Referential Translation Machines for Predicting Translation Quality

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

Andy Way
Centre for Next Generation Localisation
School of Computing
Dublin City University, Dublin, Ireland.
away@computing.dcu.ie

Abstract

We use 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 interpretants selected in the same domain, which are effective when making monolingual and bilingual similarity judgments. 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 and achieve the top performance in WMT13 quality estimation task (QET13). We improve our RTM models with the Parallel FDA5 instance selection model, with additional features for predicting the translation performance, and with improved learning models. We develop RTM models for each WMT14 QET (QET14) subtask, obtain improvements over QET13 results, and rank 1st in all of the tasks and subtasks of QET14.

1 Introduction

We use 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. 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 can predict translation quality by looking at the test source sentences and becomes the 2nd overall after also looking at the translation outputs as well in QET12 (Callison-Burch et al., 2012). RTMs achieve the top performance in QET13 (Bojar et al., 2013), ranking 1st or 2nd in all of the subtasks. RTMs rank 1st in all of the tasks and subtasks of QET14 (Bojar et al., 2014).

Referential translation models (Section 2) present an accurate and language independent solution for predicting the performance of natural language tasks such as the quality estimation of translation. We improve our RTM models (Biçici, 2013) by:

- using a parameterized, fast implementation of FDA, FDA5, and our Parallel FDA5 instance selection model (Biçici et al., 2014),
- better modeling of the language in which similarity judgments are made with improved optimization and selection of the LM data,
- increased feature set for also modeling the structural properties of sentences,
- extended learning models.

2 Referential Translation Machine (RTM)

Referential translation machines provide a computational model for quality and semantic similarity judgments in monolingual and bilingual settings using retrieval of relevant training data (Biçici, 2011; Biçici and Yuret, 2014) as interpretants for reaching shared semantics (Biçici, 2008). RTMs achieve top performance when predicting the quality of translations in QET14 and QET13 (Biçici,

2013), top performance when predicting monolingual cross-level semantic similarity (Jurgens et al., 2014), good performance when evaluating the semantic relatedness of sentences and their entailment (Marelli et al., 2014), and a language independent solution and good performance when judging the semantic similarity of sentences (Agirre et al., 2014; Biçici and Way, 2014).

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. An RTM model is based on the selection of interpretants, data close to both the training set and the test set, which allow shared semantics by providing context for similarity judgments. In semiotics, an interpretant I interprets the signs used to refer to the real objects (Biçici, 2008). Each RTM model is a data translation model between the instances in the training set and the test set. We use the Parallel FDA5 (Feature Decay Algorithms) instance selection model for selecting the interpretants (Biçici et al., 2014; Biçici and Yuret, 2014) this year, which allows efficient parameterization, optimization, and implementation of FDA, and build an MTPP model (Section 2.1). We view that acts of translation are ubiquitously used during communication:

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

Given a training set $\texttt{train}$, a test set $\texttt{test}$, and some corpus $\mathcal{C}$, preferably in the same domain as the training and test sets, the RTM steps are:

1. FDA5$(\texttt{train}, \texttt{test}, \mathcal{C}) \rightarrow \mathcal{I}$
2. MTPP$(\mathcal{I}, \texttt{train}) \rightarrow \mathcal{F}_{\texttt{train}}$
3. MTPP$(\mathcal{I}, \texttt{test}) \rightarrow \mathcal{F}_{\texttt{test}}$
4. learn$(M, \mathcal{F}_{\texttt{train}}) \rightarrow \mathcal{M}$
5. predict$(\mathcal{M}, \mathcal{F}_{\texttt{test}}) \rightarrow \hat{q}$

Step 1 selects the interpretants, $\mathcal{I}$, relevant to both the training and test data. Steps 2 and 3 use $\mathcal{I}$ to map $\texttt{train}$ and $\texttt{test}$ to a new space where similarities between translation acts can be derived more easily. Step 4 trains a learning model M over the training features, $\mathcal{F}_{\texttt{train}}$, and Step 5 obtains the predictions. RTM relies on the representativeness of $\mathcal{I}$ as a medium for building data translation models between $\texttt{train}$ and $\texttt{test}$.

Our encouraging results in QET provides a greater understanding of the acts of translation we ubiquitously use and how they can be used to predict the performance of translation and judging the semantic similarity between text. RTM and MTPP models are not data or language specific and their modeling power and good performance are applicable in different domains and tasks.

2.1 The Machine Translation Performance Predictor (MTPP)

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.

2.2 MTPP Features for Translation Acts

MTPP measures the coverage of individual test sentence features found in the training set and derives indicators of 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. 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. 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, representing the grammatical information instantiated in the training and test data.

We extend the MTPP model we used last year (Biçici, 2013) in its learning module and the features included. Categories for the features (S for source, T for target) used are listed below where the number of features are given in brackets for S and T, $\{$#S, #T$\}$, and the detailed descriptions for some of the features are presented in (Biçici et al., 2013). The number of features for each task differs since we perform an initial feature selection step on the tree structural features (Section 2.3). The number of features are in the range $337-437$.

- *Coverage* $\{$*56, 54*$\}$*:* Measures the degree to which the test features are found in the training set for both S ($\{56\}$) and T ($\{54\}$).
- *Perplexity* $\{$*45, 45*$\}$*:* 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.

- *TreeF {0, 10-110}:* 10 base features and up to 100 selected features of T among parse tree structures (Section 2.3).
- *Retrieval Closeness {16, 12}:* 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) and BLEU, F_1 (Biçici, 2011), *dice*, and tf-idf cosine similarity metrics.
- *IBM2 Alignment Features {0, 22}:* 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. We also compute word alignment percentage as in (Camargo de Souza et al., 2013) and potential BLEU, F_1, WER, PER scores for S and T.
- *IBM1 Translation Probability {4, 12}:* Calculates the translation probability of test sentences using the selected training set, $\mathcal{I}$ (Brown et al., 1993).
- *Feature Vector Similarity {8, 8}:* Calculates similarities between vector representations.
- *Entropy {2, 8}:* Calculates the distributional similarity of test sentences to the training set over top N retrieved sentences (Biçici et al., 2013).
- *Length {6, 3}:* Calculates the number of words and characters for S and T and their average token lengths and their ratios.
- *Diversity {3, 3}:* Measures the diversity of co-occurring features in the training set.
- *Synthetic Translation Performance {3, 3}:* Calculates translation scores achievable according to the n-gram coverage.
- *Character n-grams {5}:* Calculates cosine between character n-grams (for n=2,3,4,5,6) obtained for S and T (Bär et al., 2012).
- *Minimum Bayes Retrieval Risk {0, 4}:* Calculates the translation probability for the translation having the minimum Bayes risk among the retrieved training instances.
- *Sentence Translation Performance {0, 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.
- *LIX {1, 1}:* Calculates the LIX readability score (Wikipedia, 2013; Björnsson, 1968) for S and T. [1]

For Task 1.1, we have additionally used comparative BLEU, NIST, and F_1 scores as additional features, which are obtained by comparing the translations with each other and averaging the result (Biçici, 2011).

2.3 Bracketing Tree Structural Features

We use the parse tree outputs obtained by CCL to derive features based on the bracketing structure. We derive 5 statistics based on the geometric properties of the parse trees: number of brackets used (numB), depth (depthB), average depth (avg depthB), number of brackets on the right branches over the number of brackets on the left (R/L) [2], average right to left branching over all internal tree nodes (avg R/L). The ratio of the number of right to left branches shows the degree to which the sentence is right branching or not. Additionally, we capture the different types of branching present in a given parse tree identified by the number of nodes in each of its children.

Table 1 depicts the parsing output obtained by CCL for the following sentence from WSJ23 [3]:

Many fund managers argue that now 's the time to buy .

We use Tregex (Levy and Andrew, 2006) for visualizing the output parse trees presented on the left. The bracketing structure statistics and features are given on the right hand side. The root node of each tree structural feature represents the number of times that feature is present in the parsing output of a document.

3 RTM in the Quality Estimation Task

We participate in all of the four challenges of the quality estimation task (QET) (Bojar et al., 2014), which include English to Spanish (en-es), Spanish to English (es-en), English to German (en-de), and German to English (de-en) translation directions. There are two main categories of challenges: sentence-level prediction (Task 1.*) and

[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] For nodes with uneven number of children, the nodes in the odd child contribute to the right branches.

[3] Wall Street Journal (WSJ) corpus section 23, distributed with Penn Treebank version 3 (Marcus et al., 1993).

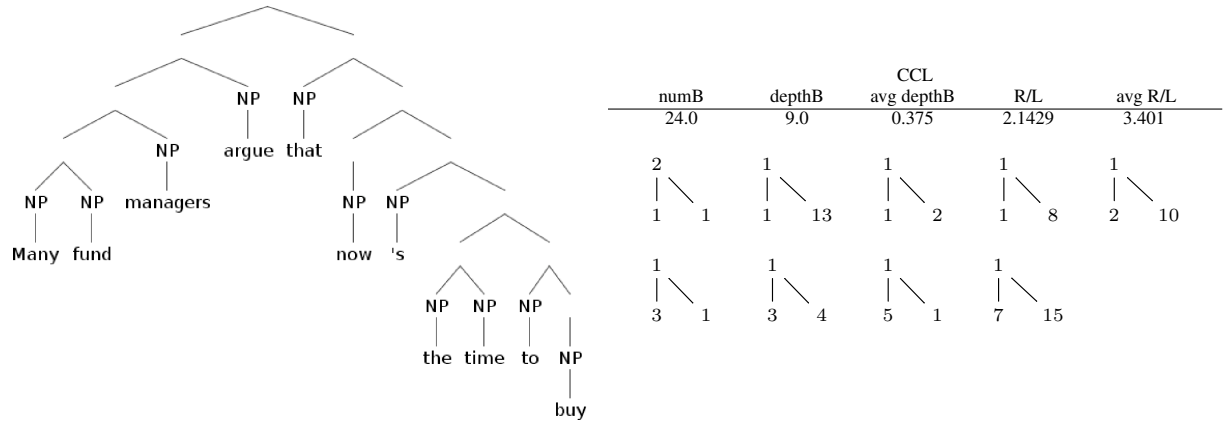

Table 1: Tree features for a parsing output by CCL (immediate non-terminals replaced with NP).

word-level prediction (Task 2). Task 1.1 is about predicting post-editing effort (PEE), Task 1.2 is about predicting HTER (human-targeted translation edit rate) (Snover et al., 2006) scores of translations, Task 1.3 is about predicting post-editing time (PET), and Task 2 is about binary, ternary, or multi-class classification of word-level quality.

For each task, we develop individual RTM models using the parallel corpora and the LM corpora distributed by the translation task (WMT14) (Bojar et al., 2014) and the LM corpora provided by LDC for English (Parker et al., 2011) and Spanish (Ângelo Mendonça, 2011) [4]. The parallel corpora contain 4.5M sentences for de-en with 110M words for de and 116M words for en and 15.1M sentences for en-es with 412M words for en and 462M words for es. We do not use any resources provided by QET including data, software, or baseline features. Instance selection for the training set and the language model (LM) corpus is handled by parallel FDA5 (Biçici et al., 2014), whose parameters are optimized for each translation task. 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 available training corpus using Moses (Koehn et al., 2007). Table 2 lists the number of sentences in the training and test sets for each task.

For each task or subtask, we select 375 thousand (K) training instances from the available parallel training corpora as interpretants for the individual RTM models using parallel FDA5. We add the selected training set to the 3 million (M) sentences selected from the available monolingual corpora for each LM corpus. The statistics of the training data selected by and used as interpretants in the RTM models is given in Table 3. The details of instance selection with parallel FDA5 are provided in (Biçici et al., 2014).

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

Task	Train	Test
Task 1.1 (en-es)	3816	600
Task 1.1 (es-en)	1050	450
Task 1.1 (en-de)	1400	600
Task 1.1 (de-en)	1050	450
Task 1.2 (en-es)	896	208
Task 1.3 (en-es)	650	208
Task 2 (en-es)	1957	382
Task 2 (es-en)	900	150
Task 2 (en-de)	715	150
Task 2 (de-en)	350	100

Table 2: Number of sentences in different tasks.

Task	S	T
Task 1.1 (en-es)	6.2	6.9
Task 1.1 (es-en)	7.9	7.4
Task 1.1 (en-de)	6.1	6
Task 1.1 (de-en)	6.9	6.4
Task 1.2 (en-es)	6.1	6.7
Task 1.3 (en-es)	6.2	6.8
Task 2 (en-es)	6.2	6.8
Task 2 (es-en)	7.5	7
Task 2 (en-de)	5.9	5.9
Task 2 (de-en)	6.3	6.8

Table 3: Number of words in $\mathcal{I}$ (in millions) selected for each task (S for source, T for target).

3.1 Learning Models and Optimization:

We use ridge regression (RR), support vector regression (SVR) with RBF (radial basis functions) kernel (Smola and Schölkopf, 2004), and ex-

Task	Translation	Model	r	RMSE	MAE	RAE
Task1.1	es-en	FS-RR	0.3512	0.6394	0.5319	0.9114
	es-en	PLS-RR	0.3579	0.6746	0.5488	0.9405
	en-de	PLS-TREE	0.2922	0.7496	0.6223	0.9404
	en-de	TREE	0.2845	0.7485	0.6241	0.9431
	en-es	TREE	0.4485	0.619	0.45	0.9271
	en-es	PLS-TREE	0.4354	0.6213	0.4723	0.973
	de-en	RR	0.3415	0.7475	0.6245	0.9653
	de-en	PLS-RR	0.3561	0.7711	0.6236	0.9639
Task1.2	en-es	SVR	0.4769	0.203	0.1378	0.8443
	en-es	TREE	0.4708	0.2031	0.1372	0.8407
Task1.3	en-es	SVR	0.6974	21543	14866	0.6613
	en-es	RR	0.6991	21226	15325	0.6817

Table 4: Training performance of the top 2 individual RTM models prepared for different tasks.

tremely randomized trees (TREE) (Geurts et al., 2006) as the learning models. TREE is an ensemble learning method over randomized decision trees. These models learn a regression function using the features to estimate a numerical target value. 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). We optimize the learning parameters, the number of features to select, the number of dimensions used for PLS, and the parameters for parallel FDA5. More detailed descriptions of the optimization processes are given in (Biçici et al., 2013; Biçici et al., 2014). We optimize the learning parameters by selecting ε close to the standard deviation of the noise in the training set (Biçici, 2013) 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 select the top 2 systems according to their performance on the training set. For Task 2, we use both Global Linear Models (GLM) (Collins, 2002) and GLM with dynamic learning (GLMd) we developed last year (Biçici, 2013). GLM relies on Viterbi decoding, perceptron learning, and flexible feature definitions. GLMd extends the GLM framework by parallel perceptron training (McDonald et al., 2010) and dynamic learning with adaptive weight updates in the perceptron learning algorithm:

$$\mathbf{w} = \mathbf{w} + \alpha \left(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}}) \right), \quad (1)$$

where Φ returns a global representation for instance i and the weights are updated by α, which dynamically decays the amount of the change during weight updates at later stages and prevents large fluctuations with updates.

3.2 Training Results

We use mean absolute error (MAE), relative absolute error (RAE), root mean squared error (RMSE), and correlation (r) to evaluate (Biçici, 2013). DeltaAvg (Callison-Burch et al., 2012) calculates the average quality difference between the top $n-1$ quartiles and the overall quality for the test set. Table 4 provides the training results.

3.3 Test Results

Task 1.1: Predicting the Post-Editing Effort for Sentence Translations: Task 1.1 is about predicting post-editing effort (PEE) and their ranking. The results on the test set are given in Table 5 where QuEst (Shah et al., 2013) SVR lists the baseline system results. Rank lists the overall ranking in the task out of about 10 submissions. We obtain the rankings by sorting according to the predicted scores and randomly assigning ranks in case of ties. RTMs with SVR PLS learning is able to achieve the top rank in this task.

Task 1.2: Predicting HTER of Sentence Translations Task 1.2 is about predicting HTER (human-targeted translation edit rate) (Snover et al., 2006), where case insensitive translation edit rate (TER) scores obtained by TERp (Snover et al., 2009) and their ranking. We derive features over sentences that are true-cased. The results on the test set are given in Table 6 where the ranks are out of about 11 submissions. We are also able to achieve the top ranking in this task.

Ranking Translations		DeltaAvg	r	Rank
en-es	TREE	0.26	-0.41	1
	PLS-TREE	0.26	-0.38	2
	QuEst SVR	0.14	-0.22	
es-en	PLS-RR	0.20	-0.35	2
	FS-RR	0.19	-0.36	3
	QuEst SVR	0.12	-0.21	
en-de	TREE	0.39	-0.54	1
	PLS-TREE	0.33	-0.42	2
	QuEst SVR	0.23	-0.34	
de-en	RR	0.38	-0.51	1
	PLS-RR	0.35	-0.45	2
	QuEst SVR	0.21	-0.25	
Scoring Translations		MAE	RMSE	Rank
en-es	TREE	0.49	0.61	1
	PLS-TREE	0.49	0.61	2
	QuEst SVR	0.52	0.66	
es-en	FS-RR	0.53	0.64	1
	PLS-RR	0.55	0.71	2
	QuEst SVR	0.57	0.68	
en-de	TREE	0.58	0.68	1
	PLS-TREE	0.60	0.71	2
	QuEst SVR	0.64	0.76	
de-en	RR	0.55	0.67	1
	PLS-RR	0.57	0.74	2
	QuEst SVR	0.65	0.78	

Table 5: RTM-DCU Task1.1 results on the test set and baseline results.

Ranking Translations		DeltaAvg	r	Rank
en-es	SVR	9.31	0.53	1
	TREE	8.57	0.48	2
	QuEst SVR	5.08	0.31	
Scoring Translations		MAE	RMSE	Rank
en-es	SVR	13.40	16.69	2
	TREE	14.03	17.48	4
	QuEst SVR	15.23	19.48	

Table 6: RTM-DCU Task1.2 results on the test set and baseline results.

Task 1.3: Predicting Post-Editing Time for Sentence Translations Task 1.3 involves the prediction of the post-editing time (PET) for a translator to post-edit the MT output. The results on the test set are given in Table 7 where the ranks are out of about 10 submissions. RTMs become the top in all metrics with RR and SVR learning models.

Task 2: Prediction of Word-level Translation Quality Task 2 is about binary, ternary, or multi-class classification of word-level quality. We develop individual RTM models for each subtask and use the GLM and GLMd learning models (Biçici, 2013), for predicting the quality at the word-level. The features used are similar to last year's (Biçici, 2013) and broadly categorized as CCL links, word context based on surrounding words, word alignments, word lengths, word locations, word prefixes and suffixes, and word forms (i.e. capital, contains digit or punctuation).

Ranking Translations		DeltaAvg	r	Rank
en-es	RR	17.02	0.68	1
	SVR	16.60	0.67	2
	QuEst SVR	14.71	0.57	
Scoring Translations		MAE	RMSE	Rank
en-es	SVR	16.77	26.17	1
	RR	17.50	25.97	7
	QuEst SVR	21.49	34.28	

Table 7: RTM-DCU Task1.3 results on the test set and baseline results.

The results on the test set are given in Table 8 where the ranks are out of about 8 submissions. RTMs with GLM or GLMd learning becomes the top this task as well.

	Model	Binary		Ternary		Multi-class	
		wF_1	Rank	wF_1	Rank	wF_1	Rank
en-es	GLM	0.351	6	0.299	5	0.268	1
	GLMd	0.329	7	0.266	6	0.032	7
es-en	GLM	0.269	2	0.220	2	0.087	1
	GLMd	0.291	1	0.239	1	0.082	2
en-de	GLM	0.453	1	0.211	2	0.150	1
	GLMd	0.369	2	0.219	1	0.125	2
en-es	GLM	0.261	1	0.083	2	0.024	2
	GLMd	0.230	2	0.086	1	0.031	1

Table 8: RTM-DCU Task 2 results on the test set. wF_1 is the average weighted F_1 score.

3.4 RTMs Across Tasks and Years

We compare the difficulty of tasks according to the RAE levels achieved. RAE measures the error relative to the error when predicting the actual mean. A high RAE is an indicator that the task is hard. In Table 9, we list the test results including the RAE obtained for different tasks and subtasks including RTM results at QET13 (Biçici, 2013). The best results are obtained for Task 1.3, which shows that we can only reduce the error with respect to knowing and predicting the mean by about 28%.

4 Conclusion

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.

Task	Translation	Model	r	RMSE	MAE	RAE
Task1.1	es-en	FS-RR	0.3285	0.6373	0.5308	0.9
	es-en	PLS-RR	0.3105	0.7124	0.5549	0.9409
	en-de	PLS-TREE	0.4427	0.7091	0.6028	0.8883
	en-de	TREE	0.5256	0.6788	0.5838	0.8602
	en-es	TREE	0.4087	0.6114	0.4938	1.0983
	en-es	PLS-TREE	0.4163	0.6084	0.4852	1.0794
	de-en	RR	0.5399	0.6735	0.5513	0.8204
	de-en	PLS-RR	0.4878	0.737	0.567	0.8437
Task1.2	en-es	SVR	0.5499	0.1669	0.134	0.8532
	en-es	TREE	0.5175	0.1748	0.1403	0.8931
Task1.3	en-es	SVR	0.6336	26174	16770	0.7223
	en-es	RR	0.6359	25966	17496	0.7536
QET13 Task1.1	en-es	PLS-SVR	0.5596	0.1683	0.1326	0.8849
		SVR	0.5082	0.1728	0.1385	0.924
QET13 Task1.3	en-es	PLS-SVR	0.6752	86.62	49.62	0.6919
		SVR	0.6682	90.36	49.21	0.6862

Table 9: Test performance of the top 2 individual RTM models prepared for different tasks and RTM results from QET13 on similar tasks (Biçici, 2013).

Acknowledgments

This work is supported in part by SFI (07/CE/I1142) as part of the CNGL Centre for Global Intelligent Content (www.cngl.org) 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

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

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 Andy Way. 2014. RTM-DCU: Referential translation machines for semantic similarity. In *SemEval-2014: Semantic Evaluation Exercises - International Workshop on Semantic Evaluation*, Dublin, Ireland, 23-24 August.

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 and Deniz Yuret. 2014. Optimizing instance selection for statistical machine translation with feature decay algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*, 27:171–192, December.

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

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

Ergun Biçici. 2013. Referential translation machines for quality estimation. 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.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, 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 1–44, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Matouš Macháček, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, and Lucia Specia. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proc. of the Ninth Workshop on Statistical Machine Translation*, Balrimore, USA, June. Association for Computational Linguistics.

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, 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.

José Guilherme Camargo de Souza, Christian Buck, Marco Turchi, and Matteo Negri. 2013. FBK-UEdin participation to the WMT13 quality estimation shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 352–358, Sofia, Bulgaria, August. Association for Computational Linguistics.

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.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

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.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. 2014. SemEval-2014 Task 3: Cross-level semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

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.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

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.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword fifth edition, Linguistic Data Consortium.

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

Kashif Shah, Eleftherios Avramidis, Ergun Biçici, and Lucia Specia. 2013. Quest - design, implementation and extensions of a framework for machine translation quality estimation. *Prague Bull. Math. Linguistics*, 100:19–30.

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, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas,*.

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.

David Graff Denise DiPersio Ângelo Mendonça, Daniel Jaquette. 2011. Spanish Gigaword third edition, Linguistic Data Consortium.