Prediction of Learning Curves in Machine Translation

Prasanth Kolachina[*] **Nicola Cancedda**[†] **Marc Dymetman**[†] **Sriram Venkatapathy**[†]
∗ LTRC, IIIT-Hyderabad, Hyderabad, India
† Xerox Research Centre Europe, 6 chemin de Maupertuis, 38240 Meylan, France

Abstract

Parallel data in the domain of interest is the key resource when training a statistical machine translation (SMT) system for a specific purpose. Since ad-hoc manual translation can represent a significant investment in time and money, a prior assesment of the amount of training data required to achieve a satisfactory accuracy level can be very useful. In this work, we show how to *predict* what the learning curve would look like *if* we were to manually translate increasing amounts of data.

We consider two scenarios, *1*) Monolingual samples in the source and target languages are available and *2*) An additional small amount of parallel corpus is also available. We propose methods for predicting learning curves in both these scenarios.

1 Introduction

Parallel data in the domain of interest is the key resource when training a statistical machine translation (SMT) system for a specific business purpose. In many cases it is possible to allocate some budget for manually translating a limited sample of relevant documents, be it via professional translation services or through increasingly fashionable crowdsourcing. However, it is often difficult to predict how much training data will be required to achieve satisfactory translation accuracy, preventing sound provisional budgetting. This prediction, or more generally the prediction of the *learning curve* of an SMT system as a function of available in-domain parallel data, is the objective of this paper.

We consider two scenarios, representative of realistic situations.

1. In the first scenario (S1), the SMT developer is given only monolingual source and target samples from the relevant domain, and a small *test* parallel corpus.
2. In the second scenario (S2), an additional small *seed* parallel corpus is given that can be used to train small in-domain models and measure (with some variance) the evaluation score at a few points on the initial portion of the learning curve.

In both cases, the task consists in predicting an evaluation score (BLEU, throughout this work) on the test corpus as a function of the size of a subset of the source sample, assuming that we could have it manually translated and use the resulting bilingual corpus for training.

In this paper we provide the following contributions:

1. An extensive study across six parametric function families, empirically establishing that a certain three-parameter power-law family is well suited for modeling learning curves for the Moses SMT system when the evaluation score is BLEU. Our methodology can be easily generalized to other systems and evaluation scores (Section 3);
2. A method for *inferring* learning curves based on features computed from the resources available in scenario S1, suitable for both the scenarios described above (S1) and (S2) (Section 4);
3. A method for *extrapolating* the learning curve from a few measurements, suitable for scenario S2 (Section 5);
4. A method for *combining* the two approaches above, achieving on S2 better prediction accuracy than either of the two in isolation (Section 6).

In this study we limit tuning to the mixing parameters of the Moses log-linear model through MERT, keeping all meta-parameters (e.g. maximum phrase length, maximum allowed distortion, etc.) at their default values. One can expect further tweaking to lead to performance improvements, but this was a

[*] This research was carried out during an internship at Xerox Research Centre Europe.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 22–30,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

necessary simplification in order to execute the tests on a sufficiently large scale.

Our experiments involve 30 distinct language pair and domain combinations and 96 different learning curves. They show that without any parallel data we can predict the expected translation accuracy at 75K segments within an error of 6 BLEU points (Table 4), while using a seed training corpus of 10K segments narrows this error to within 1.5 points (Table 6).

2 Related Work

Learning curves are routinely used to illustrate how the performance of experimental methods depend on the amount of training data used. In the SMT area, Koehn et al. (2003) used learning curves to compare performance for various meta-parameter settings such as *maximum phrase length*, while Turchi et al. (2008) extensively studied the behaviour of learning curves under a number of test conditions on Spanish-English. In Birch et al. (2008), the authors examined corpus features that contribute most to the machine translation performance. Their results showed that the most predictive features were the morphological complexity of the languages, their linguistic relatedness and their word-order divergence; in our work, we make use of these features, among others, for predicting translation accuracy (Section 4).

In a Machine Learning context, Perlich et al. (2003) used learning curves for predicting *maximum performance* bounds of learning algorithms and to compare them. In Gu et al. (2001), the learning curves of two classification algorithms were modelled for eight different large data sets. This work uses similar *a priori* knowledge for restricting the form of learning curves as ours (see Section 3), and also similar empirical evaluation criteria for comparing curve families with one another. While both application and performance metric in our work are different, we arrive at a similar conclusion that a power law family of the form $y = c - a\, x^{-\alpha}$ is a good model of the learning curves.

Learning curves are also frequently used for determining empirically the number of iterations for an incremental learning procedure.

The crucial difference in our work is that in the previous cases, learning curves are plotted *a posteriori* i.e. once the labelled data has become available and the training has been performed, whereas in our work the learning curve itself is the object of the prediction. Our goal is to learn to *predict* what the learning curve will be *a priori* without having to label the data at all (S1), or through labelling only a very small amount of it (S2).

In this respect, the academic field of Computational Learning Theory has a similar goal, since it strives to identify bounds to performance measures[1], typically including a dependency on the training sample size. We take a purely empirical approach in this work, and obtain useful estimations for a case like SMT, where the complexity of the mapping between the input and the output prevents tight theoretical analysis.

3 Selecting a parametric family of curves

The first step in our approach consists in selecting a suitable family of shapes for the learning curves that we want to produce in the two scenarios being considered.

We formulate the problem as follows. For a certain bilingual test dataset d, we consider a set of observations $O_d = \{(x_1, y_1), (x_2, y_2)...(x_n, y_n)\}$, where y_i is the performance on d (measured using BLEU (Papineni et al., 2002)) of a translation model trained on a parallel corpus of size x_i. The corpus size x_i is measured in terms of the number of segments (sentences) present in the parallel corpus.

We consider such observations to be generated by a regression model of the form:

$$y_i = F(x_i; \theta) + \epsilon_i \qquad 1 \leq i \leq n \tag{1}$$

where F is a function depending on a vector parameter θ which depends on d, and ϵ_i is Gaussian noise of constant variance.

Based on our prior knowledge of the problem, we limit the search for a suitable F to families that satisfies the following conditions- monotonically increasing, concave and bounded. The first condition just says that more training data is better. The second condition expresses a notion of "diminishing returns", namely that a given amount of additional training data is more advantageous when added to a small rather than to a big amount of initial data. The last condition is related to our use of BLEU — which is bounded by 1 — as a performance measure; It should be noted that some growth patterns which are sometimes proposed, such as a logarithmic regime of the form $y \simeq a + b \log x$, are not

[1] More often to a *loss*, which is equivalent.

compatible with this constraint.

We consider six possible families of functions satisfying these conditions, which are listed in Table 1. Preliminary experiments indicated that curves from the "Power" and "Exp" family with only two parameters underfitted, while those with five or more parameters led to overfitting and solution instability. We decided to only select families with three or four parameters.

Model	**Formula**
Exp_3	$y = c - e^{-ax+b}$
Exp_4	$y = c - e^{-ax^{\alpha}+b}$
ExpP_3	$y = c - e^{(x-b)^{\alpha}}$
Pow_3	$y = c - ax^{-\alpha}$
Pow_4	$y = c - (-ax+b)^{-\alpha}$
ILog_2	$y = c - (a/\log x)$

Table 1: Curve families.

Curve fitting technique Given a set of observations $\{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$ and a curve family $F(x; \theta)$ from Table 1, we compute a best fit $\hat{\theta}$ where:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} [y_i - F(x_i; \theta)]^2, \qquad (2)$$

through use of the *Levenberg-Marquardt* method (Moré, 1978) for non-linear regression.

For selecting a learning curve family, and for all other experiments in this paper, we trained a large number of systems on multiple *configurations* of training sets and sample sizes, and tested each on multiple *test* sets; these are listed in Table 2. All experiments use Moses (Koehn et al., 2007). [2]

Domain	**Source Language**	**Target Language**	**# Test sets**
Europarl (Koehn, 2005)	Fr, De, Es En	En Fr, De, Es	4
KFTT (Neubig, 2011)	Jp, En	En, Jp	2
EMEA (Tiedemann, 2009)	Da, De	En	4
News (Callison-Burch et al., 2011)	Cz,En,Fr,De,Es	Cz,En,Fr,De,Es	3

Table 2: The translation systems used for the curve fitting experiments, comprising 30 language-pair and domain combinations for a total of 96 learning curves. Language codes: Cz=Czech, Da=Danish, En=English, De=German, Fr=French, Jp=Japanese, Es=Spanish

The goodness of fit for each of the families is evaluated based on their ability to *i*) fit over the entire set of observations, *ii*) extrapolate to points beyond the observed portion of the curve and *iii*) generalize well over different datasets .

We use a recursive fitting procedure where the curve obtained from fitting the first i points is used to predict the observations at two points: x_{i+1}, i.e. the point to the immediate right of the currently observed x_i and x_n, i.e. the largest point that has been observed.

The following error measures quantify the goodness of fit of the curve families:

1. Average root mean-squared error (RMSE):

$$\frac{1}{N} \sum_{c \in S} \sum_{t \in T_c} \left\{ \frac{1}{n} \sum_{i=1}^{n} [y_i - F(x_i; \hat{\theta})]^2 \right\}_{ct}^{1/2}$$

where S is the set of training datasets, T_c is the set of test datasets for training configuration c, $\hat{\theta}$ is as defined in Eq. 2, N is the total number of combinations of training configurations and test datasets, and i ranges on a grid of training subset sizes.The expressions $n, x_i, y_i, \hat{\theta}$ are all local to the combination ct.

2. Average root mean squared residual at next point $X = x_{i+1}$ (NPR):

$$\frac{1}{N} \sum_{c \in S} \sum_{t \in T_c} \left\{ \frac{1}{n-k-1} \sum_{i=k}^{n-1} [y_{i+1} - F(x_{i+1}; \hat{\theta}^i)]^2 \right\}_{ct}^{1/2}$$

where $\hat{\theta}^i$ is obtained using only observations up to x_i in Eq. 2 and where k is the number of parameters of the family.[3]

3. Average root mean squared residual at the last point $X = x_n$ (LPR):

$$\frac{1}{N} \sum_{c \in S} \sum_{t \in T_c} \left\{ \frac{1}{n-k-1} \sum_{i=k}^{n-1} [y_n - F(x_n; \hat{\theta}^i)]^2 \right\}_{ct}^{1/2}$$

Curve fitting evaluation The evaluation of the goodness of fit for the curve families is presented in Table 3. The average values of the root mean-squared error and the average residuals across all the learning curves used in our experiments are shown in this table. The values are on the same scale as the BLEU scores. Figure 1 shows the curve fits obtained

[2] The settings used in training the systems are those described in `http://www.statmt.org/wmt11/baseline.html`

[3] We start the summation from $i = k$, because at least k points are required for computing $\hat{\theta}^i$.

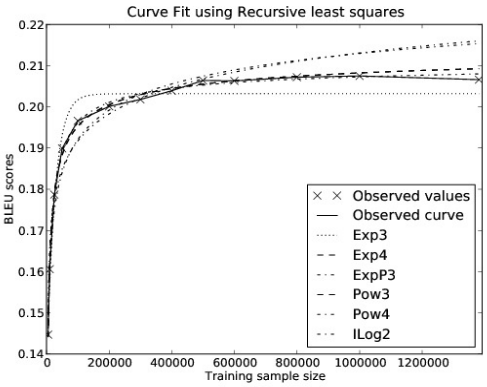

Figure 1: Curve fits using different curve families on a test dataset

for all the six families on a test dataset for English-German language pair.

Curve Family	RMSE	NPR	LPR
Exp_3	0.0063	0.0094	0.0694
Exp_4	0.0030	*0.0036*	*0.0072*
ExpP_3	0.0040	0.0049	0.0145
Pow_3	**0.0029**	**0.0037**	**0.0091**
Pow_4	*0.0026*	0.0042	0.0102
ILog_2	0.0050	0.0067	0.0146

Table 3: Evaluation of the goodness of fit for the six families.

Loooking at the values in Table 3, we decided to use the Pow_3 family as the best overall compromise. While it is not systematically better than Exp_4 and Pow_4, it is good overall and has the advantage of requiring only 3 parameters.

4 Inferring a learning curve from mostly monolingual data

In this section we address scenario S1: we have access to a source-language monolingual collection (from which portions to be manually translated could be sampled) and a target-language in-domain monolingual corpus, to supplement the target side of a parallel corpus while training a language model. The only available parallel resource is a very small test corpus. Our objective is to predict the evolution of the BLEU score on the given test set as a function of the size of a random subset of the training data that we manually translate[4]. The intuition behind this is that the source-side and target-side monolingual data already convey significant information about the difficulty of the translation task.

We proceed in the following way. We first train models to predict the BLEU score at m *anchor* sizes $s_1, \ldots, s_m$, based on a set of features globally characterizing the configuration of interest. We restrict our attention to linear models:

$$\mu_j = \boldsymbol{w}_j{}^\top \boldsymbol{\phi}, j \in \{1 \ldots m\}$$

where $\boldsymbol{w}_j$ is a vector of feature weights specific to predicting at anchor size j, and $\boldsymbol{\phi}$ is a vector of size-independent configuration features, detailed below. We then perform inference using these models to predict the BLEU score at each anchor, for the test case of interest. We finally estimate the parameters of the learning curve by weighted least squares regression using the anchor predictions.

Anchor sizes can be chosen rather arbitrarily, but must satisfy the following two constraints:

1. They must be three or more in number in order to allow fitting the tri-parameter curve.
2. They should be spread as much as possible along the range of sample size.

For our experiments, we take $m = 3$, with anchors at 10K, 75K and 500K segments.

The feature vector $\boldsymbol{\phi}$ consists of the following features:

1. General properties: number and average length of sentences in the (source) test set.
2. Average length of tokens in the (source) test set and in the monolingual source language corpus.
3. Lexical diversity features:
   (a) type-token ratios for n-grams of order 1 to 5 in the monolingual corpus of both source and target languages
   (b) perplexity of language models of order 2 to 5 derived from the monolingual source corpus computed on the source side of the test corpus.

[4] We specify that it is a random sample as opposed to a subset deliberately chosen to maximize learning effectiveness. While there are clear ties between our present work and active learning, we prefer to keep these two aspects distinct at this stage, and intend to explore this connection in future work.

4. Features capturing divergence between languages in the pair:
   (a) average ratio of source/target sentence lengths in the test set.
   (b) ratio of type-token ratios of orders 1 to 5 in the monolingual corpus of both source and target languages.

5. Word-order divergence: The divergence in the word-order between the source and the target languages can be captured using the part-of-speech (pos) tag sequences across languages. We use cross-entropy measure to capture similarity between the n-gram distributions of the pos tags in the monolingual corpora of the two languages. The order of the n-grams ranges between $n = 2, 4 \dots 12$ in order to account for long distance reordering between languages. The pos tags for the languages are mapped to a reduced set of twelve pos tags (Petrov et al., 2012) in order to account for differences in tagsets used across languages.

These features capture our intuition that translation is going to be harder if the language in the domain is highly variable and if the source and target languages diverge more in terms of morphology and word-order.

The weights $\boldsymbol{w}_j$ are estimated from data. The training data for fitting these linear models is obtained in the following way. For each configuration (combination of language pair and domain) c and test set t in Table 2, a *gold curve* is fitted using the selected tri-parameter power-law family using a fine grid of corpus sizes. This is available as a byproduct of the experiments for comparing different parametric families described in Section 3. We then compute the value of the gold curves at the m anchor sizes: we thus have m "gold" vectors $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m$ with accurate estimates of BLEU at the anchor sizes[5]. We construct the design matrix $\boldsymbol{\Phi}$ with one column for each feature vector $\boldsymbol{\phi}_{ct}$ corresponding to each combination of training configuration c and test set t.

We then estimate weights $\boldsymbol{w}_j$ using Ridge regression (L^2 regularization):

$$\boldsymbol{w}_j = \arg\min_{\boldsymbol{w}} ||\boldsymbol{\Phi}^\top \boldsymbol{w} - \boldsymbol{\mu}_j||^2 + C||\boldsymbol{w}||^2 \qquad (3)$$

where the regularization parameter C is chosen by cross-validation. We also run experiments using Lasso (L^1) regularization (Tibshirani, 1994) instead of Ridge. As baseline, we take a *constant mean* model predicting, for each anchor size s_j, the average of all the μ_{jct}.

We do not assume the difficulty of predicting BLEU at all anchor points to be the same. To allow for this, we use (non-regularized) *weighted* least-squares to fit a curve from our parametric family through the m anchor points[6]. Following (Croarkin and Tobias, 2006, Section 4.4.5.2), the *anchor confidence* is set to be the inverse of the cross-validated mean square residuals:

$$\omega_j = \left(\frac{1}{N} \sum_{c \in S} \sum_{t \in T_c} (\boldsymbol{\phi}_{ct}^\top \boldsymbol{w}_j^{\backslash c} - \mu_{jct})^2 \right)^{-1} \qquad (4)$$

where $\boldsymbol{w}_j^{\backslash c}$ are the feature weights obtained by the regression above on all training configurations except c, μ_{jct} is the gold value at anchor j for training/test combination c, t, and N is the total number of such combinations[7]. In other words, we assign to each anchor point a confidence inverse to the cross-validated mean squared error of the model used to predict it.

For a new unseen configuration with feature vector ϕ_u, we determine the parameters θ_u of the corresponding learning curve as:

$$\theta_u = \arg\min_\theta \sum_j \omega_j \left(F(s_j; \theta) - \boldsymbol{\phi}_{\boldsymbol{u}}^\top \boldsymbol{w}_j \right)^2 \qquad (5)$$

5 Extrapolating a learning curve fitted on a small parallel corpus

Given a small "seed" parallel corpus, the translation system can be used to train small in-domain models and the evaluation score can be measured at a few initial sample sizes $\{(x_1, y_1), (x_2, y_2) ... (x_p, y_p)\}$. The performance of the system for these initial points provides evidence for predicting its performance for larger sample sizes.

In order to do so, a learning curve from the family Pow$_3$ is first fit through these initial points. We

[5] Computing these values from the gold curve rather than directly from the observations has the advantage of smoothing the observed values and also does not assume that observations at the anchor sizes are always directly available.

[6] When the number of anchor points is the same as the number of parameters in the parametric family, the curve can be fit exactly through all anchor points. However the general discussion is relevant in case there are more anchor points than parameters, and also in view of the combination of inference and extrapolation in Section 6.

[7] Curves on different test data for the same training configuration are highly correlated and are therefore left out.

assume that $p \geq 3$ for this operation to be well-defined. The best fit $\hat{\eta}$ is computed using the same curve fitting as in Eq. 2.

At each individual anchor size s_j, the accuracy of prediction is measured using the root mean-squared error between the prediction of extrapolated curves and the gold values:

$$\left(\frac{1}{N}\sum_{c\in S}\sum_{t\in T_c}[F(s_j;\hat{\eta}_{ct})-\mu_{ctj}]^2\right)^{1/2} \tag{6}$$

where $\hat{\eta}_{ct}$ are the parameters of the curve fit using the initial points for the combination ct.

In general, we observed that the extrapolated curve tends to over-estimate BLEU for large samples.

6 Combining inference and extrapolation

In scenario S2, the models trained from the seed parallel corpus and the features used for inference (Section 4) provide complementary information. In this section we combine the two to see if this yields more accurate learning curves.

For the inference method of Section 4, predictions of models at anchor points are weighted by the inverse of the model empirical squared error (ω_j). We extend this approach to the extrapolated curves. Let u be a new configuration with seed parallel corpus of size x_u, and let x_l be the largest point in our grid for which $x_l \leq x_u$. We first train translation models and evaluate scores on samples of size $x_1, \ldots, x_l$, fit parameters $\hat{\eta}_u$ through the scores, and then extrapolate BLEU at the anchors s_j: $F(s_j;\hat{\eta}_u), j \in \{1, \ldots, m\}$. Using the models trained for the experiments in Section 3, we estimate the squared extrapolation error at the anchors s_j when using models trained on size up to x_l, and set the confidence in the extrapolations[8] for u to its inverse:

$$\xi_j^{<l} = \left(\frac{1}{N}\sum_{c\in S}\sum_{t\in T_c}(F(s_j;\eta_{ct}^{<l})-\mu_{ctj})^2\right)^{-1} \tag{7}$$

where N, S, T_c and μ_{ctj} have the same meaning as in Eq. 4, and $\eta_{ct}^{<l}$ are parameters fitted for configuration c and test t using only scores measured at $x_1, \ldots, x_l$. We finally estimate the parameters θ_u of the combined curve as:

$$\begin{aligned}\theta_u = \arg\min_{\theta}\sum_j &\omega_j(F(s_j;\theta)-\phi_u^{\top}\boldsymbol{w}_j)^2 \\ &+ \xi_j^{<l}(F(s_j;\theta)-F(s_j;\hat{\eta}_u))^2\end{aligned}$$

where ϕ_u is the feature vector for u, and w_j are the weights we obtained from the regression in Eq. 3.

7 Experiments

In this section, we report the results of our experiments on predicting the learning curves.

7.1 Inferred Learning Curves

Regression model	**10K**	**75K**	**500K**
Ridge	0.063	**0.060**	**0.053**
Lasso	**0.054**	**0.060**	0.062
Baseline	0.112	0.121	0.121

Table 4: Root mean squared error of the linear regression models for each anchor size

In the case of inference from mostly monolingual data, the accuracy of the predictions at each of the anchor sizes is evaluated using root mean-squared error over the predictions obtained in a *leave-one-out* manner over the set of configurations from Table 2. Table 4 shows these results for Ridge and Lasso regression models at the three anchor sizes. As an example, the model estimated using Lasso for the 75K anchor size exhibits a root mean squared error of 6 BLEU points. The errors we obtain are lower than the error of the baseline consisting in taking, for each anchor size s_j, the average of all the μ_{ctj}. The Lasso regression model selected four features from the entire feature set: *i*) Size of the test set (sentences & tokens) *ii*) Perplexity of language model (order 5) on the test set *iii*) Type-token ratio of the target monolingual corpus . Feature correlation measures such as Pearsons R showed that the features corresponding to type-token ratios of both source and target languages and size of test set have a high correlation with the BLEU scores at the three anchor sizes.

Figure 2 shows an instance of the inferred learning curves obtained using a weighted least squares method on the predictions at the anchor sizes. Table 7 presents the cumulative error of the inferred learning curves with respect to the gold curves, measured as the average distance between the curves in the range $x \in [0.1K, 100K]$.

[8] In some cases these can actually be interpolations.

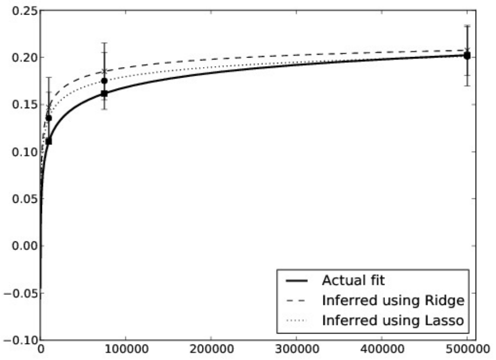

Figure 2: Inferred learning curve for English-Japanese test set. The error-bars show the *anchor confidence* for the predictions.

7.2 Extrapolated Learning Curves

As explained in Section 5, we evaluate the accuracy of predictions from the *extrapolated* curve using the root mean squared error (see Eq. 6) between the predictions of this curve and the gold values at the anchor points.

We conducted experiments for three sets of initial points, *1*) 1K-5K-10K, *2*) 5K-10K-20K, and *3*) 1K-5K-10K-20K. For each of these sets, we show the prediction accuracy at the anchor sizes, 10K[9], 75K, and 500K in Table 5.

Initial Points	10K	75K	500K
1K-5K-10K	0.005	0.017	0.042
5K-10K-20K	0.002	0.015	0.034
1K-5K-10K-20K	**0.002**	**0.008**	**0.019**

Table 5: Root mean squared error of the extrapolated curves at the three anchor sizes

The root mean squared errors obtained by extrapolating the learning curve are much lower than those obtained by prediction of translation accuracy using the monolingual corpus only (see Table 4), which is expected given that more direct evidence is available in the former case . In Table 5, one can also see that the root mean squared error for the sets 1K-5K-10K and 5K-10K-20K are quite close for anchor sizes 75K and 500K. However, when a configuration of four initial points is used for the same amount of "seed" parallel data, it outperforms both the configurations with three initial points.

7.3 Combined Learning Curves and Overall Comparison

In Section 6, we presented a method for combining the predicted learning curves from inference and extrapolation by using a weighted least squares approach. Table 6 reports the root mean squared error at the three anchor sizes from the combined curves.

Initial Points	Model	10K	75K	500K
1K-5K-10K	Ridge	0.005	0.015	0.038
	Lasso	0.005	0.014	0.038
5K-10K-20K	Ridge	0.001	0.006	0.018
	Lasso	0.001	0.006	0.018
1K-5K-10K-20K	Ridge	**0.001**	**0.005**	**0.014**
	Lasso	**0.001**	**0.005**	**0.014**

Table 6: Root mean squared error of the combined curves at the three anchor sizes

We also present an overall evaluation of all the predicted learning curves. The evaluation metric is the average distance between the predicted curves and the gold curves, within the range of sample sizes x_{min}=0.1K to x_{max}=500K segments; this metric is defined as:

$$\frac{1}{N}\sum_{c\in S}\sum_{t\in T_c}\frac{\sum_{x=x_{min}}^{x_{max}}|F(x;\hat{\eta}_{ct})-F(x;\hat{\theta}_{ct})|}{x_{max}-x_{min}}$$

where $\hat{\eta}_{ct}$ is the curve of interest, $\hat{\theta}_{ct}$ is the gold curve, and x is in the range $[x_{min}, x_{max}]$, with a step size of 1. Table 7 presents the final evaluation.

Initial Points	IR	IL	EC	CR	CL
1K-5K-10K	0.034	0.050	0.018	0.015	**0.014**
5K-10K-20K	0.036	0.048	0.011	0.010	**0.009**
1K-5K-10K-20K	0.032	0.049	0.008	**0.007**	**0.007**

Table 7: Average distance of different predicted learning curves relative to the gold curve. Columns: IR="Inference using Ridge model", IL="Inference using Lasso model", EC="Extrapolated curve", CR="Combined curve using Ridge", CL="Combined curve using Lasso"

We see that the combined curves (CR and CL) perform slightly better than the inferred curves (IR

[9] The 10K point is not an extrapolation point but lies within the range of the set of initial points. However, it does give a measure of the closeness of the curve fit using only the initial points with the gold fit using all the points; the value of this gold fit at 10K is not necessarily equal to the observation at 10K.

and IL) and the extrapolated curves (EC). The average distance is on the same scale as the BLEU score, which suggests that our best curves can predict the gold curve within 1.5 BLEU points on average (the best result being 0.7 BLEU points when the initial points are 1K-5K-10K-20K) which is a telling result. The distances between the predicted and the gold curves for all the learning curves in our experiments are shown in Figure 3.

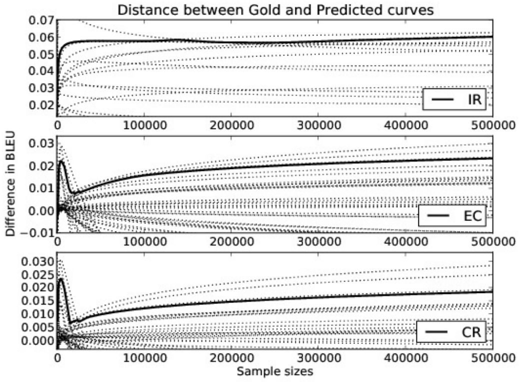

Figure 3: Distances between the predicted and the gold learning curves in our experiments across the range of sample sizes. The dotted lines indicate the distance from gold curve for each instance, while the bold line indicates the 95^{th} quantile of the distance between the curves. IR="Inference using Ridge model", EC="Extrapolated curve", CR="Combined curve using Ridge".

We also provide a comparison of the different predicted curves with respect to the gold curve as shown in Figure 4.

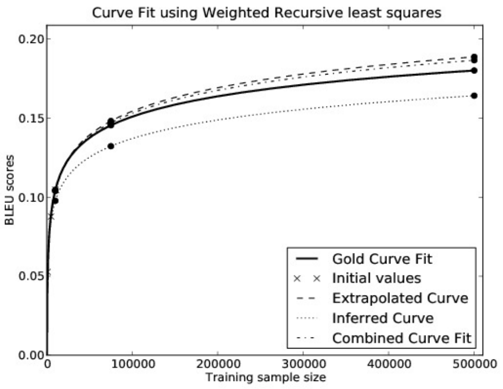

Figure 4: Predicted curves in the three scenarios for Czech-English test set using the Lasso model

8 Conclusion

The ability to predict the amount of parallel data required to achieve a given level of quality is very valuable in planning business deployments of statistical machine translation; yet, we are not aware of any rigorous proposal for addressing this need.

Here, we proposed methods that can be directly applied to predicting learning curves in realistic scenarios. We identified a suitable parametric family for modeling learning curves via an extensive empirical comparison. We described an *inference* method that requires a minimal initial investment in the form of only a small parallel *test* dataset. For the cases where a slightly larger in-domain "seed" parallel corpus is available, we introduced an *extrapolation* method and a *combined* method yielding high-precision predictions: using models trained on up to 20K sentence pairs we can predict performance on a given test set with a root mean squared error in the order of 1 BLEU point at 75K sentence pairs, and in the order of 2-4 BLEU points at 500K. Considering that variations in the order of 1 BLEU point on a same test dataset can be observed simply due to the instability of the standard MERT parameter tuning algorithm (Foster and Kuhn, 2009; Clark et al., 2011), we believe our results to be close to what can be achieved in principle. Note that by using gold curves as labels instead of actual measures we implicitly average across many rounds of MERT (14 for each curve), greatly attenuating the impact of the instability in the optimization procedure due to randomness.

For enabling this work we trained a multitude of instances of the same phrase-based SMT system on 30 distinct combinations of language-pair and domain, each with fourteen distinct training sets of increasing size and tested these instances on multiple in-domain datasets, generating 96 learning curves. BLEU measurements for all 96 learning curves along with the gold curves and feature values used for inferring the learning curves are available as additional material to this submission.

We believe that it should be possible to use insights from this paper in an active learning setting, to select, from an available monolingual source, a subset of a given size for manual translation, in such a way at to yield the highest performance, and we plan to extend our work in this direction.

References

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting Success in Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754, Honolulu, Hawaii, October. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

Carroll Croarkin and Paul Tobias. 2006. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH, July. Available online: http://www.itl.nist.gov/div898/handbook/.

George Foster and Roland Kuhn. 2009. Stabilizing Minimum Error Rate Training. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 242–249, Athens, Greece, March. Association for Computational Linguistics.

Baohua Gu, Feifang Hu, and Huan Liu. 2001. Modelling Classification Performance for Large Data Sets. In *Proceedings of the Second International Conference on Advances in Web-Age Information Management*, WAIM '01, pages 317–328, London, UK. Springer-Verlag.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of Human Language Technologies: The 2003 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada, May. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit*, Phuket, Thailand, September.

Jorge J. Moré. 1978. The Levenberg-Marquardt Algorithm: Implementation and Theory. *Numerical Analysis. Proceedings Biennial Conference Dundee 1977*, 630:105–116.

Graham Neubig. 2011. The Kyoto Free Translation Task. http://www.phontron.com/kftt.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Claudia Perlich, Foster J. Provost, and Jeffrey S. Simonoff. 2003. Tree Induction vs. Logistic Regression: A Learning-Curve Analysis. *Journal of Machine Learning Research*, 4:211–255.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth conference on International Language Resources and Evaluation (LREC'12)*, Istanbul, May. European Language Resources Association (ELRA).

Robert Tibshirani. 1994. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288.

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Marco Turchi, Tijl De Bie, and Nello Cristianini. 2008. Learning Performance of a Machine Translation System: a Statistical and Computational Analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 35–43, Columbus, Ohio, June. Association for Computational Linguistics.