Referential Translation Machines for Quality Estimation

Ergun Biçici
Centre for Next Generation Localisation,
Dublin City University, Dublin, Ireland.
ergun.bicici@computing.dcu.ie

Abstract

We introduce referential translation machines (RTM) for quality estimation of translation outputs. RTMs are a computational model for identifying the translation acts between any two data sets with respect to a reference corpus selected in the same domain, which can be used for estimating the quality of translation outputs, judging the semantic similarity between text, and evaluating the quality of student answers. RTMs achieve top performance in automatic, accurate, and language independent prediction of sentence-level and word-level statistical machine translation (SMT) quality. RTMs remove the need to access any SMT system specific information or prior knowledge of the training data or models used when generating the translations. We develop novel techniques for solving all subtasks in the WMT13 quality estimation (QE) task (QET 2013) based on individual RTM models. Our results achieve improvements over last year's QE task results (QET 2012), as well as our previous results, provide new features and techniques for QE, and rank 1st or 2nd in all of the subtasks.

1 Introduction

Quality Estimation Task (QET) (Callison-Burch et al., 2012; Callison-Burch et al., 2013) aims to develop quality indicators for translations and predictors without access to the references. Prediction of translation quality is important because the expected translation performance can help in estimating the effort required for correcting the translations during post-editing by human translators.

Bicici et al. (2013) develop the Machine Translation Performance Predictor (MTPP), a state-of-the-art, language independent, and SMT system extrinsic machine translation performance predictor, which achieves better performance than the competitive QET baseline system (Callison-Burch et al., 2012) by just looking at the test source sentences and becomes the 2nd overall after also looking at the translation outputs in QET 2012.

In this work, we introduce referential translation machines (RTM) for quality estimation of translation outputs, which is a computational model for identifying the acts of translation for translating between any given two data sets with respect to a reference corpus selected in the same domain. RTMs reduce our dependence on any task dependent resource. In particular, we do not use the baseline software or the SMT resources provided with the QET 2013 challenge. We believe having access to glass-box features such as the phrase table or the n-best lists is not realistic especially for use-cases where translations may be provided by different MT vendors (not necessarily from SMT products) or by human translators. Even the prior knowledge of the training corpora used for building the SMT models or any other model used when generating the translations diverges from the goal of independent and unbiased prediction of translation quality. Our results show that we do not need to use any SMT system dependent information to achieve the top performance when predicting translation output quality.

2 Referential Translation Machine (RTM)

Referential translation machines (RTMs) provide a computational model for quality and semantic similarity judgments using retrieval of relevant training data (Biçici and Yuret, 2011a; Biçici, 2011) as interpretants for reaching shared semantics (Biçici, 2008). RTMs achieve very good performance in judging the semantic similarity of sentences (Biçici and van Genabith, 2013a) and we can also use RTMs to automatically assess the

correctness of student answers to obtain better results (Biçici and van Genabith, 2013b) than the state-of-the-art (Dzikovska et al., 2012).

RTM is a computational model for identifying the acts of translation for translating between any given two data sets with respect to a reference corpus selected in the same domain. RTM can be used for predicting the quality of translation outputs. An RTM model is based on the selection of common training data relevant and close to both the training set and the test set of the task where the selected relevant set of instances are called the interpretants. Interpretants allow shared semantics to be possible by behaving as a reference point for similarity judgments and providing the context. In semiotics, an interpretant I interprets the signs used to refer to the real objects (Biçici, 2008). RTMs provide a model for computational semantics using interpretants as a reference according to which semantic judgments with translation acts are made. Each RTM model is a data translation model between the instances in the training set and the test set. We use the FDA (Feature Decay Algorithms) instance selection model for selecting the interpretants (Biçici and Yuret, 2011a) from a given corpus, which can be monolingual when modeling paraphrasing acts, in which case the MTPP model (Section 2.1) is built using the interpretants themselves as both the source and the target side of the parallel corpus. RTMs map the training and test data to a space where translation acts can be identified. We view that acts of translation are ubiquitously used during communication:

Every act of communication is an act of translation (Bliss, 2012).

Translation need not be between different languages and paraphrasing or communication also contain acts of translation. When creating sentences, we use our background knowledge and translate information content according to the current context. Given a training set `train`, a test set `test`, and some monolingual corpus $\mathcal{C}$, preferably in the same domain as the training and test sets, the RTM steps are:

1. $T = \texttt{train} \cup \texttt{test}$.
2. select$(T, \mathcal{C}) \rightarrow \mathcal{I}$
3. MTPP$(\mathcal{I}, \texttt{train}) \rightarrow \mathcal{F}_{\texttt{train}}$
4. MTPP$(\mathcal{I}, \texttt{test}) \rightarrow \mathcal{F}_{\texttt{test}}$
5. learn$(M, \mathcal{F}_{\texttt{train}}) \rightarrow \mathcal{M}$
6. predict$(\mathcal{M}, \mathcal{F}_{\texttt{test}}) \rightarrow \hat{q}$

Step 2 selects the interpretants, $\mathcal{I}$, relevant to the instances in the combined training and test data. Steps 3 and 4 use $\mathcal{I}$ to map `train` and `test` to a new space where similarities between translation acts can be derived more easily. Step 5 trains a learning model M over the training features, $\mathcal{F}_{\texttt{train}}$, and Step 6 obtains the predictions. RTM relies on the representativeness of $\mathcal{I}$ as a medium for building translation models for translating between `train` and `test`.

Our encouraging results in the QET challenge provides a greater understanding of the acts of translation we ubiquitously use when communicating and how they can be used to predict the performance of translation, judging the semantic similarity between text, and evaluating the quality of student answers. RTM and MTPP models are not data or language specific and their modeling power and good performance are applicable across different domains and tasks. RTM expands the applicability of MTPP by making it feasible when making monolingual quality and similarity judgments and it enhances the computational scalability by building models over smaller but more relevant training data as interpretants.

2.1 The Machine Translation Performance Predictor (MTPP)

In machine translation (MT), pairs of source and target sentences are used for training statistical MT (SMT) models. 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. MTPP (Biçici et al., 2013) is a state-of-the-art and top performing machine translation performance predictor, which uses machine learning models over features measuring how well the test set matches the training set to predict the quality of a translation without using a reference translation. MTPP measures the coverage of individual test sentence features and syntactic structures found in the training set and derives feature functions measuring the closeness of test sentences to the available training data, the difficulty of translating the sentence, and the presence of acts of translation for data transformation.

2.2 MTPP Features for Translation Acts

MTPP uses n-gram features defined over text or common cover link (CCL) (Seginer, 2007) structures as the basic units of information over which similarity calculations are made. Unsupervised

parsing with CCL extracts links from base words to head words, resulting in structures representing the grammatical information instantiated in the training and test data. Feature functions use statistics involving the training set and the test sentences to determine their closeness. Since they are language independent, MTPP allows quality estimation to be performed extrinsically.

We extend MTPP (Biçici et al., 2013) in its learning module, the features included, and their representations. Categories for the 308 features (S for source, T for target) used are listed below where the number of features are given in {#} and the detailed descriptions for some of the features are presented in (Biçici et al., 2013).

- *Coverage {110}:* Measures the degree to which the test features are found in the training set for both S ({56}) and T ({54}).
- *Synthetic Translation Performance {6}:* Calculates translation scores achievable according to the n-gram coverage.
- *Length {7}:* Calculates the number of words and characters for S and T and their average token lengths and their ratios.
- *Feature Vector Similarity {16}:* Calculates similarities between vector representations.
- *Perplexity {90}:* Measures the fluency of the sentences according to language models (LM). We use both forward ({30}) and backward ({15}) LM features for S and T.
- *Entropy {9}:* Calculates the distributional similarity of test sentences to the training set over top N retrieved sentences.
- *Retrieval Closeness {24}:* Measures the degree to which sentences close to the test set are found in the selected training set, $\mathcal{I}$, using FDA (Biçici and Yuret, 2011a).
- *Diversity {6}:* Measures the diversity of co-occurring features in the training set.
- *IBM1 Translation Probability {16}:* Calculates the translation probability of test sentences using the selected training set, $\mathcal{I}$, (Brown et al., 1993).
- *IBM2 Alignment Features {11}:* Calculates the sum of the entropy of the distribution of alignment probabilities for S ($\sum_{s \in S} -p \log p$ for $p = p(t|s)$ where s and t are tokens) and T, their average for S and T, the number of entries with $p \geq 0.2$ and $p \geq 0.01$, the entropy of the word alignment between S and T and its average, and word alignment log probability and its value in terms of bits per word.
- *Minimum Bayes Retrieval Risk {4}:* Calculates the translation probability for the translation having the minimum Bayes risk among the retrieved training instances.
- *Sentence Translation Performance {3}:* Calculates translation scores obtained according to $q(T, R)$ using BLEU (Papineni et al., 2002), NIST (Doddington, 2002), or F_1 (Biçici and Yuret, 2011b) for q.
- *Character n-grams {4}:* Calculates cosine between character n-grams (for n=2,3,4,5) obtained for S and T (Bär et al., 2012).
- *LIX {2}:* Calculates the LIX readability score (Wikipedia, 2013; Björnsson, 1968) for S and T. [1]

For retrieval closeness, we use FDA instead of dice for sentence selection. We also improve FDA's instance selection score by scaling with the length of the sentence (Biçici and Yuret, 2011a). IBM2 alignments and their probabilities are obtained by first obtaining IBM1 alignments and probabilities, which become the starting point for the IBM2 model. Both models are trained for 25 to 75 iterations or until convergence.

3 Quality Estimation Task Results

We participate in all of the four challenges of the quality estimation task (QET) (Callison-Burch et al., 2013), which include English to Spanish (en-es) and German to English translation directions. There are two main categories of challenges: sentence-level prediction (Task 1.*) and word-level prediction (Task 2). Task 1.1 is about predicting post-editing effort (PEE), Task 1.2 is about ranking translations from different systems, Task 1.3 is about predicting post-editing time (PET), and Task 2 is about binary or multi-class classification of word-level quality.

For each task, we develop RTM models using the parallel corpora and the LM corpora distributed by the translation task (WMT13) (Callison-Burch et al., 2013) and the LM corpora provided by LDC for English and Spanish [2]. The parallel corpora contain 4.3M sentences for de-en with 106M words for de and 111M words for en and 15M sentences for en-es with 406M words for en and 455M words for

[1] LIX=$\frac{A}{B} + C\frac{100}{A}$, where A is the number of words, C is words longer than 6 characters, B is words that start or end with any of ".", ":", "!", "?" similar to (Hagström, 2012).

[2] English Gigaword 5th, Spanish Gigaword 3rd edition.

es. We do not use any resources provided by QET including data, software, or baseline features since they are SMT system dependent or language specific. Instance selection for the training set and the language model (LM) corpus is handled by a parallel implementation of FDA (Biçici, 2013). LM are trained using SRILM (Stolcke, 2002). We tokenize and true-case all of the corpora. The true-caser is trained on all of the training corpus using Moses (Koehn et al., 2007). We prepare the corpora by following this procedure: tokenize $\rightarrow$ train the true-caser $\rightarrow$ true-case. Table 1 lists the statistics of the data used in the training and test sets for the tasks.

	Task	1.1	1.2 (de-en)	1.2 (en-es)	1.3 & 2
Train	sents	2254	32730	22338	803
	words	63K (en)	762K (de)	528K (en)	18K (en)
		67K (es)	786K (en)	559K (es)	20K (es)
Test	sents	500	1810	1315	284

Table 1: Data statistics for different tasks. The number of words is listed after tokenization.

Since we do not know the best training set size that will maximize the performance, we rely on previous SMT experiments (Biçici and Yuret, 2011a; Biçici and Yuret, 2011b) and quality estimation challenges (Biçici and van Genabith, 2013a; Biçici and van Genabith, 2013b) to select the proper training set size. For each training and test sentence provided in each subtask, we choose between 65 and 600 sentences from the parallel training corpora to be added to the training set, which creates roughly 400K sentences for training. We add the selected training set to the 8 million sentences selected for each LM corpus. The statistics of the training data selected by the parallel FDA and used as interpretants in the RTM models is given in Table 2.

Task	1.1	1.2 (de-en)	1.2 (en-es)	1.3	2
sents	406K	318K	299K	398K	397K
words	6.3M (en)	4.8M (de)	4.3M (en)	6.6M (en)	6.6M (en)
	6.9M (es)	4.9M (en)	4.6M (es)	7.2M (es)	7.2M (es)

Table 2: Statistics of the training data used as interpretants in the RTM models in thousands (K) of sentences or millions (M) of words.

3.1 Evaluation

In this section, we describe the metrics we use to evaluate the learning performance. 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 (MAE):* $|\bar{\epsilon}| = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$
- *Relative Absolute Error (RAE) :* $\overrightarrow{|\epsilon|} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |\bar{y} - y_i|}$
- *Root Mean Squared Error:* RMSE $= \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}$
- *DeltaAvg:* $\bar{\Delta}(V, S) = \frac{1}{|S|/2 - 1} \sum_{n=2}^{|S|/2} \left(\sum_{k=1}^{n-1} \frac{\sum_{s \in \bigcup_{i=1}^{k} q_i} V(s)}{|\bigcup_{i=1}^{k} q_i|} \right)$
- *Correlation:* $r = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (\hat{y}_i - \bar{\hat{y}})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}}$

DeltaAvg (Callison-Burch et al., 2012) calculates the average quality difference between the scores for the top $n-1$ quartiles and the overall quality for the test set. Relative absolute error measures the error relative to the error when predicting the actual mean. We use the coefficient of determination, $R^2 = 1 - \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{n} (\bar{y} - y_i)^2$, during optimization where the models are regression based and higher R^2 values are better.

3.2 Task 1: Sentence-level Prediction of Quality

In this subsection, we develop techniques for the prediction of quality at the sentence-level. We first discuss the learning models we use and how we optimize them and then provide the results for the individual subtasks and the settings used.

3.2.1 Learning Models and Optimization

The learning models we use for predicting the translation quality include the ridge regression (RR) and support vector regression (SVR) with RBF 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 HTER score, the F_1 score (Biçici and Yuret, 2011b), or the PET score. We also use these learning models after a feature subset selection with recursive feature elimination (RFE) (Guyon et al., 2002) or a dimensionality reduction and mapping step using partial least squares (PLS) (Specia et al., 2009), both of which are described in (Biçici et al., 2013). 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 optimize the learning parameters, the number of features to select, and the number of dimensions used for PLS. More detailed description of the optimization process is given in (Biçici et al., 2013). Our submissions use the results from SVR and SVR after PLS (SVRPLS) since they perform the best during training.

Optimization can be a challenge for SVR due to the large number of parameter settings to search. In this work, we decrease the search space by selecting ε close to the theoretically optimal values. We select ε close to the standard deviation of the noise in the training set since the optimal value for ε is shown to have linear dependence to the noise level for different noise models (Smola et al., 1998). We use RMSE of RR on the training set as an estimate for the noise level (σ of noise) and the following formulas to obtain the ε with $\tau = 3$:

$$\varepsilon = \tau\sigma\sqrt{\frac{\ln n}{n}} \quad (1)$$

and the C (Cherkassky and Ma, 2004; Chalimourda et al., 2004):

$$C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \quad (2)$$

Since the C obtained could be low (Chalimourda et al., 2004), we use a range of C values in addition to the obtained C value including C values with a couple of σ_y values larger.

Table 3 lists the RMSE of the RR model on the training set and the corresponding ε and C values for different subtasks. We also present the optimized parameter values for SVR and SVRPLS. Table 3 shows that, empirically, Equation 1 and Equation 2 gives results close to the best parameters found after optimization.

Task	1.1	1.2 (de-en)	1.2 (en-es)	1.3
RMSE RR	.1397	.1169	.1569	68.06
ε	.0245	.0062	.01	18.64
C	.8398	.8713	1.02	371.28
$\hat{C}$ (SVR)	.8398	.5	.5	100
γ (SVR)	.0005	.001	.0001	.0005
$\hat{C}$ (SVRPLS)	1.5	.8713	1.02	100
γ (SVRPLS)	.0001	.0001	.0001	.001
# dim (SVRPLS)	60	60	60	60

Table 3: Optimal parameters predicted by Equation 1 and Equation 2 and the optimized parameter values, $\hat{C}$ and γ for SVR and SVRPLS and the number of dimensions (# dim) for SVRPLS.

3.2.2 Task 1.1: Scoring and Ranking for Post-Editing Effort

Task 1.1 involves the prediction of the case insensitive translation edit rate (TER) scores obtained by TERp (Snover et al., 2009) and their ranking. In contrast, we derive features over sentences that are true-cased. We obtain the rankings by sorting according to the predicted TER scores.

Task 1.1	R^2	r	RMSE	MAE	RAE
RR	0.3510	0.5965	0.1393	0.1086	0.7888
RR PLS	0.4232	0.6509	0.1313	0.1023	0.7430
SVR	0.4394	0.6647	0.1295	0.0967	0.7023
SVR PLS	0.4305	0.6569	0.1305	0.1003	0.7284

Table 4: Task1.1 results on the training set.

Table 4 presents the learning performance on the training set using the optimized parameters. We are able to significantly improve the results when compared with the QET 2012 (Callison-Burch et al., 2012) and our previous results (Biçici et al., 2013) especially in terms of MAE and RAE.

The results on the test set are given in Table 5. Rank lists the overall ranking in the task. RTMs with SVR PLS learning is able to achieve the top rank in this task.

Ranking	DeltaAvg	r	Rank
CNGL SVRPLS	11.09	0.55	1
CNGL SVR	9.88	0.51	4
Scoring	MAE	RMSE	Rank
CNGL SVRPLS	13.26	16.82	3
CNGL SVR	13.85	17.28	8

Table 5: Task1.1 results on the test set.

3.2.3 Task 1.2: Ranking Translations from Different Systems

Task 1.2 involves the prediction of the ranking among up to 5 translation outputs produced by different MT systems. Evaluation is done against the human rankings using the Kendall's τ correlation (Callison-Burch et al., 2013): $\tau = (c - d)/\frac{n(n-1)}{2} = \frac{c-d}{c+d}$ where a pair is concordant, c, if the ordering agrees, discordant, d, if their ordering disagrees, and neither concordant nor discordant if their rankings are equal.

We use sentence-level F_1 scores (Biçici and Yuret, 2011b; Biçici, 2011) as the target to predict. We use F_1 because it can be easily interpreted and it correlates well with human judgments (more than TER) (Biçici and Yuret, 2011b; Callison-Burch et al., 2011). We also found that the τ of the rankings obtained according to the F_1 score over

the training set (0.2040) is better than BLEU (Papineni et al., 2002) (0.1780) and NIST (Doddington, 2002) (0.1907) for de-en. Table 6 presents the learning performance on the training set using the optimized parameters. Learning F_1 becomes an easier task than learning TER as observed from the results but we have significantly more training instances. We use the SVR model for predicting the F_1 scores on the training set and the test set. MAE is a more important performance metric here since we want to be as precise as possible when predicting the actual performance.

	Task 1.2	R^2	r	RMSE	MAE	RAE
de-en	RR	0.6320	0.7953	0.1169	0.0733	0.5535
	SVR	0.7528	0.8692	0.0958	0.0463	0.3494
en-es	RR	0.5101	0.7146	0.1569	0.1047	0.6323
	SVR	0.4819	0.7018	0.1613	0.0973	0.5873

Table 6: Task1.2 results on the training set.

Our next goal is to learn a threshold for judging if two translations are equal over the predicted F_1 scores. This threshold is used to determine whether we need to alter the ranking. We try to mimic the human decision process when determining whether two translations are equivalent. On some occasions where the sentences are close enough, humans give them equal ranking. This is also related to the granularity of the differences visible with a 1 to 5 ranking schema.

We compared different threshold formulations and used the following condition in our submissions to decide whether the ranking of item i in a set S of translations, $i \in S$, should be different:

$$\sum_{j \neq i} \frac{F_1(j) - F_1(i)}{|j - i|} / |S| > t, \quad (3)$$

where t is the optimized threshold minimizing the following loss for n training instances:

$$\sum_{i=1}^{n} \tau(f(t, q_i), r_i) \quad (4)$$

where $f(t, q_i)$ is a function returning rankings based on the threshold t and the quality scores for instance i, q_i and $\tau(r_j, r_i)$ calculates the τ score based on the rankings r_j and r_i.

For both de-en and en-es subtasks, we found the thresholds obtained to be very similar or the same. The optimized values are given in Table 7. On the test set, we used the same threshold, $t = 0.00275$ for both de-en and en-es, which is a little higher than the optimal t to prevent overfitting.

Task 1.2	τ	t	# same	# all
de-en	.2339	.00013	236	25644
	.2287	.00275	494	
en-es	.2801	.00073	136	17752
	.2764	.00275	233	

Table 7: Task1.2 optimized thresholds and the corresponding comparisons that were found to be equal (# same) over all comparisons (# all).

We believe that human judgments of linguistic equality and the corresponding thresholds we learned in this work can be useful for developing better automatic evaluation metrics and can improve the correlation of the scores obtained with human judgments (as we did here). The results on the test set are given in Table 8. We are also able to achieve the top ranking in this task.

Ties penalized	model	τ	Rank
de-en	CNGL SVRPLS F_1	0.17	3
	CNGL SVR F_1	0.17	4
en-es	CNGL SVRPLS F_1	0.15	1
	CNGL SVR F_1	0.13	2
Ties ignored	model	τ	Rank
de-en	CNGL SVRPLS F_1	0.17	3
	CNGL SVR F_1	0.17	4
en-es	CNGL SVRPLS F_1	0.16	2
	CNGL SVR F_1	0.13	3

Table 8: Task1.2 results on the test set.

3.2.4 Task 1.3: Predicting Post-Editing Time

Task 1.3 involves the prediction of the post-editing time (PET) for a translator to post-edit the MT output. Table 9 presents the learning performance on the training set using the optimized parameters.

Task 1.3	R^2	r	RMSE	MAE	RAE
RR	0.4463	0.6702	68.0628	39.5250	0.6694
RR PLS	0.5917	0.7716	58.4464	35.8759	0.6076
SVR	0.4062	0.6753	70.4853	36.5132	0.6184
SVR PLS	0.5316	0.7604	62.6031	33.5490	0.5682

Table 9: Task1.3 results on the training set.

The results on the test set are given in Table 10. We are able to become the 2nd best system according to MAE in this task.

3.3 Task 2: Word-level Prediction of Quality

In this subsection, we develop a learning model, global linear models with dynamic learning rate (GLMd), for the prediction of quality at the word-level where the word-level quality is a binary (K: keep, C: change) or multi-class classification problem (K: keep, S: substitute, D: delete). We first discuss the GLMd learning model, then we present

Task 1.3	MAE	Rank
CNGL SVR	49.2121	3
CNGL SVRPLS	49.6161	4
	RMSE	Rank
CNGL SVRPLS	86.6175	4
CNGL SVR	90.3650	7

Table 10: Task1.3 results on the test set.

the word-level features we use, and then present our results on the test set.

3.3.1 Global Linear Models with Dynamic Learning (GLMd)

Collins (2002) develops global learning models (GLM), which rely on Viterbi decoding, perceptron learning, and flexible feature definitions. We extend the GLM framework by parallel perceptron training (McDonald et al., 2010) and dynamic learning with adaptive weight updates in the perceptron learning algorithm:

$$\mathbf{w} = \mathbf{w} + \alpha \left(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}}) \right), \qquad (5)$$

where Φ returns a global representation for instance i and the weights are updated by $\alpha = \exp(\log_{10}(3\epsilon_{-1}/\epsilon_0))$ with ϵ_{-1} and ϵ_0 representing the error of the previous and first iteration respectively. α decays the amount of the change during weight updates at later stages and prevents large fluctuations with updates. We used both the GLM model and the GMLd models in our submissions.

3.3.2 Word-level Features

We introduce a number of novel features for the prediction of word-level translation quality. In broad categories, these word-level features are:

- *CCL:* Uses CCL links.
- *Word context:* Surrounding words.
- *Word alignments:* Alignments, their probabilities, source and target word contexts.
- *Length:* Word lengths, n-grams over them.
- *Location:* Location of the words.
- *Prefix and Suffix:* Word prefixes, suffixes.
- *Form:* Capital, contains digit or punctuation.

We found that CCL links are the most discriminative feature among these. In total, we used 511K features for binary and 637K for multi-class classification. The learning curve is given in Figure 1.

The results on the test set are given in Table 11. P, R, and A stand for precision, recall, and accuracy respectively. We are able to become the 2nd according to A in this task.

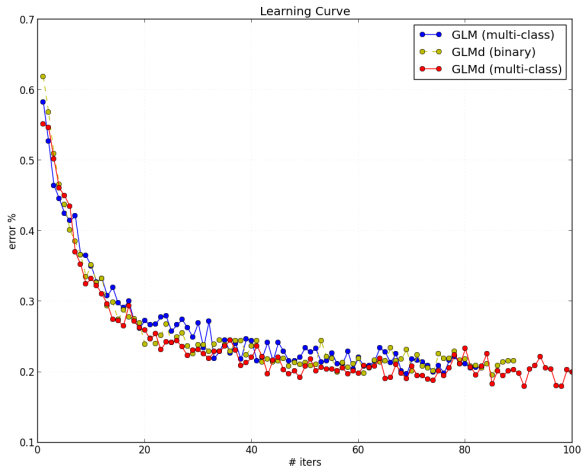

Figure 1: Learning curve with the parallel GLM and GLMd models.

Binary	A	P	R	F_1	Rank (A)
CNGL GLMd	.7146	.7392	.9261	.8222	2
CNGL GLM	.7010	.7554	.8581	.8035	5
Multi-class	A	Rank			
CNGL GLMd	.7162	3			
CNGL GLM	.7116	4			

Table 11: Task 2 results on the test set.

4 Contributions

Referential translation machines achieve top performance in automatic, accurate, and language independent prediction of sentence-level and word-level statistical machine translation (SMT) quality. RTMs remove the need to access any SMT system specific information or prior knowledge of the training data or models used when generating the translations. We develop novel techniques for solving all subtasks in the quality estimation (QE) task (QET 2013) based on individual RTM models. Our results achieve improvements over last year's QE task results (QET 2012), as well as our previous results, provide new features and techniques for QE, and rank 1st or 2nd in all of the subtasks.

Acknowledgments

This work is 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). We also thank the SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational facilities and support.

References

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 435–440, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Ergun Biçici and Josef van Genabith. 2013a. CNGL-CORE: Referential translation machines for measuring semantic similarity. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*, Atlanta, Georgia, USA, 13-14 June. Association for Computational Linguistics.

Ergun Biçici and Josef van Genabith. 2013b. CNGL: Grading student answers by acts of translation. In **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics and Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, 14-15 June. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011a. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici and Deniz Yuret. 2011b. RegMT system for machine translation, system combination, and evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 323–329, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*.

Ergun Biçici. 2011. *The Regression Model of Machine Translation*. Ph.D. thesis, Koç University. Supervisor: Deniz Yuret.

Ergun Biçici. 2013. Feature decay algorithms for fast deployment of accurate statistical machine translation systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ergun Biçici. 2008. Consensus ontologies in socially interacting multiagent systems. *Journal of Multiagent and Grid Systems*.

Carl Hugo Björnsson. 1968. *Läsbarhet*. Liber.

Chris Bliss. 2012. Comedy is translation, February. http://www.ted.com/talks/chris_bliss_comedy_is_translation.html.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omer F. Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, England, July. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 workshop on statistical machine translation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 10–51. Association for Computational Linguistics, August.

Athanassia Chalimourda, Bernhard Schölkopf, and Alex J. Smola. 2004. Experimentally optimal ν in support vector regression for different noise models and parameter settings. *Neural Networks*, 17(1):127–141, January.

Vladimir Cherkassky and Yunqian Ma. 2004. Practical selection of svm parameters and noise estimation for svm regression. *Neural Netw.*, 17(1):113–126, January.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. 2012. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, Montréal, Canada, June. Association for Computational Linguistics.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

Kenth Hagström. 2012. Swedish readability calculator. https://github.com/keha76/Swedish-Readability-Calculator.

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 *ACL*, pages 177–180, Prague, Czech Republic, June.

Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464, Los Angeles, California, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Yoav Seginer. 2007. *Learning Syntactic Structure*. Ph.D. thesis, Universiteit van Amsterdam.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August.

A. J. Smola, N. Murata, B. Schölkopf, and K.-R. Müller. 1998. Asymptotically optimal choice of ε-loss for support vector machines. In L. Niklasson, M. Boden, and T. Ziemke, editors, *Proceedings of the International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, pages 105–110, Berlin. Springer.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lucia Specia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 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)*, pages 28–35, Barcelona, May. EAMT.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, pages 901–904.

Wikipedia. 2013. Lix. http://en.wikipedia.org/wiki/LIX.