives both the determiner and the noun an equal status. Figure 1 depicts the parsing output obtained by CCL for an example sentence. Double links in the output increase the recall and help us find relevant sentences from the training set more easily.

2.2 mono feature functions

mono feature functions use statistics involving only one side of the parallel corpus when determining the closeness of the test sentences to the training set.

– *Feature coverage {45}*: We define the coverage of features as the percentage of test features found in the training set. The features can be defined over the n-grams or the CCLs observed. Let $\mathcal{F}(.)$ be a function returning the features found in its argument, then $\mathcal{F}(\mathcal{D}_S)$ represents the features found in the source side of the parallel training sentences $\mathcal{D}$. Given a source test sentence, S, we define the following measures:

$$\mathrm{P} = \frac{|\mathcal{F}(\mathcal{D}_S) \cap \mathcal{F}(S)|}{|\mathcal{F}(\mathcal{D}_S)|}, \quad \mathrm{R} = \frac{|\mathcal{F}(\mathcal{D}_S) \cap \mathcal{F}(S)|}{|\mathcal{F}(S)|}, \quad F_1 = \frac{2\,\mathrm{P}\,\mathrm{R}}{\mathrm{P} + \mathrm{R}},$$
$$\mathrm{R}^w = \sum_{f \in \mathcal{F}(\mathcal{D}_S) \cap \mathcal{F}(S)} w_f \tag{3}$$

where P is the *precision*, which measures the percentage of accurate feature predictions, and R is the *recall*, which measures the percentage of features found.

[1] Version 1.0.0.2, available from http://www.seggu.net/ccl/.

R corresponds to the source coverage of the test features in the parallel training sentences. P and R are important since most automatic evaluation metrics depend on them (i.e. BLEU is precision based). We obtain the weighted feature coverage measure $\mathtt{R}^w$ with $w_f = C(f, \mathcal{F}(S))/C(\mathcal{F}(S))$ being the weight for feature f, $C(f, \mathcal{F}(S))$ returning the number of times feature f appears in the set of features of the test sentence, and $C(\mathcal{F}(S))$ being the total number of features in S. w_f represents the probability of observing feature f and $\sum_{f \in \mathcal{F}(S)} w_f = 1$. We can also obtain similar features on the target side over $\mathcal{D}_T$ and the translation outputs T'. {45: P, R, F_1, $\mathtt{R}^w$, F_1^w, number of features, geometric mean (GM) instead of F_1, $\mathtt{GM}^w$, number of OOV words for 1, 2, 3, 1&2, 1&2&3-g}

– *Relative coverage {2}*: We evaluate how well the features of the test sentences are covered by the training set relative to the lengths of sentences they appear in. We obtain relative coverage for $S \in \mathcal{T}$ as follows:

$$\phi_C(S) = \frac{1}{|\mathcal{F}(S)|} \sum_{f \in \mathcal{F}(S)} \frac{1}{z} \sum_{S' \in Z(f)} \frac{1}{|S'|} \tag{4}$$

where $Z(f) = \{S' : f \in \mathcal{F}(S'), \forall_{S' \in \mathcal{D}}\}$, $z = |Z(f)|$ is the number of S' found. We also use $\phi_C(S)$ with $z = 1$.

– *Synthetic translation performance {6}*: We estimate a bound on the BLEU translation score achievable with synthetic translations generated conditioned on the n-gram coverage (i.e. 2gram BLEU bnd). For instance, for 1-g coverage, we can randomly select a number of tokens that would correspond to the coverage level and replace them with a token not found in the translation, such as <UNK>. We measure the performance over these random translation instances that are generated {3 for BLEU, 3 for F_1}.

– *Feature vector similarity {8}*: We obtain vectors of feature counts where the features can be n-grams or the CCLs obtained from the test sentences, which determines the vector length. We use the correlation coefficient or χ^2 statistic as the similarity over the feature vectors v^1 and v^2: $\chi^2(v^1, v^2) = \sum_{i=1}^{n} (v_i^1 - v_i^2)^2 / v_i^2$. Here v^1 corresponds to the observed vector of features (i.e. feature vector of the training set) and v^2 is the expected vector of features (i.e. feature vector of the test sentences). We also calculate over vectors normalized by their sum {2 for each n-gram}.

– *Perplexity {15}*: Perplexity is defined as $2^{H(p)}$ where $H(p)$ is the entropy of the distribution p and it is used for measuring how well a given language model (LM) predicts tokens in the test corpus. We also use the bits per word, log probability, and average perplexity (i.e. 1 gram log$p$) {3 for each LM order}.

– *Entropy {2}*: *Entropy* (Eq. 5) quantifies the amount of uncertainty in the random variable $\mathbf{x}$ and *conditional entropy* quantifies the amount of uncertainty in $\mathbf{y}$ given $\mathbf{x}$ (Eq. 6) (Cover and Thomas 2006). Let $\mathbf{x}$ and $\mathbf{y}$ be two discrete random variables (e.g. features) and p, q represent two distributions (e.g. training set and test sentences), then we define measures for comparing random variables (Eqs. 8, 9). We use the average of *mutual information*, $I(\mathcal{F}(S); \mathcal{F}(S'))$ for $S \in \mathcal{T}$ and $S' \in \mathcal{D}$, over the top N training instances retrieved for each S (Eq. 10), where $\mathsf{R}_{\phi_{\mathfrak{D}}}(S, \mathcal{D}, N)$

retrieves the top N training instances using $\phi_{\mathfrak{D}}$, which is defined next. We define average *information distance*, $\phi_{\mathrm{D}}(\mathcal{T}, \mathcal{D}, N)$, similarly.

$$H(\mathbf{x}) = -\sum_{x \in \mathbf{x}} p(x) \log p(x) \tag{5}$$

$$H(\mathbf{y}|\mathbf{x}) = \sum_{x \in \mathbf{x}} \sum_{y \in \mathbf{y}} p(x, y) \log \left(\frac{p(x)}{p(x, y)} \right) \tag{6}$$

$$H(\mathbf{x}, \mathbf{y}) = H(\mathbf{x}) + H(\mathbf{y}|\mathbf{x}) \tag{7}$$

$$I(\mathbf{x}; \mathbf{y}) = \sum_{x \in \mathbf{x}} \sum_{y \in \mathbf{y}} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \tag{8}$$

$$D(\mathbf{x}; \mathbf{y}) = \frac{H(\mathbf{x}, \mathbf{y}) - I(\mathbf{x}; \mathbf{y})}{H(\mathbf{x}, \mathbf{y})} \tag{9}$$

$$\phi_{\mathrm{I}}(\mathcal{T}, \mathcal{D}, N) = \frac{1}{N|\mathcal{T}|} \sum_{S \in \mathcal{T}} \sum_{S' \in \mathsf{R}_{\phi_{\mathfrak{D}}}(S, \mathcal{D}, N)} I(\mathcal{F}(S); \mathcal{F}(S')). \tag{10}$$

2.3 `duo` feature functions

`duo` feature functions use statistics involving both sides of a parallel corpus to calculate the similarity.

- $\phi_{\mathfrak{D}}$ {12}: We define Dice's coefficient score as $\mathfrak{D}(x, y) = \frac{2C(x,y)}{C(x)C(y)}$, where $C(x, y)$ is the number of times features x and y co-occur and $C(x)$ is the number of times x appears in the training set. We use $\phi_{\mathfrak{D}}$, *dice* training instance selection method scorer, to estimate the goodness of a training sentence pair, $(S^\star, T^\star)$, by the sum of the alignment scores and to select training sentences given a test source sentence, S (Bicici and Yuret 2011):

$$\phi_{\mathfrak{D}}(S, S^\star, T^\star) = \frac{1}{z} \sum_{x \in \mathcal{F}(S)} \sum_{j=1}^{|T^\star|} \sum_{y \in Y(x)} \mathfrak{D}(y, t_j), \tag{11}$$

  where z is a scaling factor proportional to $|S|(|T^\star| \log |S^\star| + |S^\star| \log |T^\star|)$, $Y(x)$ lists the tokens in feature x, and $t_j \in T^\star = \{t_1, \ldots, t_m\}$ is a token.
- *Diversity {3}*: Diversity of the features provided in a training set increases the recall on the target side, which can reduce the out-of-vocabulary (OOV) tokens and improve the translation performance. We derive the diversity score using the co-occurrence tables for the features observed in the paired set of training sentences. We obtain the diversity score as follows:

$$\phi_V(S) = \frac{1}{|\mathcal{F}(S)|} \sum_{f \in \mathcal{F}(S)} C(f, *). \tag{12}$$

– *IBM1 translation probability {12}*: We estimate the translation probability of test sentences with IBM translation model 1 (IBM1) (Brown et al. 1993) using the co-occurrence of words:

$$P(T|S) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} \mathrm{tr}(t_j|s_i), \tag{13}$$

where S is a source sentence, T is its translation, $\mathrm{tr}(t_j|s_i)$ is the translation probability of the word $t_j \in T = \{t_1, \ldots, t_m\}$ given $s_i \in S = \{s_1, \ldots, s_l\}$, and ϵ is some small fixed number. Moore (2002) uses $P(S, T)$ instead of $P(T|S)$, due to the noisy channel model (Knight 1999), which assumes that S is hypothetically corrupted by some "noise" and turned into T. Following Moore, we estimate m according to a Poisson distribution with estimated target sentence length as the mean.[2] We use the following Bayes translation probability, which uses the most probable t_j for each s_i:

$$P(T|S)_{\mathfrak{B}} = \frac{\epsilon}{(l+1)^{\hat{m}}} \prod_{j=1}^{\hat{m}} \sum_{i=0}^{l} \max_{t_j \in c(s_i,*)} \mathrm{tr}(t_j|s_i), \tag{14}$$

where $\hat{m}$ is the estimated target sentence length and $c(s_i, *)$ returns all the possible target features co-occurring with s_i.

$$P(T|S)_{\mathfrak{B}_k} = \frac{\epsilon}{(l+1)^{\hat{m}}} \prod_{j=1}^{\hat{m}} \sum_{i=0}^{l} \sum_{t_j \in topk(c(s_i,*))} \mathrm{tr}(t_j|s_i), \tag{15}$$

$P(T|S)_{\mathfrak{B}_k}$ uses the probability of the top k translations for s_i. We also obtain the joint probability using $\mathrm{tr}(t_j, s_i)$.

– *Minimum Bayes retrieval risk (MBRR) {4}*: We define MBRR translation as the Minimum Bayes Risk (MBR) translation over a set of top N retrieved training instances for a given source sentence. MBR uses a loss function to account for the contributions of different translations selected from a set of highly probable translations (Koehn 2010). We define MBRR translation as follows:

$$T_{\mathtt{MBRR}} = \arg\min_{T^*} \sum_{(S^\star, T^\star) \in \mathsf{R}_{\phi_{\mathfrak{D}}}(S, \mathcal{D}, N)} L(S, S^\star)\, P(T^\star|S). \tag{16}$$

The loss function is defined using the normalized retrieval score over the top N retrieved sentences: $L(S, S^\star) = (1 - \underline{\phi}_{\mathfrak{D}}(S, S^\star))$. We obtain the log probability of $T_{\mathtt{MBRR}}$ using Eq. 13. MBRR gives us a score about how good we can translate the source sentence whereas $\phi_{\mathfrak{D}}$ gives a score about how close we can find matching items from the training set.

[2] $P(m|l) = \frac{e^{-lr}(lr)^m}{m!}$ where r represents the ratio of the number of target words to the number of source words found in the training set.

3 Translation quality estimation

Translation quality estimation involves predicting sentence-level translation performance given the source sentence and its translation. The quality estimation task (QET) (Callison-Burch et al. 2012) aims to develop quality indicators and predictors for translations at the sentence-level without access to the reference. QET provides access to a baseline prediction system using features derived from a given training set, a baseline SMT system built using the training set, a test set, and evaluation metrics. The translation task involves translating news text in the English–Spanish language pair using an SMT system built with Moses (Koehn et al. 2007) with training corpus provided by WMT'12 (Callison-Burch et al. 2012). The training corpus contains about 1.714 million sentences and is composed of Europarl and News Commentary corpora. The training and test data for the prediction task contains source sentences, their SMT translation outputs, and the corresponding post-editing effort (PEE) score manually annotated by humans. The sentence-level scores used in the QET are the average of three measures of PEE scores in the range [1, 5] where the higher scores correspond to better translations. The reference translations and the post-edited translations are also provided for completeness but they are not used when training the prediction models. The number of sentences in the training set and the test set are 1, 832 and 422, respectively.

The 17 baseline features the QET baseline system provides are numbered as fn for $1 \leq n \leq 17$ and they contain the following information. Let S be a source sentence and T' its translation, then: $f1$: $|S|$ where $|.|$ returns the length of its argument, $f2$: $|T'|$, $f3$: average source token length, $f4$: LM probability of S, $f5$: LM probability of T', $f6$: average number of times words occur within the target sentence, $f7$: average number of translations per source word as found in the GIZA++ (Och and Ney 2003) phrase table used by the SMT system generating the translations whose probability is above 0.2, ($p(t|s) > 0.2$), $f8$: similar to $f7$ but with $p(t|s) > 0.01$ and weighted by the inverse frequency of source words, $f9$: percentage of 1-g in quartile 1 or low frequency words in the source corpus, $f10$: percentage of 1-g in quartile 4 or high frequency words in the source corpus, $f11$: same as $f9$ but with 2-g, $f12$: same as $f10$ but with 2-g, $f13$: same as $f9$ but with 3-g, $f14$: same as $f10$ but with 3-g, $f15$: percentage of 1-g in S seen in source corpus, $f16$: number of punctuation symbols in S, $f17$: number of punctuation symbols in T'. The baseline setting is a highly competitive baseline, as it contains features based on the GIZA++ alignments, which are used for generating the phrase tables used by the SMT system itself that generated the translations. In fact, only 3 participating systems in the quality estimation shared task managed to beat the baseline.

3.1 Learning settings

We obtain results for the QET dataset as follows. Using our feature functions, we extract the features both on the training set of 1, 832 sentences and the test set of 422 sentences, which consist of pairs of a source sentence and its translation output. The corresponding PEE scores for each of these datasets are given and are added to the

datasets as the target to predict. For our LM, we do not use the LM resources provided by QET, instead we only use the training corpus provided to build a 5-g target LM. At testing time, the predictions are bound so as to have scores in the range [1, 5]. We standardize the features in both the training and test data before learning and prediction (e.g. by centering and scaling to have zero mean and unit variance: $X = (X - \bar{X})/\sigma$). We perform experiments in 4 separate learning settings that use different feature sets. The following settings are used:

- `S`: In this setting, we only use the source sentences in the test sets to derive both the `mono` and `duo` features and we do not use the translation outputs to predict the translation quality. We obtain 133 features representing the relationship between the test set and the training set. `S` solves the blind translation quality prediction problem.
- `SB`: We add the 17 QET baseline features in addition to the source side only features we obtain for `S`, which totals the number of features to 150.
- `ST`: We use the source test sentences and their translations to derive the features. We obtain 236 features overall in this setting.
- `STB`: We add the 17 QET baseline features in addition to the 236 features in the `ST` setting to obtain 253 features.

The features in setting `S` are composed of 102 `mono` features out of which 45, 2, 6, 8, 15, 24, and 2 are based on coverage, relative coverage score, BLEU and F_1 bounds, feature vector similarity, LM, CCL structures, and mutual information and information distance, respectively and 31 `duo` features out of which 12, 3, 12, and 4 are based on $\phi_{\mathfrak{D}}$, feature diversity, IBM1 translation probability, and MBRR, respectively.

In setting `ST`, in addition to the 133 features from setting `S`, we obtain features that not only look at the conditional and the joint relationship between the source sentence S and its translation, T', but also additional `mono` and `duo` features using T'. Thus we obtain 236 features in setting `ST` where the additional 103 features are distributed as follows: 45, 15, 24, 8, and 2 are `mono` and are based on coverage, LM, CCL structures, feature vector similarity, and $\phi_{\mathtt{I}}$ and $\phi_{\mathtt{D}}$, respectively, and 3, 2, and 4, are `duo` and are based on feature diversity, IBM1 translation probability, and $\phi_{\mathfrak{D}}$, respectively.

3.2 Performance evaluation

This section describes the metrics we use to evaluate the performance by looking at either the sentence-level translation quality predictions or by looking at the rankings of the quality of the translations for the test set sentences. Let y_i represent the actual target value for instance i, $\bar{y}$ the mean of the actual target values, $\hat{y}_i$ the value estimated by the learning model, and $\bar{\hat{y}}$ the mean of the estimated target values, then we use the following metrics to evaluate the learning models:

- *Mean absolute error*:

$$|\bar{\epsilon}| = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

– *Relative absolute error*:

$$\overrightarrow{|\epsilon|} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |\bar{y} - y_i|}$$

– *Root mean squared error*:

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

– *DeltaAvg*:

$$\bar{\Delta}(V, S) = \frac{1}{|S|/2 - 1} \sum_{n=2}^{|S|/2} \left(\sum_{k=1}^{n-1} \frac{\sum_{s \in \bigcup_{i=1}^{k} q_i} V(s)}{|\bigcup_{i=1}^{k} q_i|} \right)$$

– *Spearman's correlation*:

$$r = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

We try to minimize the errors and maximize the DeltaAvg ($\bar{\Delta}$) score and the correlation of the predictions with the actual results. QET uses $\bar{\Delta}$, r_{rank}, $\bar{|\epsilon|}$, and RMSE as the official evaluation metrics. r_{rank} is defined as the correlation over the sentence quality prediction rankings. QET accepts either the estimated scores for the translations or a ranking of the translations without ties from 1 to N where $N = 422$ for the test set. Scores are used to obtain $\bar{|\epsilon|}$, $\overrightarrow{|\epsilon|}$, and RMSE and the rankings are used to measure Spearman's correlation score (r_{rank}) and the DeltaAvg ($\bar{\Delta}$) score (Callison-Burch et al. 2012). DeltaAvg calculates the average quality difference between the scores for the top $n - 1$ quartiles and the overall quality for the test set. In the DeltaAvg equation, V is a function returning the value of its argument sentence, S is the test set, and q_i corresponds to the ith top quartile. DeltaAvg is an interpretable score where a DeltaAvg of 0.5 corresponds to a 0.5 difference between the average obtained over the top quartiles and the overall average (Callison-Burch et al. 2012). In Spearman's correlation, ranks of the test instances are used to calculate the scores whereas in DeltaAvg, V returns the PEE score. Relative absolute error measures the error relative to the error when predicting the actual mean. We use the coefficient of determination, R^2, during parameter optimization where the models are regression based and to select optimal parameter settings. R^2 is defined as: $R^2 = 1 - \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{n} (\bar{y} - y_i)^2$, the higher the value the better.

4 Learning models

In this section, we discuss the learning models we use for predicting the translation quality for each given sentence. We use ridge regression (RR) and support vector regression (SVR) with RBF (radial basis functions) kernel (Smola and Schölkopf 2004). Both of these models learn a regression function using the features to estimate a numerical target value such as the PEE score or the BLEU score.

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)^T \in \mathbb{R}^{m \times p}$ represent the variables for m training instances and p variables and $\mathbf{y} = (y_1, \ldots, y_m)^T$ represent the output variable, then the cost function that RR is minimizing can be written as below:

$$\mathcal{J}(\mathbf{w}) = \| \mathbf{y} - \mathbf{X}\mathbf{w} \|_2^2 + \lambda \| \mathbf{w} \|_2^2, \tag{17}$$

where $\mathbf{w} \in \mathbb{R}^{p \times 1}$ is a weight vector we are trying to learn, $\| \cdot \|_2$ stands for the L_2 norm, and $\lambda \geq 0$ is a parameter for regularization, making sure that the learned weights do not have large values. $\epsilon = \mathbf{y} - \mathbf{X}\mathbf{w}$ is the training error.

The cost function for SVR is given below (Smola and Schölkopf 2004):

$$\begin{aligned} \mathcal{J}(\mathbf{w}) = \frac{1}{2} \| \mathbf{w} \|_2^2 + C \sum_{i=1}^{m} (\xi_i^+ + \xi_i^-) \\ s.t. \quad \mathbf{y} - \mathbf{X}\mathbf{w} \leq \epsilon + \xi_i^+ \\ \mathbf{X}\mathbf{w} - \mathbf{y} \leq \epsilon + \xi_i^- \\ \xi_i^+, \xi^- \geq 0 \end{aligned} \tag{18}$$

The parameter C determines the complexity where a large C fits more complex functions as it gives more weight for minimizing the error. ϵ defines a ϵ-insensitive tube out of which ξ_i^+ and ξ_i^- obtain non-zero values for points above and below, respectively (Drucker et al. 1996). ϵ helps find a $\mathbf{w}$, which represents the decision boundary even in cases where the training data is not separable. The RBF kernel is defined as follows: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \| \mathbf{x}_i - \mathbf{x}_j \|_2^2)$, where $\gamma > 0$. The parameters C, ϵ, and γ determine the behavior of SVR.

We also use these learning models after a feature subset selection (FS) step with recursive feature elimination (RFE) (described further in Sect. 4.1) or a dimensionality reduction and mapping step using partial least squares (PLS) (described further in Sect. 4.2). In the following subsections, we discuss the RFE approach to feature subset selection, dimensionality reduction with PLS, and parameter optimization for the learning models we use.

4.1 Feature subset selection

Guyon and Elisseeff (2003) present examples demonstrating that (1) independently and identically distributed variables are not truly redundant and therefore addition of

redundant attributes can lead to better class separation; (2) highly correlated variables can still be separated; (3) a variable that is not useful by itself can be useful together with other variables or two variables that are not useful by themselves can be useful together. These examples show that selecting subsets of features rather than ranking them according to their individual predictive power is a better alternative for feature selection. As the search methodology for the subset, they suggest using backward selection since forward selection does not evaluate the importance of the variables with respect to the ones that are not included.

Recursive feature elimination (RFE) (Guyon et al. 2002) is an iterative technique for backward feature subset selection that involves iteratively: (i) training a classifier, (ii) computing a ranking of the features according to the classifier, and (iii) removing the features with the lowest ranks. As we eliminate features, we obtain a ranking among them.

4.2 Dimensionality reduction with partial least squares (PLS)

The variance of a feature in the presence of changes in the target reflects its importance. It is important to scale the inputs so that their variance is correlated with their importance and when no prior information exists, standardizing the input features to have mean 0 and variance 1 (Sarle 2002). Partial least squares (PLS) maps the source features to latent features but it uses both the source and the target features. PLS is similar to Gram Schmidt orthogonalization in the sense that orthogonal latent features are obtained and it can be useful when the source features are correlated (Specia et al. 2009). Latent features are generated by an iterative process of orthogonalization or deflation (projection onto a space orthogonal to the latent feature vector) and vector selection.

The nonlinear iterative partial least squares (NIPALS) algorithm (Rosipal and Trejo 2001) is an iterative algorithm for finding the PLS solution as given in 1. Let $\mathbf{X} \in \mathbb{R}^{m \times p}$ represent the standardized input matrix and $\mathbf{y} \in \mathbb{R}^{m \times 1}$ be the output vector, then the PLS algorithm constructs basis directions that have high variance and at the same time have high correlation with the response (Hastie et al. 2009). The input variables $\mathbf{x}_j$ are weighted with $w_{i,j}$ by their effect on $\mathbf{y}$ where $\mathbf{w}_{i,j} = \mathbf{y}^T \mathbf{x}_j$. PLS outputs a set of orthogonal directions $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_M]$ for $M \leq p$. With few training instances, we can obtain a fat input matrix X when $p > m$, which makes the features used non-orthogonal with each other. In such scenarios, dimensionality reduction with PLS or feature selection can help improve the learning performance.

4.3 Parameter optimization

The learning parameters that govern the behavior of RR and SVR are the regularization λ for RR and the C, ϵ, and γ parameters for SVR. We perform iterative optimization of the learning parameters, the number of features to select, and the number of dimensions used for PLS. For all of the learning settings, we performed tuning on a subset of the training data. We optimize against the performance evaluated with R^2. Optimizing against Δ does not improve the performance since the parameters found will be

Algorithm 1: Partial least aquares algorithm

Input: Input variables $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{m \times p}$, output $\mathbf{y} \in \mathbb{R}^{m \times 1}$, number of dimensions to select M.
Data: Weights for PLS directions, $W = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$, $X_0 = X$.
Output: Orthogonal PLS directions, $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_M]$ for $M \leq p$.

$i \leftarrow 1$
while $i \leq M$ **do**
  $\mathbf{w}_i = \mathbf{y}^T X_i$
  $\mathbf{z}_i = X_i \mathbf{w}_i^T$
  $X_i = X_{i-1} - \mathbf{z}_i \mathbf{z}_i^T X_{i-1}$ (orthogonalization)
  $i \leftarrow i + 1$

optimizing a ranking of the data, which may not be meaningful. Optimizing for MAE also does not improve the performance since it is not regularized by the prediction error obtained when predicting the mean.

We start with an initial parameter grid and we gradually refine the searched best parameters in the later steps of the optimization. Each searched setting performance is 5-fold cross validated on the development set and their average R^2 is used as the score for the setting. For learning the optimal number of dimensions to map to with PLS, we first select the number of dimensions from a set of possibilities to try (some number less than the number of available features) and search for the optimal parameters in this lower dimensional space. For selecting the optimal feature subset, we perform RFE, eliminating only one feature at each step and search for the optimal parameters in this smaller feature set. We do not apply PLS and RFE together to discriminate their individual effects better. Once we obtain a set of good parameters at a previous optimization step, we use those parameters to refine and search for parameter settings that are close but better.

We plot the parameter optimization contour plot for setting `ST` with or without PLS and with or without FS in Fig. 2 where the performance is evaluated with R^2 for results with $R^2 \geq 0$. In this figure, we look at the changes in the performance with respect to changes in two parameters at a time, selecting the maximum score achievable overall. We explore regions with higher performance in finer detail. We observe that there is no single global optimum parameter setting for SVR. The best values for C, ϵ, and γ vary around 10, 0.5, and 10^{-3}, respectively. Optimizing the λ parameter is much easier for RR as there is only one parameter that governs the behavior. The best parameter settings we arrive at are given in Table 1.

5 QET results

In this section, we present the results we obtain on the QET. Table 2 presents the QET challenge results where the first row presents the baseline performance, which is the official result obtained with the baseline features using SVR with the RBF kernel and the following two rows list the performance of the best QET submission (Soricut et al. 2012), which uses 42 features including the baseline features and M5P or SVR as the learning model.

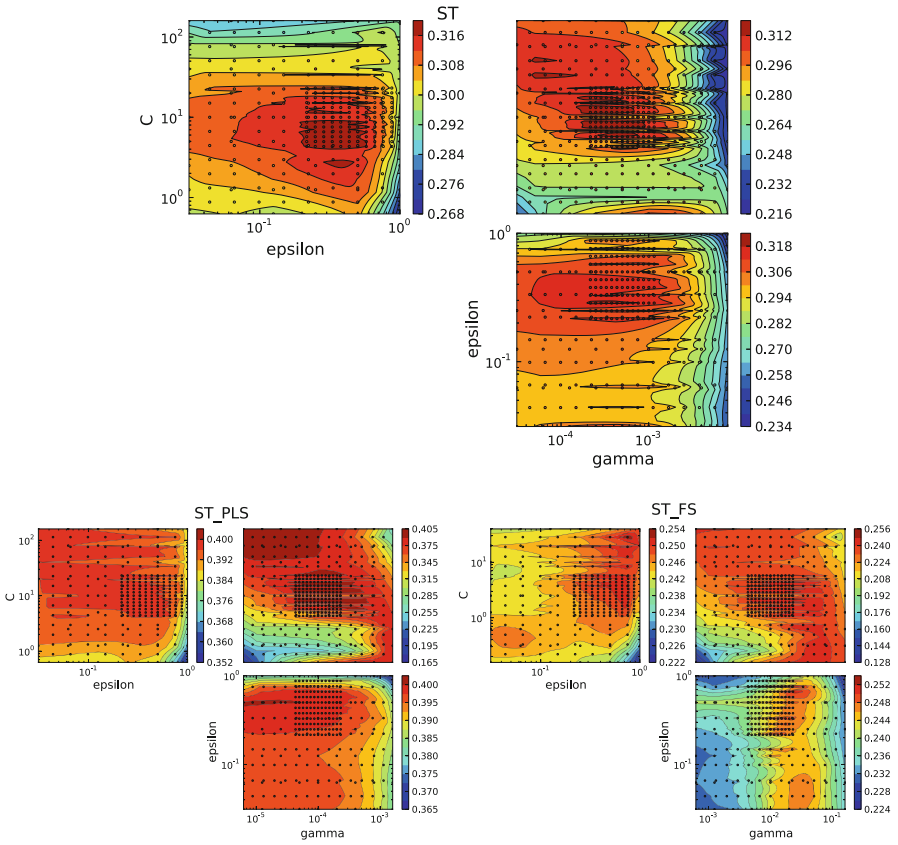

Fig. 2 SVR parameter optimization contour plot for learning setting ST without feature selection or PLS (*top*), with PLS (*bottom, left*), and with feature selection (*bottom, right*) with the C, γ, and ϵ parameters where the heat map shows the R^2 score and warmer colors show better performance. *Upper right corner* of each plot is C versus γ

Table 3 presents the results we obtain without PLS or FS. The first two rows list the results with setting S. We observe that we can obtain better performance than the baseline QET system in terms of both $|\bar{\epsilon}|$ and $|\overrightarrow{\epsilon}|$, which shows that by just looking at the test source sentences alone and not using their translation outputs, we are able to obtain as precise results as the baseline or better. The addition of baseline features in the SB setting improves the performance in all metrics except $|\bar{\epsilon}|$. The ST setting achieves significant improvements in the performance in all evaluation metrics and setting STB achieves a small increase in $\bar{\Delta}$, r_{rank}, and decrease in RMSE. According to the official results (Callison-Burch et al. 2012), ST and STB score 3rd from the top based on $\bar{\Delta}$ (2nd best submission and feature set) and 2nd based on r_{rank}. We obtain statistically significant improvements over the baseline system where p values $\lesssim 0.05$ are super-scripted in the results tables using paired t testing (Wasserman 2004) over absolute errors.[3]

[3] Overview of statistical testing, especially for machine translation (Bicici 2011, App. B).

Table 1 Best parameter settings found by the optimization process

Setting	RR		SVR			
	α	# dim	# dim	C	ϵ	γ
S						
–	60.0	–	–	10.0	0.5	0.001
PLS	275.0	50	36	20.0	0.5	6.25e−05
RFE	60.0	102	38	25.0	0.5	0.001
SB						
–	70.0	–	–	5.0	0.5	0.001
PLS	350.0	45	56	5.0	0.5	0.00025
RFE	10.0	16	70	125.0	0.25	0.0002
ST						
–	225.0	–	–	10.0	0.5	0.0005
PLS	300.0	30	45	10.0	0.5	0.0001
RFE	38.4	47	31	2.5	0.5	0.01
STB						
–	250.0	–	–	5.0	0.25	0.001
PLS	400.0	35	50	5.0	0.25	0.0002
RFE	7.5	54	27	10.0	0.5	0.01

Table 2 QET challenge baseline and best submission results

Exp.	Model	# dim	$\bar{\Delta}$	r_{rank}	RMSE	$\lvert\bar{\epsilon}\rvert$	$\overrightarrow{\lvert\epsilon\rvert}$
Baseline	SVR	17	0.55	0.58	0.82	0.69	0.86
QET best	M5P	42	0.63	0.64	0.75	0.61	–
QET best	SVR	42	0.61	0.60	0.78	0.64	–

Table 3 Results without dimensionality mapping and reduction with PLS or feature subset selection with RFE

Exp.	Model	# dim	$\bar{\Delta}$	r_{rank}	RMSE	$\lvert\bar{\epsilon}\rvert$	$\overrightarrow{\lvert\epsilon\rvert}$
S	RR	133	0.52	0.55	0.84	**0.68**	**0.85**
	SVR	133	0.47	0.52	0.86	0.7	0.87
SB	RR	150	**0.55**	**0.56**	**0.83**	0.68	**0.84**
	SVR	150	0.53	0.55	0.84	0.7	0.86
ST	RR	236	**0.58**	**0.6**	**0.81**	$\mathbf{0.65}^{p_1}$	**0.81**
	SVR	236	0.56	0.59	0.81	0.65^{p_2}	0.81
STB	RR	253	**0.59**	**0.61**	**0.8**	0.65^{p_3}	0.81
	SVR	253	0.57	0.59	0.81	0.66	0.82

dim corresponds to the number of features used. We bold the results that are discussed in the text. Statistically significant improvements over the baseline results with p values $\lesssim 0.05$ are super-scripted: $p_1 = 0.057$, $p_2 = 0.052$, $p_3 = 0.022$

Table 4 PLS results after dimensionality mapping and reduction with PLS

Exp.	Model	# dim	$\bar{\Delta}$	r_{rank}	RMSE	$\lvert\bar{\epsilon}\rvert$	$\overrightarrow{\lvert\epsilon\rvert}$
S	RR	50	0.51	0.54	0.85	0.68	**0.84**
	SVR	36	**0.51**	**0.55**	**0.84**	**0.68**	**0.84**
SB	RR	45	0.55	**0.57**	0.83	**0.67**	**0.83**
	SVR	56	**0.56**	**0.57**	**0.82**	**0.67**	**0.83**
ST	RR	30	0.57	0.58	0.82	0.65	0.81
	SVR	45	**0.57**	0.59	0.81	0.65^{p_1}	**0.8**
STB	RR	35	0.59	0.6	0.8	$\mathbf{0.64}^{p_2}$	**0.8**
	SVR	50	**0.59**	**0.6**	**0.8**	$\mathbf{0.64}^{p_3}$	**0.79**

We bold the results that improve the performance. STB becomes the top in QET according to r_{rank} and $\lvert\bar{\epsilon}\rvert$, showing that our feature set achieves the top performance in QET. Statistically significant improvements over the baseline results with p values $\lesssim 0.05$ are super-scripted: $p_1 = 0.038$, $p_2 = 0.01$, $p_3 = 0.007$

Table 5 FS results after feature subset selection with RFE

Setting	Model	# dim	$\bar{\Delta}$	r_{rank}	RMSE	$\lvert\bar{\epsilon}\rvert$	$\overrightarrow{\lvert\epsilon\rvert}$
S	RR	102	0.52	0.55	0.84	0.68	0.85
	SVR	38	0.48	0.52	0.86	0.7	0.87
SB	RR	16	0.53	0.55	0.84	0.69	0.85
	SVR	70	**0.54**	**0.56**	**0.83**	**0.68**	**0.84**
ST	RR	47	**0.6**	0.6	**0.8**	0.65^{p_1}	**0.8**
	SVR	31	0.53	0.57	0.83	0.68	0.84
STB	RR	54	**0.61**	0.61	**0.79**	$\mathbf{0.64}^{p_2}$	**0.8**
	SVR	27	0.52	0.56	0.84	0.68	0.84

We bold the results that improve the performance. We obtain the best results overall in setting STB using RR. Statistically significant improvements over the baseline results with p values $\lesssim 0.05$ are super-scripted: $p_1 = 0.02$, $p_2 = 0.012$

The results after PLS are given in Table 4 where improvements over no PLS are presented in bold. The performance of SVR improves after PLS as we observe in settings S and SB. The performance in general improves after PLS and more so for SVR. According to the official results, STB is placed 2nd from the top with respect to $\lvert\bar{\epsilon}\rvert$. STB SVR performs as good as the QET best SVR according to r_{rank} and $\lvert\bar{\epsilon}\rvert$, which shows that our representation of the modeling problem and our feature set can achieve the top performance in the QET challenge. The results after FS are given in Table 5 where improvements over no FS are presented in bold. Performance of the settings SB and STB improve and STB is placed 2nd from the top according to $\bar{\Delta}$ and $\lvert\bar{\epsilon}\rvert$.

The addition of QET baseline features helps the prediction; however, these features are glass-box features (Callison-Burch et al. 2012), which use information from the internal workings of an MT system. In our MTPP model we use only extrinsic features, or features that are ignorant of any information intrinsic to, and dependent on, a given SMT system. Our learning setting S solves the *blind translation quality prediction*

Table 6 Ranking of the top 38 features (remaining are not listed) selected by the RFE algorithm for settings ST and STB

S		SB	
RR 102	SVR 38	RR 16	SVR 70
3 g bpw	1&2 g wF_1	f4	f4
1 g GM	2 g vec|χ^2|	2 g vec|χ^2|	2 g vec|χ^2|
2 g vec|χ^2|	4 g log$p$	1 g GM	Link ppl 2
Max log$p$	1 g log$p$	1 g log$p$	Link ppl 5
Max log$p$ joint bpw	3 g F_1 bound	2 g bpw	1 g GM
Max log$p$ bpw	3 g BLEU bound	Max log$p$ bpw	1 g log$p$
Max log$p$2 bpw	Link ppl 2	Max log$p$ joint bpw	1&2 g wF_1
Link rec 1	Link ppl 5	Max log$p$3 bpw	Max log$p$ joint bpw
1 g log$p$	Max log$p$3 joint bpw	Link GM 1	Max log$p$3 joint bpw
Link log$p$ 3	Max log$p$ joint bpw	Link rec 1	4 g bpw
2 g bpw	4 g bpw	Link prec 1	5 g bpw
Avg BLEU20	5 g bpw	Link log$p$ 3	3 g F_1 bound
3 g vec|χ^2|	2 g bpw	3 g F_1 bound	3 g BLEU bound
Max log$p$2 joint bpw	2 g wF_1	3 g BLEU bound	Max log$p$ bpw
MBRR log$p$ joint	1 g GM	Avg BLEU20	Max log$p$3 bpw
1&2 g GM	Max log$p$ bpw	Avg BLEU5	f3
1 g F_1 bound	Max log$p$3 bpw		2 g log$p$
Avg BLEU5	Link ppl 4		2 g bpw
Avg $F_1$3	Max log$p$3 joint		Link ppl 4
Coverscore	1 g vec|χ^2|		1 g vec|χ^2|
Avg BLEU10	Max log$p$ joint		f1
Avg BLEU3	1 g F_1 bound		1&2&3 g prec
4 g bpw	1 g BLEU bound		Link log$p$ 3
Link GM 1	1 g rec		Link GM 1
Link oov 1	Link log$p$ 3		Max log$p$3 joint
1 g oov	Link GM 1		Max log$p$ joint
Link log$p$ 2	1&2&3 g prec		2 g wF_1
1 Link vec|χ^2|	1&2&3 g F_1		1 g F_1 bound
2 g prec	3 g vec|χ^2|		1 g BLEU bound
2 g F_1	Link log$p$ 1		1 g rec
1 g bpw	Link rec 1		1&2&3 g F_1
3 g F_1 bound	1 g prec		3 g vec|χ^2|
3 g rec	2 g ppl		Link wF_1 1
2 g ppl	3 g ppl		Link rec 1
1 g ppl	2 g log$p$		Link F_1 1
Max log$p$3 bpw	3 g log$p$		2 g GM

Table 6 continued

S		SB	
RR	SVR	RR	SVR
102	38	16	70
2 g BLEU bound	Link wF_1 1		2 g F_1 bound
2 g rec	3 g prec		Link prec 1

The third row lists the number of features selected

problem by being totally uninformed of the translation output the quality of which is being predicted. Our most informed learning setting ST only uses extrinsic information to derive its features. Learning after feature selection with RFE achieves higher top performance than learning after applying PLS. Overall, we obtain the best results in the STB setting using RR after RFE. Analysis of the worst 10 predictions made by the best system reveal that the prediction system tends to make larger absolute errors on very short sentences (3/10, with 5 or less words), which are harder to obtain features for, or on inverted sentences (6/10), which the LM features do not favor, or on sentences with quotations (1/10). The performance of the QET baseline does not improve after PLS or FS. A high $|\overrightarrow{\epsilon}|$ is an indicator that the quality estimation task is hard and currently, we can only reduce the error with respect to knowing and predicting the mean by about 20 %. We observe that even in the ST setting, $|\overrightarrow{\epsilon}|$ is 0.79, which may be due to the use of a single SMT model in QET, making the translations similar and therefore indistinguishable.

5.1 Feature subset selection results

We list the ranking of the top features selected by the RFE algorithm for settings S and SB in Table 6 and for settings ST and STB in Table 7. The features are listed starting from rank 1 arranged top to bottom and the third row in each table lists the number of features selected. The ranking of the top features selected for each learning setting provides an exhaustive analysis of the features used. The most time consuming part of our MTPP model is the feature extraction step. Since the size of the training and test sets are small in the QET, training and testing is relatively fast. If we can identify the most relevant features for achieving the top or close to the top performance, we can focus on only deriving those important features. Therefore, feature subset selection and ranking is very important for guiding further modeling efforts.

We observe that $f4$ is ranked highly, which is the LM score of source sentences and it is adding new information since we only used the training corpus for training our LM. We also observe that $f7$ is among the features selected, which is based on the phrase table used by the SMT model generating the translations. Below we describe the abbreviations used for features in Tables 6 and 7. If a number is present in the abbreviation, it corresponds to the order of the n-grams or the LM order. For instance, *link ppl* 5 corresponds to the perplexity score using a 5-g LM over CCLs. 1 *gram GM*

Table 7 Ranking of the top 31 features (remaining are not listed) features selected by the RFE algorithm for settings ST and STB

ST		STB	
RR 47	SVR 31	RR 54	SVR 27
T log$p$ bpw	1 g GM	f4	T link F_1 1
2 g bpw	2 g vec$\|\chi^2\|$	2 g vec$\|\chi^2\|$	T link wF_1 1
T 1 g bpw	2 g wF_1	1 g GM	f4
T 5 g bpw	Link ppl 2	T 1 g log$p$	2 g vec$\|\chi^2\|$
1 g GM	Link ppl 4	T log$p$ bpw	Link ppl 2
2 g vec$\|\chi^2\|$	T 2 g ppl	T Link log$p$ 1	Link ppl 5
Link log$p$ 3	T 5 g ppl	T 5 g bpw	1 g GM
Link rec 1	T 4 g ppl	T 1 g bpw	1 g log$p$
T 4 g bpw	4 g bpw	2 g bpw	T 1 g f
Max log$p$2 bpw	5 g bpw	T 2 g ppl	T log$p$
Max log$p$ bpw	Link ppl 5	T 5 g ppl	T 2 g ppl
Max log$p$ joint bpw	T log$p$ joint	Max log$p$ joint bpw	T 5 g ppl
Avg BLEU20	1 g log$p$	Max log$p$3 joint bpw	2 g wF_1
Avg BLEU5	Max log$p$ joint bpw	Max log$p$ bpw	T 4 g ppl
T 1 g ppl	Max log$p$ bpw	Max log$p$3 bpw	4 g bpw
T 5 g ppl	Max log$p$3 joint bpw	Max log$p$ joint	5 g bpw
Avg $F_1$3	Max log$p$3 bpw	1 g F_1 bound	Link ppl 4
MBRR log$p$ joint	T link bpw 3	T 2 g log$p$	Max log$p$ joint bpw
T link log$p$ 3	T link bpw 4	T 1 g oov	Max log$p$ bpw
1 g log$p$	T 1 g f	T 2 g oov	Max log$p$3 joint bpw
2 g f	1 g F_1 bound	T 1 g GM	Max log$p$3 bpw
T 2 g oov	1 g rec	f5	Link GM 1
3 g bpw	Link wF_1 1	1 g log$p$	Link rec 1
T 1 g oov	Link GM 1	Link log$p$ 3	3 g vec$\|\chi^2\|$
1 g F_1 bound	Link rec 1	T 3 g wGM	3 g prec
1&2 g GM	T 5 g bpw	T 3 g wrec	Link F_1 1
2 g prec	2 g ppl	T 3 g prec	Link log$p$ 3
T 3 g F_1	3 g ppl	Avg $F_1$5	
T 3 g rec	2 g bpw	Avg $F_1$3	
T gavg BLEUTS	T 1 g bpw	Avg BLEU20	
2 g F_1	T 1 g ppl	Avg BLEU3	

uses the geometric mean between the precision and recall for 1-g. *max logp k* is the Bayes IBM1 translation log probability for top k translations (we omit k when $k = 1$). *MBRR logp joint* is the log $P(S, T)$ of the MBRR translation. *avg [BLEU/F_1/$\phi_{\mathfrak{D}}$]n* is the average BLEU or F_1 or $\phi_{\mathfrak{D}}$ score of the top n closest training instances selected by the Dice scorer, $\phi_{\mathfrak{D}}$. *link logp 3* corresponds to the 3-g log probability score obtained

with a LM over CCLs. 2 *gram f* is the number of 2-g in a given sentence. T specifies that the feature is calculated over the target translation T', which can be observed in settings ST and STB.

6 Conclusion

Accurate prediction of translation quality is useful for estimating the post-editing effort for correcting MT translation outputs, for domain adaptation in MT, and for selecting better SMT models. We define extrinsic and blind translation quality prediction problems and present our MTPP model for solving both. MTPP relies only on extrinsic features, which leads to faster training. We describe machine learning techniques including partial least squares and recursive feature elimination, which improve the performance. Overall, we develop novel, language independent, and SMT system extrinsic features for predicting the SMT performance, which also rank highly during feature ranking and selection evaluations.

Our experiments involving different learning settings help differentiate the contribution of different feature sets. Our results achieve top performance for automatic, accurate, and language independent prediction of sentence-level SMT quality in the QET challenge with or without looking at the translations. Our representation and features allow us to obtain as precise results as the baseline or better by just looking at the test source sentences and not using their translations and it enables us to achieve the top performance in the QET challenge among the models using the SVR learning model and 2nd overall.

Acknowledgments This work was supported in part by SFI (07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University and in part by the European Commission through the QTLaunchPad FP7 project (No: 296347). The authors wish to acknowledge the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support. We also thank the reviewers for their constructive comments.

References

Albrecht JS, Hwa R (2008) Regression for machine translation evaluation at the sentence level. Mach Transl 22(1–2):1–27

Bicici E (2011) The regression model of machine translation. PhD thesis, Koç University.

Bicici E, Yuret D (2011) Instance selection for machine translation using feature decay algorithms. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Edinburgh, pp 272–283

Birch A, Osborne M, Koehn P (2008) Predicting success in machine translation. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Honolulu, pp 745–754

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

Brown PF, Pietra SAD, Pietra VJD, Mercer RL (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist 19(2):263–311

Callison-Burch C, Koehn P, Monz C, Post M, Soricut R, Specia L (2012) Findings of the 2012 workshop on statistical machine translation. In: Proceedings of the Seventh Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montréal, pp 10–51

Cover TM, Thomas JA (2006) Elements of information theory. Wiley-Interscience, New York

Drucker H, Burges CJC, Kaufman L, Smola AJ, Vapnik V (1996) Support vector regression machines. In: NIPS, pp 155–161

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining inference and prediction, 2nd edn. Springer-Verlag, New York

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

Knight K (1999) A statistical machine translation tutorial workbook. http://www.isi.edu/natural-language/mt/wkbk.rtf

Koehn P (2010) Statistical machine translation. Cambridge University Press, Cambridge

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: Annual Meeting of the Association for Computational Linguistics, Prague, pp 177–180

Moore RC (2002) Fast and accurate sentence alignment of bilingual corpora. In: AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users, Springer-Verlag, London, pp 135–144

Och FJ, Ney H (2003) A systematic comparison of various statistical alignment models. Comput Linguist 29(1):19–51

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, pp 311–318

Rosipal R, Trejo LJ (2001) Kernel partial least squares regression in reproducing kernel hilbert space. J Mach Learn Res 2:97–123

Sarle WS (2002) Faq: should i normalize/standardize/rescale the data? http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-16.html

Seginer Y (2007) Learning syntactic structure. PhD thesis, Universiteit van Amsterdam.

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Soricut R, Bach N, Wang Z (2012) The SDL language weaver systems in the wmt12 quality estimation shared task. In: Proceedings of the Seventh Workshop on Statistical Machine Translation. Association for Computational Linguistics, Montréal, pp 145–151

Specia L, Cancedda N, Dymetman M, Turchi M, Cristianini N (2009) Estimating the sentence-level quality of machine translation systems. In: Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT), Barcelona, pp 28–35

Specia L, Raj D, Turchi M (2010) Machine translation evaluation versus quality estimation. Mach Transl 24(1):39–50

Wasserman L (2004) All of statistics: a concise course in statistical inference. Springer, New York