PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 103 APRIL 2015 43–64

QuEst for High Quality Machine Translation

Ergun Biçici[a], Lucia Specia[b]

[a] ADAPT CNGL Centre for Global Intelligent Content
School of Computing, Dublin City University
[b] Department of Computer Science
University of Sheffield

Abstract

In this paper we describe the use of QuEst, a framework that aims to obtain predictions on the quality of translations, to improve the performance of machine translation (MT) systems without changing their internal functioning. We apply QuEst to experiments with:

i. multiple system translation ranking, where translations produced by different MT systems are ranked according to their estimated quality, leading to gains of up to 2.72 BLEU, 3.66 BLEUs, and 2.17 F_1 points;

ii. n-best list re-ranking, where n-best list translations produced by an MT system are reranked based on predicted quality scores to get the best translation ranked top, which lead to improvements on sentence NIST score by 0.41 points;

iii. n-best list combination, where segments from an n-best list are combined using a lattice-based re-scoring approach that minimize word error, obtaining gains of 0.28 BLEU points; and

iv. the ITERPE strategy, which attempts to identify translation errors regardless of prediction errors (ITERPE) and build sentence-specific SMT systems (SSSS) on the ITERPE sorted instances identified as having more potential for improvement, achieving gains of up to 1.43 BLEU, 0.54 F_1, 2.9 NIST, 0.64 sentence BLEU, and 4.7 sentence NIST points in English to German over the top 100 ITERPE sorted instances.

1. Introduction

QuEst is a quality estimation framework that offers a wide range of feature extractors that can be used to describe source and translations texts in order to build

Corresponding author: ergun.bicici@computing.dcu.ie

Cite as: Ergun Biçici, Lucia Specia. QuEst for High Quality Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 103, 2015, pp. 43–64. doi: 10.1515/pralin-2015-0003.

and apply models to predict the quality of translations. It was developed within QT-LaunchPad (Preparation and Launch of a Large-Scale Action for Quality Translation Technology),[1] a project aimed at high quality machine translation through, among other things, the use of novel metrics to systematically measure and estimate translation quality.

We use QuEst to predict and improve the quality of MT systems without changing their internal functioning and evaluate with automatic evaluation methods. In what follows we describe the experimental settings (Section 2) and results of several experiments focusing on four approaches: (i) multiple system translation ranking (Section 3), (ii) n-best list re-ranking and (iii) n-best list combination (Section 4), and (iv) ITERPE to identify translations with potential for improvement and build sentence-specific SMT systems (Section 5). SMT performance improvements according to all these approaches are summarized in Table 16.

2. Experimental settings

2.1. Datasets

The multiple MT system translation ranking experiments in Section 3 use the following datasets where multiple machine translations are available for each source sentence:

DEAMT09 English to Spanish translations by four SMT systems, denoted by s_1-s_4, scored for post-editing effort (PEE) [2] in 1-4 (highest-lowest) in absolute terms (Specia et al., 2009). $3,095$ sentences are used for training and 906 for testing.

DQET13-HTER English to Spanish translations scored for HTER with $2,254$ sentences for training and 500 for testing (Task 1.1 dataset used in quality estimation task (QET13) at WMT13 (Bojar et al., 2013)).

DQET13-rank(de-en) German to English set of up to five alternative translations produced by different MT systems human ranked relative to each other according to their quality. $7,098$ source sentences and $32,922$ translations are used for training and 365 source sentences and $1,810$ translations for testing (Task 1.2 dataset in QET13).

DQET13-rank(en-es) English to Spanish DQET13-rank dataset with $4,592$ source sentences and $22,447$ translations for training and 264 source sentences and $1,315$ translations for testing.

The re-ranking and combination experiments in Section 4 use the following datasets:

DQET13-nbest English to Spanish n-best lists provided in Task 1.1 of QET13.

[1] http://www.qt21.eu

[2] as perceived by the post-editors

DFDA13-nbest English to Spanish and Spanish to English distinct 1000-best lists from Moses (Koehn et al., 2007) SMT systems developed for the WMT13 translation task using FDA5 (Biçici, 2013a; Biçici and Yuret, 2015), which is developed for efficient parameterization, optimization, and implementation of state-of-the-art instance selection model feature decay algorithms (FDA) (Biçici, 2011; Biçici and Yuret, 2015). FDA try to increase the diversity of the training set by decaying the weights of n-gram features from the test set.

The ITERPE and SSSS experiments in Section 5 use the following datasets:

DFDA14-train English to German and German to English translations of separate $3,000$ sentences randomly selected from the development sentences available at WMT14 that are unused when training the Parallel FDA5 Moses SMT systems (Biçici et al., 2014) with translations obtained using the Parallel FDA5 Moses SMT systems.

DFDA14-test English to German with $2,737$ sentences and German to English with $3,003$ sentences WMT14 translation task test set with baseline translations obtained with the Parallel FDA5 Moses SMT systems developed for the WMT14 translation task (Biçici et al., 2014).

ParFDA5 WMT14 dataset available at `https://github.com/bicici/ParFDA5WMT` provides training data for building Parallel FDA5 Moses SMT systems used. Other datasets used for the experiments, as well as the QuEst open source QE toolkit, are available for download at `http://www.quest.dcs.shef.ac.uk/`.

2.2. Evaluation metrics

We evaluate the learning performance with root mean squared error (RMSE), mean absolute error (MAE), relative absolute error (RAE), MAE relative (MAER), mean RAE relative (MRAER). RAE measures the absolute error relative to the absolute error of the mean target value, where y_i represents the actual target value for instance i, $\bar{y}$ the mean of all these instances, and $\hat{y_i}$ a prediction for y_i:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |\hat{y_i} - y_i|}{n} \quad \text{RAE} = \frac{\sum_{i=1}^{n} |\hat{y_i} - y_i|}{\sum_{i=1}^{n} |\bar{y} - y_i|} \tag{1}$$

We define MAER and MRAER for easier replication and comparability with relative errors for each instance:

$$\text{MAER}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\sum_{i=1}^{n} \frac{|\hat{y_i} - y_i|}{\lfloor |y_i| \rfloor_\epsilon}}{n} \quad \text{MRAER}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\sum_{i=1}^{n} \frac{|\hat{y_i} - y_i|}{\lfloor |\bar{y} - y_i| \rfloor_\epsilon}}{n} \tag{2}$$

MAER is the mean absolute error relative to the magnitude of the target and MRAER is the mean relative absolute error relative to the absolute error of a predictor always predicting the target mean assuming that target mean is known. MAER and MRAER are capped from below[3] with $\epsilon = \text{MAE}(\hat{\mathbf{y}}, \mathbf{y})/2$, which is the measurement error and it is estimated as half of the mean absolute error or deviation of the predictions from target mean. ϵ represents half of the score step with which a decision about a change in measurement's value can be made. ϵ is similar to half of the standard deviation, σ, of the data but over absolute differences. For discrete target scores, $\epsilon = \frac{\textit{step size}}{2}$. A method for learning decision thresholds for mimicking the human decision process when determining whether two translations are equivalent is described in (Biçici, 2013b).

Additionally, acc (accuracy) represents the percentage of source sentences for which the first-ranked translation by the ranker model agree with humans. For correlation with human judgments, we use Kendall's τ (Bojar et al., 2013). Translation performance is evaluated using BLEU (Papineni et al., 2002), F_1 (Biçici and Yuret, 2011; Biçici, 2011), 1-WER (WER for word error rate), and averaged sentence-level scores BLEUs [4] and NISTs (sentence NIST (Doddington, 2002)). F_1 has been shown to correlate with human judgments better than TER (Biçici and Yuret, 2011; Callison-Burch et al., 2011). Predicting F_1 also allowed us to achieve top results in DQET13-rank (Biçici, 2013b).

2.3. Algorithms

We use Support Vector Regression (SVR) (Smola and Schölkopf, 2004) as the learning algorithm and also use Partial Least Squares (PLS) or feature selection (FS). Feature selection is based on recursive feature elimination (RFE) (Guyon et al., 2002; Biçici, 2013b). We use `scikit-learn`[5] implementation. Some of the results may be rounded with `round(.)` function from `python`[6] and some with `numpy.round()` function from `numpy`[7], which may cause differences at the least significant digit[8].

2.4. QuEst Quality Estimation Features

QuEst offers a number of MT system- and language-independent features, of which we explore two sets:

[3] We use $\lfloor \, . \, \rfloor_\epsilon$ to cap the argument from below to ϵ.

[4] If an n-gram match is not found, the match count is set to $1/2|T'|$ where $|T'|$ is the length of the translation.

[5] `http://scikit-learn.org/`

[6] `https://www.python.org/`

[7] `http://www.scipy.org/`

[8] For instance, `round`$(0.8445, 3) = 0.845$ and `numpy.round`$(0.8445, 3) = 0.84399999999999997$.

IR		Source	Translation	Source and Translation
	Retrieval	15	15	0
Readability	LIX	1	1	0
	Word	1	1	1

Table 1. Counts of features in IR set.

- **BL**: 17 baseline features. These include sentence and average token lengths, number of punctuation symbols, LM probability, average number of translations per source word, and percentage of low or high frequency words.[9]
- **IR**: 35 information retrieval and readability features. Information retrieval features measure the closeness of the test source sentences and their translations to the parallel training data available indexed with Lucene (The Apache Software Foundation, 2014) to predict the difficulty of translating each sentence or finding their translations (Biçici et al., 2013; Biçici, 2013b). For the top five retrieved instances, retrieval scores, BLEU, and F_1 scores over the source sentence or its translation are computed quantifying the closeness of the instances we can find or their similarity to the source sentence or their translation. Readability features attempt to capture the difficulty of translating a sentence by computing the LIX readability score (Björnsson, 1968; Wikipedia, 2013) for source and target sentences, the average number of characters in source and target words, and their ratios. Table 1 shows the number of features in IR categorized according to information source.

The combined feature set, BL+IR, contains 52 features.

For experiments in Section 4, we only consider IR on the translations, since the source sentence is the same for all translation candidates. These results in 18 features, derived for each translation, using retrieval scores, BLEU, and F_1 scores over the top five instances retrieved and three LIX readability features. In that case, the combined feature set, BL+IR, contains 35 features. In those experiments, we also use Moses SMT model-based features, which are obtained from the n-best lists generated, adding 15 more features (6 for lexical reordering, 1 for distortion, 1 for language model, 1 for word penalty, 1 for phrase penalty, 4 for the translation model, and 1 for the overall translation score). This feature set is referred to as SMT. IR+SMT contains 33 features, while BL+IR+SMT contains 50 features.

3. Multiple System Translation Ranking

In multiple MT system translation ranking, we rank translations produced by different MT systems according to their estimated quality. System combination by multiple MT system translation ranking can lead to results that are better than the best

[9] http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17.

	DEAMT09			DQET13-HTER		
	BL	IR	BL+IR	BL	IR	BL+IR
target		F_1			F_1	
RMSE	0.1356	0.0868	**0.0847**	0.1754	0.1777	**0.1743**
MAE	0.1012	0.0535	**0.0521**	**0.1173**	0.122	0.1193
RAE	0.8514	0.4499	**0.4383**	**0.9217**	0.9589	0.9377
MAER	0.5077	0.3073	**0.3068**	**0.6771**	0.7356	0.7211
MRAER	0.8066	0.4778	**0.4612**	**0.7737**	0.8045	0.8055
target		PEE			HTER	
RMSE	0.718	0.775	**0.6772**	**0.1771**	0.1888	0.1778
MAE	0.5727	0.6291	**0.5356**	**0.1426**	0.154	0.1431
RAE	0.7045	0.7738	**0.6588**	**0.9512**	1.0268	0.9544
MAER	0.3209	0.3538	**0.2912**	1.0149	1.0865	**0.9967**
MRAER	0.7103	0.7732	**0.6758**	**0.9144**	0.9816	0.9467

Table 2. Prediction results on the DEAMT09 and DQET13-HTER datasets using a single general SVR.

MT system performance (Biçici and Yuret, 2011; Biçici, 2011). Some of the results in this Section are also presented in (Specia et al., 2013). For DEAMT09, we predict F_1 scores and PEE and for DQET13-HTER, F_1 and HTER. Table 2 presents the prediction results on DEAMT09 for translations from all four systems s_1-s_4 and on DQET13-HTER using a single general SVR model, i.e., a model combining translations from all MT systems. RAE decreases with the addition of the IR; F_1 is easier to predict than PEE when IR is included.

Table 3 presents the prediction results on the DEAMT09 datasets using separate SVR models for each translation system. In Section 3.2, we observe that building separate SVR models for each translation system achieves better performance than building a single model over all of the training set available. Table 3 shows that translation system s_4 is the easiest to predict. This is the MT system with the lowest translation performance (Table 4). IR achieve better performance when predicting F_1, but slightly worse performance when predicting PEE scores. Individual models perform better than using a general model during prediction and as we see in Section 3.1, also when ranking alternative translations for the same source sentence.

		s_1			s_2			s_3			s_4		
target		BL	IR	BL+IR	BL	IR	BL+IR	BL	IR	BL+IR	BL	IR	BL+IR
F_1	RMSE	.1478	.0963	.0930	.1383	.0901	.0871	.1336	.0912	.0883	.0905	.0593	**.0585**
	MAE	.1147	.0607	.0584	.1061	.0562	.0540	.1012	.0564	.0545	.0671	.0371	**.0370**
	RAE	.9161	.4850	.4663	.8875	.4704	**.4516**	.9055	.5040	.4870	.9217	.5099	.5081
	MAER	.5330	.3272	.3201	.5136	.3132	.3082	.5222	.3287	.3229	.4887	**.2989**	.3028
	MRAER	.8962	.5074	.4864	.8387	.4774	**.4595**	.8698	.5244	.5066	.9420	.5367	.5344
PEE	RMSE	.6177	.6173	**.5817**	.6764	.675	.6499	.6679	.6538	.6243	.6792	.6026	.5822
	MAE	.4490	.4712	.4428	.5423	.5456	.5215	.5280	.5238	.4976	.3448	.3216	**.3162**
	RAE	.8669	.9099	.855	.7960	.8009	.7655	.7828	.7765	.7377	.6903	.6439	**.6331**
	MAER	.1979	.2094	.1901	.2741	.2706	.2579	.2645	.2601	.2449	**.1556**	.1605	.1621
	MRAER	.5267	.5882	.5735	.7653	.7837	.7581	.7855	.7800	.7553	**.4085**	.4109	.4106

Table 3. Prediction performance of individual SVR models on the DEAMT09 dataset.

3.1. 1-best Translations

In this section the goal is to rank alternative translations based on their predicted quality to select the best MT system for each translation. For comparison, Table 4 shows the individual performance of each MT system and oracle MT system selection results based on true sentence-level scores (PEE, BLEU, and F_1). Oracle selection using PEE (human) scores obtains worse scores than s_1, the top system, which indicates that PEE does not correlate well with BLEU or F_1.

System	BLEUs	BLEU	F_1
s_1	**0.3521**	**0.3795**	**0.3723**
s_2	0.3156	0.3450	0.3361
s_3	0.2905	0.3145	0.3137
s_4	0.1600	0.1910	0.2148
oracle PEE	0.3362	0.3678	0.3574
oracle BLEU	**0.3941**	**0.4132**	0.4014
oracle F_1	0.3932	0.4130	**0.4020**

Table 4. Performance of systems in DEAMT09.

The predicted scores for each alternative translation of a given source sentence are used to rank these alternatives. For the DQET13-HTER dataset we treat relative 5-way rankings as absolute scores in [1, 5] (best-worst). The absolute scores, $[1-4]$ for DEAMT09 and $[1-5]$ for DQET13-HTER, are referred to as PEE scores. We also predict each translation's F_1 score where y is calculated using the reference translations.

Table 5 presents the 1-best translation results for the DEAMT09 dataset obtained by ranking translations from each system based on the predictions produced by a single general SVR model or individual SVR models (Specia et al., 2010), i.e., a model built for each SMT system. In case of ties, we randomly pick among the equally scoring system outputs. As a baseline, we compute acc-best (accuracy of best-system), which is the percentage of source sentences for which the best system on average (s_1) actually provides the best translation. We achieve gains up to 2.72 BLEU, 3.66 BLEUs, and

		General			Individual		
Target	Evaluation	BL	IR	BL+IR	BL	IR	BL+IR
F1	BLEU	0.3621	0.4037	**0.4067**	0.3792	**0.4052**	**0.4052**
PEE	BLEU	0.3500	**0.4003**	0.4001	0.3792	**0.3819**	**0.3819**
F1	F1	0.3499	0.3930	**0.3940**	0.3661	0.3933	**0.3935**
PEE	F1	0.3432	**0.3886**	0.3882	0.3650	**0.3715**	**0.3715**
F1	BLEUs	0.3316	0.3839	**0.3849**	0.3512	0.3848	**0.3851**
PEE	BLEUs	0.3209	**0.3793**	0.3777	0.3503	**0.3585**	**0.3585**
F1	acc	0.6998	0.7296	**0.7351**	**0.8300**	0.7583	0.7660
PEE	acc	0.6623	0.7318	**0.7384**	0.8311	0.8344	**0.8466**
F1	acc-best	0.4724	**0.5000**	0.4989	**0.9724**	0.5828	0.5894
PEE	acc-best	0.3256	**0.5243**	0.5011	**0.9437**	0.9415	**0.9437**

*Table 5. 1-best translation results on DEAMT09 using general or individual SVR models predicting either F_1 or PEE. Top results are in **bold**.*

2.17 F_1 points compared to the top MT system. QuEst is also able to achieve higher accuracy than the previously reported 0.77 (Specia et al., 2010).

3.2. Correlation with Human Judgments

Here we rank the translations according to the predicted scores and evaluate their correlation with the human rankings. Table 6 presents the results for the DEAMT09 dataset using a single general prediction model and individual models for each MT system. The results show that PEE predictions generally correlate better with human judgments than F_1 predictions.

τ	Target	BL	IR	BL+IR
General	F1	0.6732	0.6064	**0.6382**
	PEE	0.6787	0.6124	**0.7034**
Individual	F1	**0.7719**	0.6743	0.6853
	PEE	0.7719	0.7922	**0.8070**

Table 6. Kendall's τ between the predicted ranks and human judgments for DEAMT09.

4. n-Best List Re-ranking and Combination

In this section we describe the use of QuEst to obtain predictions on the quality of translations in n-best lists in order to re-rank these lists to have the best predicted translation ranked first, or combine translations in these lists to generate a new, better translation. Translation quality improvements using re-ranking or combination allow SMT system independent gains. Re-ranking is done at the sentence-level by using quality predictions to rank translations from n-best lists and select the top ones.

	Features	Setting	BLEU	F_1	BLEUs	NISTs	1-WER
		1-best	0.1710	0.1725	0.1334	1.6445	0.1825
Re-ranking	50-best	oracle	0.2033	0.2087	0.1686	1.8168	0.2340
	BL	FS	0.1627	0.1684	0.1222	***1.6486**	0.1069
	IR	FS	**0.1705**	**0.1712**	**0.1323**	1.6414	**0.1739**
	BL+IR		0.1668	0.1696	0.1275	*1.6449	0.1576
	100-best	oracle	0.2105	0.2160	0.1769	1.8479	0.2485
	BL	FS	0.1639	0.1687	0.1233	1.6383	0.0919
	IR	FS	0.1691	0.1692	**0.1309**	1.6368	**0.1714**
	BL+IR	PLS	**0.1696**	**0.1697**	0.1293	**1.6409**	0.1564
Word Combination	250-best list	oracle	0.2196	0.2253	0.1873	1.8816	0.2609
	BL		0.1705	**0.1721**	0.1323	***1.6455**	*0.1850
	IR	FS	0.1703	0.1718	*0.1337	1.6403	*0.1931
	BL+IR		**0.1707**	**0.1721**	***0.1354**	1.6433	***0.1945**
	1000-best	oracle	0.2360	0.2412	0.2052	1.9472	0.2783
	BL	PLS	0.1707	0.1719	*0.1353	1.6328	*0.1949
	IR	PLS	***0.1716**	**0.1723**	*0.1355	1.6363	*0.1965
	BL+IR	PLS	*0.1715	0.1718	***0.1362**	**1.6384**	***0.1992**

*Table 7. DQET13-nbest results. * achieve improvements; top results are in **bold**.*

Combination is done at the word-level by using lattice re-scoring to obtain combined translations from translation hypotheses that minimize overall word error.

Re-ranking results show that we can improve over 1-best results, with 100-best lists leading to the best results. Word-level combination results show that the performance increase as we increase n and the best results are obtained with 1000-best lists where 1000 is the largest n we experimented with. We predict F_1 scores and retain the top results among different settings achieving improvements according to F_1 or overall.

Word-level combination is obtained by converting each n-best list into a word lattice and finding the word-level combination of translation hypotheses that minimizes WER. A word lattice is a partially ordered graph with word hypotheses at the nodes (Mangu et al., 2000). An n-best lattice rescoring (Mangu et al., 2000) functionality is provided by the SRILM (Stolcke, 2002) toolkit. Each hypothesis in a given n-best list is weighted with the predicted scores, converted into a word lattice format, aligned, and the best hypothesis minimizing the WER is selected as the consensus hypothesis. As we see in the results, the word-level combination approach is able to improve the performance more than sentence-level re-ranking due to reasons including (Mangu et al., 2000): (i) lattice representation is able to consider alternative translations, (ii) pruning of the lattices minimizes word errors and leads to better modeling of word posterior probabilities, (iii) WER minimization may be a good target to optimize for translation performance.

	Features	Setting	BLEU	F_1	BLEUs	NISTs	1-WER
	English-Spanish	1-best	0.2690	0.2673	0.2228	2.3278	0.3920
	100-best	oracle	0.3560	0.3667	0.3351	2.6493	0.4940
Re-ranking	BL	FS	0.2535	0.2456	0.1947	2.2702	0.3187
	IR	PLS	0.2516	0.2457	0.2010	2.2339	**0.3817**
	SMT	PLS	0.2569	**0.2498**	0.2008	**2.2804**	0.3451
	BL+IR		0.2567	0.2465	0.2011	2.2614	0.3610
	IR+SMT	PLS	**0.2576**	0.2495	**0.2024**	2.2796	0.3574
	BL+IR+SMT		0.2564	0.2482	0.2012	2.2741	0.3585
Word Comb.	BL	FS	0.2676	0.2653	0.2208	**2.3212**	*0.3925
	IR	FS	**0.2682**	**0.2661**	**0.2216**	2.3207	***0.3936**
	SMT		0.2672	0.2648	0.2196	2.3197	*0.3928
	BL+IR	PLS	0.2676	0.2651	0.2204	2.3198	0.3927
	IR+SMT	PLS	0.2677	0.2647	0.2196	2.3179	0.3915
	BL+IR+SMT		0.2677	0.2652	0.2198	2.3195	0.3926
	Spanish-English	1-best	0.2816	0.2816	0.2335	2.3696	0.4064
	100-best	oracle	0.3763	0.3902	0.3554	2.6858	0.5154
Re-ranking	BL	PLS	0.2656	**0.2614**	**0.2112**	**2.3441**	0.3663
	IR	FS	0.2646	0.2553	0.2025	2.2888	0.3819
	SMT	FS	0.2602	0.2578	0.2052	2.3094	0.3559
	BL+IR	FS	0.2660	0.2573	0.2051	2.2932	**0.3849**
	IR+SMT	PLS	0.2616	0.2552	0.2031	2.2940	0.3692
	BL+IR+SMT	FS	**0.2662**	0.2572	0.2045	2.2922	0.3827
Word Comb.	BL	PLS	***0.2818**	**0.2803**	**0.2315**	**2.3686**	*0.4124
	IR		***0.2818**	0.2801	0.2314	2.3642	*0.4127
	SMT	PLS	*0.2817	0.2795	0.2301	2.3628	*0.4116
	BL+IR	FS	0.2815	0.2792	0.2298	2.3673	*0.4128
	IR+SMT	FS	0.2815	0.2793	0.2304	2.3652	***0.4131**
	BL+IR+SMT	PLS	0.2816	0.2797	0.2304	2.3636	*0.4127

*Table 8. DFDA13-nbest results with 100-best lists. Top results are in **bold**.*

Table 7 presents 1-best translation baseline, oracle translation according to F_1, and the 1-best translation results after sentence-level re-ranking according to the predicted F_1 scores using 50-best or 100-best lists, and after word-level combination using 250-best or 1000-best lists on the DQET13-nbest dataset. QuEst is able to improve the performance on all metrics, with IR or BL+IR feature sets obtain the best results. F_1 is also improved with word-level combination using a 100-best list (Specia et al., 2014). We obtain up to 0.28 points increase in BLEUs with the word-level combination using a 1000-best list. With sentence-level re-ranking, we are able to improve only the NISTs scores using a 50-best list.

With sentence-level re-ranking, we observe that performance decrease as we increase n from 50 to 1000. Results for different n-best lists with increasing n are presented in (Specia et al., 2014). With word-level combination, performance increase as we increase n on DQET-nbest and on DFDA13-nbest, we observe increasing performance on some metrics with increasing n where the best results are obtained with 100-best lists.

Table 8 presents the corresponding results on the DFDA13-nbest English-Spanish and Spanish-English datasets using 100-best lists. Sentence-level re-ranking does not yield improvements. With word-level combination, we are able to achieve improvements according to BLEU and 1-WER. The performance gains are larger on DFDA13-nbest and the addition of SMT features improve the performance slightly. The IR feature set and other feature sets containing IR lead to the best results. Previous results such as (Blatz et al., 2004) could not improve the BLEU scores with n-best list re-scoring, but obtained some improvements in NISTs scores.

With word-level combination, performance increase as we increase n for NISTs in general and for 1-WER when translating from English to Spanish. We observed a slight decrease in 1-WER when translating from Spanish to English. NIST favors more diverse outputs by weighting less frequent n-grams more, which can explain the increase in NISTs scores with increasing n.

5. ITERPE and SSSS for Machine Translation Improvements

In these experiments we use quality estimation to identify source sentences whose translations have potential for improvement such that building sentence-specific SMT systems (SSSS) may improve their translations and the overall SMT performance. Our goal is to find instances that have suboptimal translations and for which a better translation is possible by building SSSS. In domain adaptation, we show that Moses SMT systems built with FDA-selected $10,000$ training sentences are able to obtain F_1 results as good as the baselines that use up to 2 million sentences and better performance with Moses SMT systems built with FDA-selected $50,000$ training sentences (Biçici, 2015). In fact, SSSS built using as few as $5,000$ training instances for each source sentence can achieve close performance to a baseline SMT system using 2 million sentences (Biçici, 2011). In Table 14, we show that we can achieve better NIST performance using SSSS built with FDA5-selected $5,000$ training instances. The ITERPE model allows us to identify which sentences have more potential for improvement by re-translation and we demonstrate performance improvements with SSSS. An ITERPE sorting of the translation instances can be used to group them into different quality bands for different purposes, for instance, for re-translation or for post-editing.

5.1. ITERPE: Identifying Translation Errors Regardless of Prediction Errors

We use the IR feature set to build two QuEst systems: QuEst_S and $\text{QuEst}_{(S,T')}$, where the former uses only the source sentence S and the latter uses both S and its translation T' when predicting the quality score. $\text{QuEst}_{(S,T')}$ is a more informed pre-

	Dataset	BLEU	F_1	$\hat{y}$	$\hat{y}_S$	$\hat{y} - \hat{y}_S$	$\lvert\hat{y} - \hat{y}_S\rvert$	$\bar{F}_{1S} - F_1$	$\bar{F}_{1T} - F_1$
en-de	DFDA14-train	0.2223	0.2360	0.2283	0.2436	-0.0153	0.1069	0.1734	0.1554
	DFDA14-test	0.1761	0.2093	0.2147	0.1800	0.0347	0.0738	0.1194	0.1545
de-en	DFDA14-train	0.2570	0.2580	0.2578	0.2649	-0.0071	0.0096	0.1094	0.1478
	DFDA14-test	0.2410	0.2580	0.2529	0.2663	-0.0259	0.1535	0.0400	0.1248

Table 9. Training and testing average statistics and ITERPE results. y is F_1.

dictor. Biçici et al. (2013) obtained better results than QuEst using only the source sentences with the machine translation performance prediction system.

We consider two types of errors: prediction error or translation error. Prediction errors are errors due to the prediction model, while translation errors are errors due to mistranslations by the MT engine. We want to fix translation errors regardless of potential prediction errors. Having a precise estimator (low MAE) is important for identifying the score differences. Also, if the prediction reaches the top possible target score, topScore, where $y \leq$ topScore then we do not expect to be able to further improve the translation. Let $\hat{y}_S = \text{QuEst}_S(S)$ and $\hat{y} = \text{QuEst}_{(S,T')}(S, T')$ represent the prediction from QuEst using only the source S and using both S and T′ and t_y be a positive threshold on the prediction. When predicting which instances to re-translate, we compare two strategies, which sort instances according to **d**:

MEAN: $\mathbf{d} = \bar{\hat{\mathbf{y}}} - \hat{\mathbf{y}}$. Selects instances whose expected performance is lower than the expected mean performance, which attempts to improve lower performing instances.

ITERPE: $\mathbf{d} = \hat{\mathbf{y}} - \hat{\mathbf{y}}_S$. Selects instances according to differences in predictions from different predictors, which attempts to identify the translation errors regardless of prediction errors (ITERPE).

The ITERPE strategy relies on the performance prediction of the quality of a sentence translation task by two separate predictors, one using only the source sentence and one using both the source sentence and the translation. ITERPE invention (Biçici, 2014) works as follows:

- If $\hat{y}_S > \hat{y}$ and $\hat{y}_S <$ topScore, then by looking at S, we expect a better translation performance. So, T′ is not optimal.
- If $\hat{y}_S = \hat{y}$, then either the quality score is the same for both or T′ is not giving us new information. If $(\hat{y} - \hat{y}_S) \leq t_y$ and $\hat{y}_S <$ topScore, then we assume that T′ is not optimal.
- If $(\hat{y} - \hat{y}_S) > t_y$, then T′ may be close to the possible translation we can obtain.

5.2. ITERPE Learning Results

t_y can be optimized to improve the overall F_1 performance on the training set. Table 9 shows the average English to German (en-de) and German to English (de-en)

	$\hat{y}-\hat{y}_S \leq$	n	y	MAE	MAE_S	MAER	MAER_S	MRAER	MRAER_S	$\lceil y_S \rceil - y$	$\lceil y_T \rceil - y$
en-de	-0.1642	86	0.0893	0.0356	0.1517	0.3688	2.0522	0.3306	1.0422	0.2467	0.2533
	-0.0153	1920	0.1726	0.0639	0.1123	0.3624	0.8543	0.4582	0.8612	0.199	0.1935
	0.0	123	0.2206	0.0841	0.0866	0.4393	0.4636	0.5577	0.5816	0.1578	0.1531
	0.1335	434	0.2751	0.0636	0.0731	0.2623	0.263	0.3745	0.4361	0.1246	0.1077
	0.2823	130	0.4382	0.0613	0.1973	0.1598	0.4306	0.2433	0.7727	0.1122	0.0067
	0.4311	102	0.6125	0.0579	0.3607	0.0912	0.5781	0.1372	0.9219	0.1188	-0.0274
	0.5799	71	0.8007	0.0826	0.5447	0.0971	0.6757	0.1357	0.9633	-0.0054	-0.0948
	1.0	21	0.9272	0.0683	0.6713	0.0743	0.7223	0.1006	0.9721	-0.1211	-0.0841
de-en	-0.1422	228	0.1057	0.034	0.1563	0.3457	1.8446	0.3097	1.0179	0.2026	0.2645
	-0.0071	1781	0.2076	0.0719	0.1125	0.3373	0.6719	0.4556	0.7979	0.1404	0.1843
	0.0	54	0.2357	0.0527	0.0544	0.2525	0.2627	0.4006	0.4155	0.1099	0.1608
	0.128	565	0.2871	0.0484	0.0588	0.199	0.2079	0.3018	0.3713	0.0621	0.1085
	0.2631	183	0.4328	0.0514	0.1736	0.1345	0.3869	0.2118	0.7291	0.008	0.0002
	0.3983	99	0.6079	0.06	0.341	0.0981	0.5514	0.1574	0.9173	-0.0177	-0.0778
	0.5334	56	0.7869	0.0889	0.5148	0.1163	0.6417	0.1593	0.9521	-0.1157	-0.123
	1.0	20	0.9064	0.094	0.636	0.103	0.6996	0.1442	0.9805	-0.1924	-0.1686

Table 10. DFDA14-train instances binned according to their deviation from $\bar{\mathbf{d}}$ using ITERPE. y is F_1.

training set and test set statistics. In this case, the target, y, is the F_1 score. MAE is for $\hat{y}$ and MAE_S is for $\hat{y}_S$. $\lceil y_S \rceil$ and $\lceil y_T \rceil$ are the source and target performance bounds on y calculated based on synthetic translations (Biçici et al., 2013). Score differences to bounds show a measure of how close are the translations to the bounds.

We cumulatively and separately bin instances according to their deviation from the mean of $\mathbf{d}$, $\bar{\mathbf{d}}$, in $\sigma_{\mathbf{d}}$ steps and evaluate the prediction performance of different strategies. Table 11 shows the cumulatively binning performance on the training set where predictions are obtained by cross-validation and Table 10 show the training results within separate bins. n is the number of instances in the score range. Table 11 also shows that ITERPE is able to identify hard to translate instances using only the prediction information.

Table 12 and Table 13 show the cumulatively binning of the performance of the test set instances according to their deviation from $\bar{\mathbf{d}}$ for both the en-de and de-en DFDA14-test sets. We observe that lower $\mathbf{d}$ corresponds to instances with lower F_1 scores (compared to $\bar{\mathbf{y}}$) and higher potential for improvements according to $\lceil y_S \rceil - y$ and $\lceil y_T \rceil - y$. Until about $d \leq \bar{\mathbf{d}} + 2\sigma_{\mathbf{d}}$, MAE_S is lower, which indicates that for these instances, we can trust $\hat{y}_S$ more than $\hat{y}$ and therefore, since $\hat{y}_S > \hat{y}$ when $d < 0$, these instances correspond to the instances that have some potential for improvement. Including the confidence of predictions in the decision process may also improve the performance.

	$\mathbf{d} \leq$	n	$\bar{y}$	MAE	MAE_S	MAER	MAER_S	MRAER	MRAER_S	$\lceil y_S \rceil - y$	$\lceil y_T \rceil - y$
en-de	-0.1642	86	0.0893	0.0356	0.1517	0.3688	2.0522	0.3306	1.0422	0.2467	0.2533
	-0.0153	2006	0.1691	0.0627	0.1139	0.3626	0.9056	0.4527	0.869	0.2011	0.1961
	0.0	2129	0.172	0.0639	0.1124	0.3671	0.8801	0.4588	0.8524	0.1986	0.1936
	0.1335	2563	0.1895	0.0639	0.1057	0.3493	0.7756	0.4445	0.7819	0.186	0.179
	0.2823	2693	0.2015	0.0638	0.1101	0.3402	0.759	0.4348	0.7815	0.1825	0.1707
	0.4311	2795	0.2165	0.0635	0.1193	0.3311	0.7524	0.4239	0.7866	0.1802	0.1635
	0.5799	2866	0.231	0.064	0.1298	0.3253	0.7505	0.4168	0.791	0.1756	0.1571
	1.0	2887	0.236	0.064	0.1338	0.3235	0.7502	0.4145	0.7923	0.1734	0.1553
de-en	-0.1422	228	0.1057	0.034	0.1563	0.3457	1.8446	0.3097	1.0179	0.2026	0.2645
	-0.0071	2009	0.196	0.0676	0.1175	0.3383	0.805	0.4391	0.8229	0.1475	0.1934
	0.0	2063	0.1971	0.0672	0.1158	0.336	0.7908	0.438	0.8122	0.1465	0.1925
	0.128	2628	0.2164	0.0632	0.1036	0.3066	0.6655	0.4087	0.7174	0.1283	0.1745
	0.2631	2811	0.2305	0.0624	0.1081	0.2954	0.6473	0.3959	0.7182	0.1205	0.1631
	0.3983	2910	0.2433	0.0623	0.116	0.2887	0.6441	0.3878	0.725	0.1158	0.1549
	0.5334	2966	0.2536	0.0628	0.1236	0.2854	0.644	0.3835	0.7292	0.1114	0.1497
	1.0	2986	0.258	0.063	0.127	0.2842	0.6444	0.3819	0.7309	0.1094	0.1475

Table 11. DFDA14-train instances cumulatively binned based on deviation from $\bar{\mathbf{d}}$ using ITERPE. y is F_1.

	$\hat{y} - \hat{y}_S \leq$	n	y	MAE	MAE_S	MAER	MAER_S	MRAER	MRAER_S	$\lceil y_S \rceil - y$	$\lceil y_T \rceil - y$
MEAN	-0.1035	91	0.192	0.1106	0.1202	0.475	0.9383	1.0233	0.9838	0.1738	0.2155
	0.0	1792	0.2013	0.0854	0.1025	0.5055	0.7783	0.8717	1.1362	0.1256	0.1522
	0.0	1792	0.2013	0.0854	0.1025	0.5055	0.7783	0.8717	1.1362	0.1256	0.1522
	0.1035	2449	0.2086	0.094	0.1064	0.5802	0.7722	0.9338	1.1267	0.1172	0.1503
	0.2069	2609	0.2095	0.0992	0.1065	0.6291	0.7691	1.0152	1.1211	0.1175	0.1513
	0.3104	2673	0.2094	0.1038	0.107	0.6732	0.7723	1.0691	1.1197	0.1185	0.1524
	0.4139	2703	0.2093	0.107	0.1071	0.7021	0.7738	1.1087	1.1184	0.119	0.1535
	1.0	2737	0.2093	0.1125	0.1075	0.747	0.7764	1.173	1.119	0.1195	0.1548
ITERPE	-0.129	54	0.2217	0.1215	0.1294	0.4418	0.8893	1.0072	1.1438	0.1209	0.1647
	-0.0202	1874	0.2036	0.0856	0.1041	0.4963	0.7788	0.8894	1.1932	0.1168	0.1467
	0.0	2053	0.2046	0.0878	0.105	0.5225	0.7846	0.8941	1.1761	0.1166	0.1477
	0.0885	2438	0.2083	0.0942	0.1065	0.5791	0.7737	0.9385	1.1394	0.1161	0.1497
	0.1972	2606	0.2093	0.0992	0.1065	0.6257	0.767	1.0161	1.1285	0.117	0.1513
	0.306	2671	0.2092	0.1036	0.1067	0.6682	0.7696	1.0733	1.1276	0.1181	0.1523
	0.4147	2704	0.2093	0.1071	0.1072	0.6998	0.7715	1.1057	1.1252	0.1186	0.1534
	1.0	2737	0.2093	0.1125	0.1075	0.7414	0.7728	1.1735	1.1258	0.1192	0.1546

Table 12. DFDA14-test instances cumulatively binned based on deviation from $\bar{\mathbf{d}}$ for en-de comparing different strategies. y is F_1.

	$\hat{y} - \hat{y}_S \leq$	n	y	MAE	MAE_S	MAER	MAER_S	MRAER	MRAER_S	$\lceil y_S \rceil - y$	$\lceil y_T \rceil - y$
MEAN	-0.1065	87	0.2668	0.1552	0.1299	0.4913	0.663	1.1651	0.8625	0.0633	0.1363
	-0.0	2090	0.2553	0.1023	0.1033	0.4303	0.5538	0.9059	0.9115	0.0344	0.1146
	0.0	2090	0.2553	0.1023	0.1033	0.4303	0.5538	0.9059	0.9115	0.0344	0.1146
	0.1065	2720	0.2559	0.1053	0.1041	0.4863	0.56	0.9321	0.9067	0.0383	0.1205
	0.2131	2860	0.2559	0.1094	0.1044	0.5201	0.5605	0.9921	0.9077	0.0396	0.1222
	0.3196	2922	0.2567	0.1122	0.1046	0.5379	0.5585	1.0357	0.906	0.0397	0.1226
	0.4262	2963	0.257	0.1155	0.105	0.561	0.5605	1.0695	0.9062	0.0402	0.1238
	1.0	3003	0.258	0.1205	0.1059	0.5896	0.561	1.1067	0.9045	0.04	0.1247
ITERPE	-0.126	114	0.2829	0.1408	0.1154	0.4528	0.6344	1.2341	1.0201	-0.0163	0.0774
	-0.014	2019	0.2565	0.1023	0.1035	0.4232	0.5549	0.9101	0.9206	0.0314	0.1121
	0.0	2204	0.2564	0.1029	0.1042	0.4343	0.5569	0.9049	0.9165	0.0324	0.1137
	0.098	2708	0.2558	0.1052	0.1042	0.4851	0.5615	0.9325	0.9092	0.0384	0.1202
	0.2101	2863	0.2561	0.1094	0.1045	0.5183	0.5603	0.9956	0.9085	0.0395	0.1221
	0.3221	2930	0.2567	0.1129	0.1047	0.5437	0.5605	1.041	0.906	0.0399	0.1228
	0.4341	2966	0.2569	0.116	0.1049	0.5645	0.561	1.0785	0.9058	0.0404	0.1239
	1.0	3003	0.258	0.1205	0.1059	0.59	0.5616	1.1097	0.9044	0.0401	0.1247

Table 13. DFDA14-test instances cumulatively binned based on deviation from $\bar{\mathbf{d}}$ *for de-en comparing different strategies.* y *is* F_1*.*

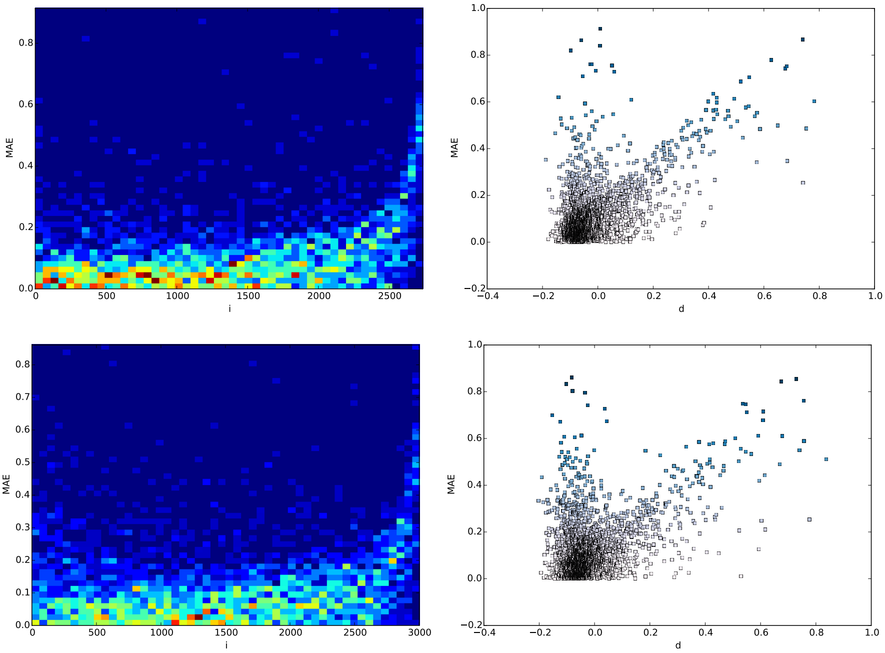

Figure 1. ITERPE sorted MAE histogram and **d** *vs. MAE plot for en-de (top) and de-en (bottom) on the DFDA14-test test set. The increase in MAE is visible.*

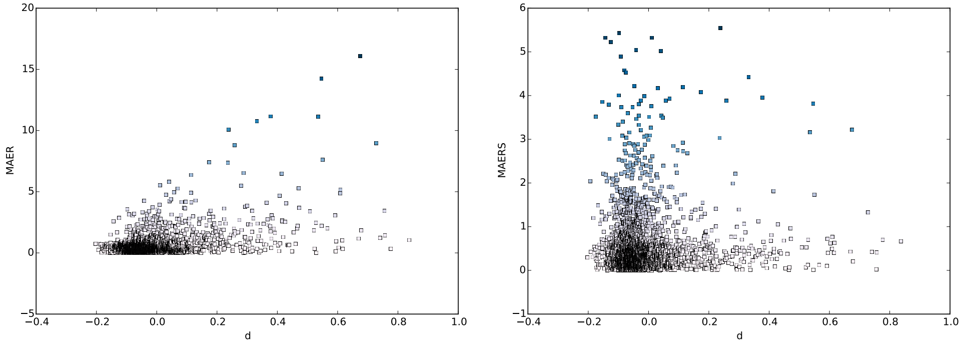

Figure 2. ITERPE sorted MAER vs. MAER$_S$ plot for de-en test set.

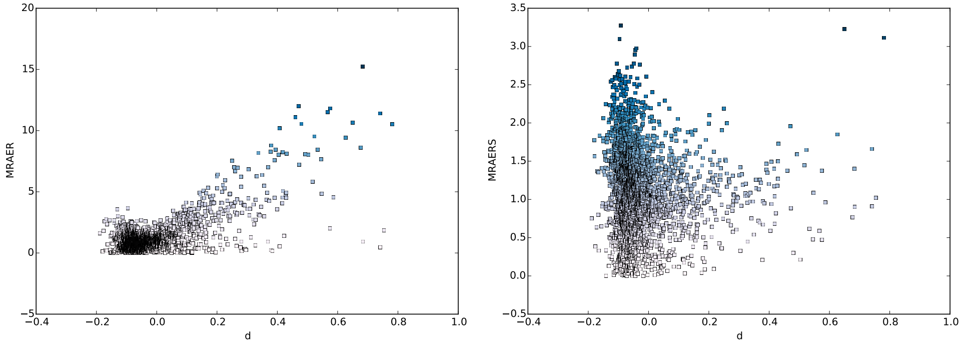

Figure 3. ITERPE sorted MRAER vs. MRAER$_S$ plot for en-de test set.

We observe that the difference with the upper bounds increase with decreasing **d**, which indicates that we are able to identify instances that have the highest potential for increase in their performance. ITERPE is able to consistently identify instances with higher potential difference to the bounds. Figure 1 plots ITERPE sorted MAE histogram and **d** vs. MAE plot for en-de and de-en on the DFDA14-test test set.

Figure 2 compares the MAER and MAER$_S$ distributions on de-en DFDA14-test test set instances sorted according to ITERPE and Figure 3 compares the MRAER and MRAER$_S$ distributions on en-de DFDA14-test test set.

5.3. SSSS with ITERPE Sorted Instances

We build SSSS over the en-de and de-en DFDA14-test set instances using up to $25,000$ training instances selected specifically for each source sentence using FDA5 (Biçici and Yuret, 2015). The results are presented in Table 14, which show that using training set of $1,000$ instances for building SSSS does not lead to better results on all of DFDA14-test. However, SSSS with $1,000$ training instances each can obtain as close results as 1 F_1 point in en-de and shows that this level of parallelism is possible with SMT. en-de translation performance also improves with SSSS where each use $5,000$ training instances.

We also run sentence-specific SMT experiments over the top instances that have more potential for improvement according to their ITERPE sorting. Table 14 also presents the results for the top 100 or 200 instances compared to baseline translation performance on those instances. Improvements of up to 1.43 BLEU, 0.54 F_1, 2.9 NIST, 0.64 BLEUs, and 4.7 NISTs points are obtained over the top 100 ITERPE sorted instances for en-de. Compared to baseline results, ITERPE sorting is able to obtain up to 1.43 BLEU points improvement over the top 100 instances by being able to identify the top instances that have more potential for improvement by re-translation with SSSS.

Figure 4 plots ITERPE sorted accumulative average performance improvement compared with the baseline over the top 500 instances in en-de and de-en on the DFDA14-test test set. Maximum accumulative gain with SSSS on the DFDA14-test test set over the top 100 ITERPE sorted instances is visible in Figure 4 and can reach 4.4 BLEUs points and 3.7 F_1 points for en-de, and 9.1 BLEUs points and 7.2 F_1 points for de-en. Maximum instance gains with SSSS on the DFDA14-test test set over the top 100 ITERPE sorted instances are presented in Table 15.

6. Conclusions

We described several ways to use quality predictions produced by the QuEst framework in order to improve SMT performance without changing the MT systems' internal functioning. In all cases, promising results were found, leading us to believe that quality predictions produced by the state of the art approaches can help in the process of achieving the high quality translation goal. Table 16 summarizes our main findings when we use quality predictions towards improving machine translation quality.

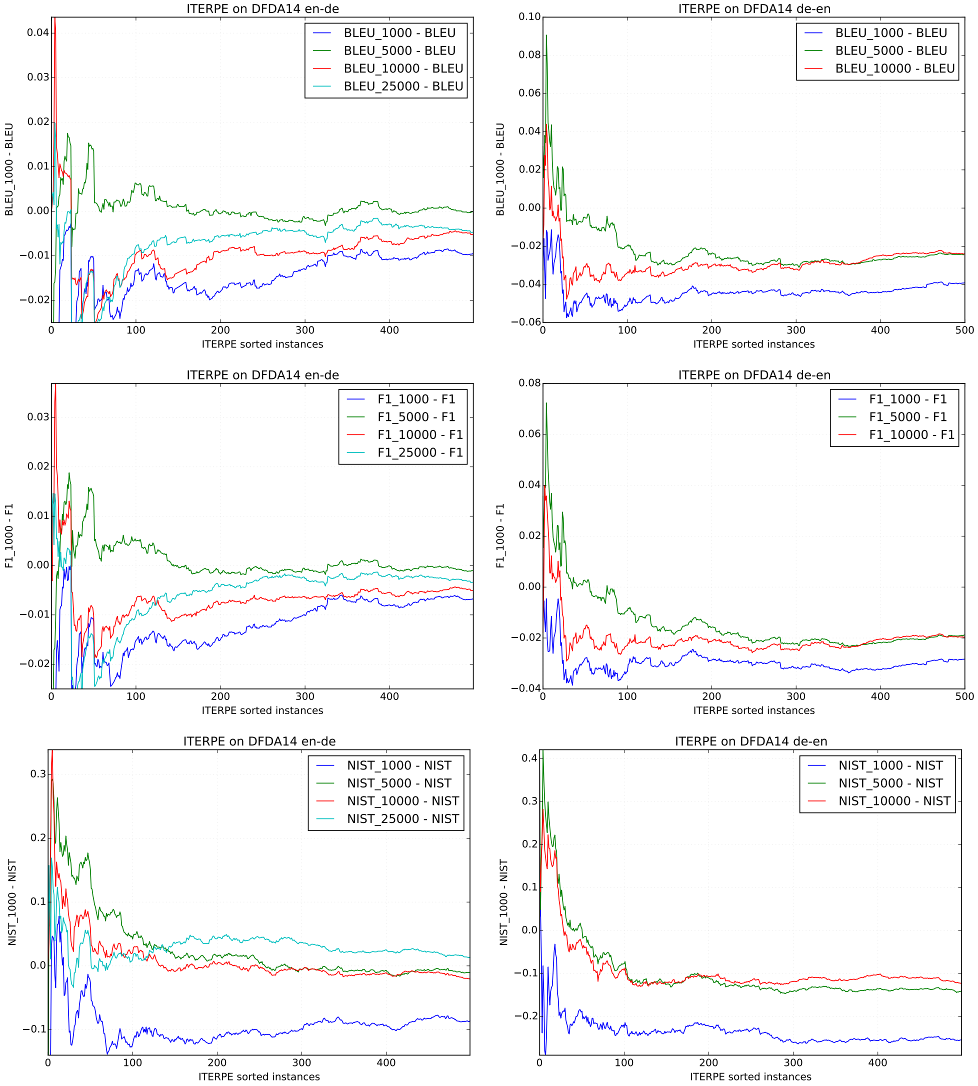

Figure 4. ITERPE sorted accumulative average F_1, BLEUs, or NISTs performance improvement compared with the baseline over the top 500 instances in en-de (left) and de-en (right) on the DFDA14-test test set.

			Baseline					ITERPE + SSSS				
	n	# train	BLEU	F_1	NIST	BLEUs	NISTs	BLEU	F_1	NIST	BLEUs	NISTs
en-de	100	1000	0.1614	0.2048	4.6227	0.1406	1.9119	0.1609	0.1893	4.6152	0.1271	1.7847
		5000						**0.1757**	**0.2102**	**4.9111**	**0.147**	**1.9587**
		10000						**0.1633**	0.1974	**4.8022**	0.1309	**1.9314**
		25000						**0.1647**	0.1942	**4.7821**	0.1323	**1.9192**
	200	1000	0.1569	0.1952	4.9408	0.1284	1.9558	0.1444	0.18	4.9064	0.1102	1.8408
		5000						**0.1577**	0.1937	**5.1326**	0.1278	**1.9715**
		10000						0.1542	0.1872	**5.0878**	0.1191	**1.9587**
		25000						**0.1574**	0.1909	**5.1462**	0.1226	**1.9982**
	2737	1000	0.1761	0.2093	5.9602	0.1435	2.0366	0.1632	0.199	5.9596	0.1295	1.9277
	2737	5000						0.1725	0.2035	**6.072**	0.1378	1.9821
	1500	10000	0.1726	0.2017	5.8253	0.1373	2.1071	0.1678	0.1958	**5.9166**	0.1314	2.0702
	750	25000	0.1741	0.1998	5.6429	0.1364	2.0981	**0.1747**	0.198	**5.7945**	0.1341	**2.109**
de-en	100	1000	0.2831	0.2842	5.8797	0.2446	2.5136	0.2402	0.2509	5.3697	0.1935	2.2959
		5000						0.2672	0.2749	5.7442	0.2246	2.4365
		10000						0.2602	0.2611	5.7401	0.2103	2.4243
	200	1000	0.2849	0.2918	6.337	0.2476	2.5262	0.2421	0.263	5.7436	0.2022	2.3065
		5000						0.2625	0.2759	6.1281	0.2234	2.4169
		10000						0.2599	0.2703	6.1347	0.2168	2.4199
	3003	1000	0.241	0.258	7.2325	0.198	2.3194	0.2007	0.233	6.4235	0.1663	2.0358
	1250	5000	0.2516	0.2623	7.0413	0.2095	2.4641	0.2269	0.2465	6.7374	0.1897	2.3049
	700	10000	0.267	0.2752	6.8915	0.2269	2.5233	0.2438	0.2592	6.6454	0.2072	2.3977

Table 14. ITERPE + SSSS results on DFDA14-test over the top n ITERPE sorted instances. We build SSSS for each instance.

		ITERPE + SSSS		
	# train	BLEUs	F_1	NISTs
en-de	1000	24.6	18.8	78.4
	5000	27.7	21.4	105.8
	10000	21.1	18.9	126.4
	25000	49.3	44.9	120.3
de-en	1000	31.4	26.9	99.8
	5000	54.2	51.1	124.2
	10000	32.0	27.1	113.3

Table 15. Maximum instance improvement points with ITERPE + SSSS on DFDA14-test test set over the top 100 ITERPE sorted instances.

Improvement Points	BLEU	F_1	NIST	BLEUs	NISTs
multiple system translation ranking	2.72	2.17		3.66	
n-best list re-ranking					0.41
n-best list combination	0.06			0.28	0.31
ITERPE (en-de, $n = 100$)	1.43	0.54	2.9	0.64	4.7

Table 16. Summary of improvement points over the baseline obtained using QuEst *for high quality machine translation.*

The ITERPE model can obtain robust sortings of the translations allowing us to answer questions about which translations do not have much potential for improvement and which may need to be re-translated or maybe post-edited. We build sentence specific SMT systems on the ITERPE sorted instances identified as having more potential for improvement and obtain improvements in BLEU, F_1, NIST, BLEUs, and NISTs.

Acknowledgements

This work is supported in part by SFI as part of the ADAPT CNGL Centre for Global Intelligent Content (`http://www.adaptcentre.ie`, 07/CE/I1142) at Dublin City University, in part by SFI for the project "Monolingual and Bilingual Text Quality Judgments with Translation Performance Prediction" (`computing.dcu.ie/~ebicici/Projects/TIDA_RTM.html`, 13/TIDA/I2740), and in part by the European Commission through the QTLaunchPad FP7 project (`http://www.qt21.eu`, No: 296347). We also thank the SFI/HEA Irish Centre for High-End Computing (ICHEC, `http://www.ichec.ie`) for the provision of computational facilities and support.

Bibliography

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

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

Biçici, Ergun. Referential translation machines for quality estimation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 343–351, Sofia, Bulgaria, August 2013b. Association for Computational Linguistics.

Biçici, Ergun. ITERPE: Identifying translation errors regardless of prediction errors, 2014. Invention Disclosure, DCU Invent Innovation and Enterprise `https://www.dcu.ie/invent/`. USPTO filing number: US62093483.

Biçici, Ergun. Domain adaptation for machine translation with instance selection. *The Prague Bulletin of Mathematical Linguistics*, 103, 2015.

Biçici, Ergun and Deniz Yuret. RegMT system for machine translation, system combination, and evaluation. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 323–329, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W11-2137`.

Biçici, Ergun and Deniz Yuret. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*, 23:339–350, 2015. doi: 10.1109/TASLP.2014.2381882.

Biçici, Ergun, Declan Groves, and Josef van Genabith. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*, 27:171–192, December 2013. ISSN 0922-6567. doi: 10.1007/s10590-013-9138-4.

Biçici, Ergun, Qun Liu, and Andy Way. Parallel FDA5 for fast deployment of accurate statistical machine translation systems. In *Proc. of the Ninth Workshop on Statistical Machine Translation*, pages 59–65, Baltimore, USA, June 2014. Association for Computational Linguistics.

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

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. Confidence estimation for machine translation. Technical report, 2004.

Bojar, Ondřej, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-2201`.

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

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

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

Koehn, Philipp, 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. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180, Prague, Czech Republic, June 2007.

Mangu, Lidia, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400, 2000.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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 2002.

Smola, Alex J. and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, Aug. 2004. ISSN 0960-3174.

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. Estimating the sentence-level quality of machine translation systems. In *Proc. of the 13th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 28–35, Barcelona, Spain, May 2009.

Specia, Lucia, Dhwaj Raj, and Marco Turchi. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50, 2010. ISSN 0922-6567. doi: 10.1007/s10590-010-9077-2. URL `http://dx.doi.org/10.1007/s10590-010-9077-2`.

Specia, Lucia, Kashif Shah, Eleftherios Avramidis, and Ergun Biçici. QTLaunchPad deliverable D2.1.3 quality estimation for dissemination, 2013. URL `http://www.qt21.eu/launchpad/deliverable/quality-estimation-dissemination`.

Specia, Lucia, Kashif Shah, Eleftherios Avramidis, and Ergun Biçici. QTLaunchPad deliverable D2.2.1 quality estimation for system selection and combination, 2014. URL `http://www.qt21.eu/launchpad/deliverable/quality-estimation-system-selection-and-combination`.

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

The Apache Software Foundation. Lucene, 2014. URL `http://lucene.apache.org/`.

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

Address for correspondence:
Ergun Biçici
`ergun.bicici@computing.dcu.ie`
ADAPT CNGL Centre for Global Intelligent Content
School of Computing
Dublin City University
Dublin 9, Dublin, Ireland