RTM Super Learner Results at Quality Estimation Task

Ergun Biçici
orcid.org/0000-0002-2293-2031
bicici.github.io

Abstract

We obtain new results using referential translation machines (RTMs) with predictions mixed to obtain a better mixture of experts prediction. Our super learner results improve the results and provide a robust combination model.

1 Introduction

Quality estimation task in WMT21 (Specia et al., 2021) (QET21) address machine translation (MT) performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence-level (Tasks 1, 2, and 3) and with classification of sentences into containing a critical error or not (Task 3). Task 1 predicts the sentence-level direct assessment (DA) in 11 language pairs categorized according to the MT resources available:

- high-resource, English–German (en-de), English–Chinese (en-zh), and Russian-English (en-ru),
- medium-resource, Romanian–English (ro-en) and Estonian–English (et-en),
- low-resource, Sinhalese–English (si-en) and Nepalese–English (ne-en), and
- no-resource, English–Czech (en-cs), English–Japanese (en-ja), Pashto–English (ps-en), and Khmer–English (km-en) for zero-shot prediction.

en-ru contains sentences from both Wikipedia and Reddit articles while others use only Wikipedia sentences with 7000 sentences for training, 1000 for development, 1000 for test QET in 2020, and 1000 for testing at QET21. The target to predict in Task 1 is z-standardised DA scores, which changes the range from $[0, 100]$ for DA scores to $[3.178, -7.542]$ in z-standardized DA scores.

	Task	Train	Test	RTM interpretants setting	Training	LM
Task 1 and Task 2	en-de	9000	1000	bilingual	0.3 M	3.5 M
	en-zh	9000	1000	bilingual	0.2 M	3.5 M
	et-en	9000	1000	bilingual	0.2 M	3.5 M
	ne-en	9000	1000	bilingual	0.2 M	3.5 M
	ro-en	9000	1000	bilingual	0.2 M	3.5 M
	ru-en	9000	1000	bilingual	0.2 M	3.5 M
	si-en	9000	1000	bilingual	0.2 M	3.5 M
	en-cs	63000	1000	bilingual	0.2 M	3.5 M
	en-ja	63000	1000	bilingual	0.2 M	3.5 M
	km-en	63000	1000	bilingual	0.2 M	3.5 M
	ps-en	63000	1000	bilingual	0.2 M	3.5 M
Task 3	en-cs	9000	1000	bilingual	0.2 M	3.5 M
	en-de	9000	1000	bilingual	0.2 M	3.5 M
	en-ja	9000	1000	bilingual	0.2 M	3.5 M
	en-zh	9000	1000	bilingual	0.2 M	3.5 M

Table 1: Number of instances in the tasks and the size of the interpretants used.

The target to predict in Task 2 is sentence HTER (human-targeted translation edit rate) scores (Snover et al., 2006). We participated in sentence-level subtasks. Table 1 lists the number of sentences in the training and test sets for each task and the number of instances used as interpretants in the referential translation machine (RTM) (Biçici and Way, 2015; Biçici, 2020) models (M for million). In zero-shot prediction, we use all of the training instances made available to the task in all 7 translation directions. We tokenize and truecase all of the corpora using Moses' (Koehn et al., 2007) processing tools.[1] Language models (LMs) are built using `kenlm` (Heafield et al., 2013).

2 RTM for MTPP

We use RTM models for building our prediction models. RTMs predict data translation between the instances in the training set and the test set using interpretants, text data selected close to the task instances in bilingual training settings or

[1] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 885–889
November 10–11, 2021.

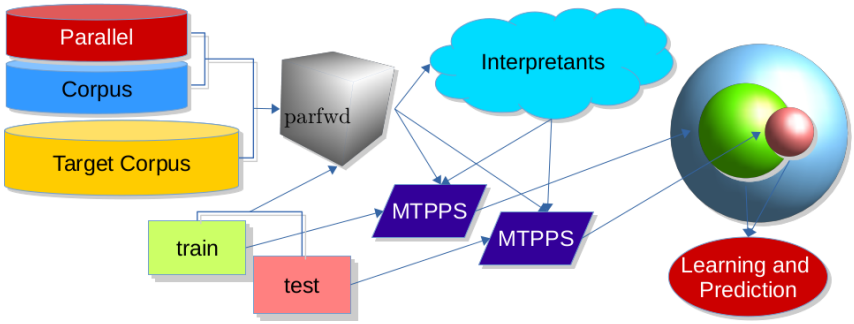

Figure 1: RTM: parfwd selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training features and another use interpretants and test data to generate test features in the same feature space (largest sphere); learning and prediction use these features as input.

monolingual LM settings. Interpretants are text data that provide context for the prediction task and are used during the derivation of the features measuring the closeness of the test sentences to the training data, the difficulty of translating them, and to identify translation acts between any two data sets for building prediction models. With the enlarging parallel and monolingual corpora made available by WMT[2], the capability of the interpretant datasets selected to provide context for the training and test sets improve with parallel feature weight decay (`parfwd`) instance selection (Biçici, 2019). RTMs use `parfwd` for instance selection and for machine translation performance prediction system (MTPPS) (Biçici et al., 2013; Biçici and Way, 2015) to obtain the features, where additional features from word alignment are added. Figure 1 depicts RTMs and explains the model building process.

We treated all of Tasks 1, 2, and 3 as bilingual tasks where parallel corpora are obtained from WMT translation task.[3] The related monolingual or bilingual datasets are used during feature extraction. The machine learning models we use include ridge regression (RR), support vector regression (SVR) (Boser et al., 1992), gradient tree boosting, extremely randomized trees (Geurts et al., 2006), and multi-layer perceptron (Bishop, 2006) in combination with feature selection (FS) (Guyon et al., 2002) and partial least squares (PLS) (Wold et al., 1984) where most of these models can be found in `scikit-learn`.[4] We use RR to estimate the noise level for SVR, which obtains accuracy with 5% error compared with estimates obtained with known noise level (Cherkassky and Ma, 2004) and set $\epsilon = \sigma/2$. We use Pearson's correlation (r), mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), relative MAE (MAER), and mean RAE relative (MRAER) as evaluation metrics (Biçici and Way, 2015). Our best non-mixed results are in Table 2. Official evaluation metric is r_P.

3 Mixture of Experts Models

We use prediction averaging (Biçici, 2018) to obtain a combined prediction from various prediction outputs better than the components, where the performance on the training set is used to obtain weighted average of the top k predictions, $\hat{y}$ with evaluation metrics indexed by $j \in J$ and weights with w:

$$\begin{aligned} w_{j,i} &= \frac{w_{j,i}}{1-w_{j,i}} \\ \hat{\boldsymbol{y}}_{\mu_k} &= \frac{1}{k}\sum_{i=1}^{k}\hat{\boldsymbol{y}}_i && \text{MEAN} \\ \hat{\boldsymbol{y}}_{j,w_k^j} &= \frac{1}{\sum_{i=1}^{k} w_{j,i}}\sum_{i=1}^{k} w_{j,i}\,\hat{\boldsymbol{y}}_i \\ \hat{\boldsymbol{y}}_k &= \frac{1}{|J|}\sum_{j\in J}\hat{\boldsymbol{y}}_{j,w_k^j} && \text{MIX} \end{aligned} \quad (1)$$

MEAN is the averaged results and MIX is the weighted average. We assume independent predictions and use $p_i/(1-p_i)$ for weights where p_i represents the accuracy of the independent classifier i in a weighted majority ensemble (Kuncheva and Rodríguez, 2014). We use the MIX prediction only when we obtain better results on the training set. We select the best model using r and mix the

[2] http://statmt.org/wmt21/

[3] http://statmt.org/wmt21/translation-task.html

[4] http://scikit-learn.org/

		r_P	MAE	RMSE
Task 1	en-de	0.212	0.4752	0.6809
	en-zh	0.223	3.8003	3.9333
	et-en	0.143	2.3699	2.5863
	ne-en	0.088	5.06	5.291
	ro-en	0.59	1.3623	1.5143
	ru-en	0.475	0.6301	0.8149
	si-en	0.21	0.8208	1.0258
	en-cs	0	7.6367	7.6871
	en-ja	0	7.5808	7.6215
	km-en	0.0209	7.4564	7.5266
	ps-en	-0.028	7.5792	7.638
Task 2	en-de	0.195	0.1605	0.2389
	en-zh	0.04	0.7707	0.8145
	et-en	0.148	0.1885	0.2271
	ne-en	0.075	0.1629	0.2058
	ro-en	0.716	0.1644	0.1927
	ru-en	0.356	0.1843	0.2383
	si-en	0.218	0.1946	0.2457
	en-cs	0.031	0.745	0.7876
	en-ja	0.031	0.3114	0.3872
	km-en	-0.094	0.3618	0.4379
	ps-en	0	0.5278	0.6322

Table 2: RTM test results in sentence-level MTPP in tasks 1 and 2 using the best non-mix result. r_P is Pearson's correlation.

results using r, RAE, MRAER, and MAER. We filter out those results with higher than 0.95 relative evaluation metric scores.

We also use generalized ensemble method (GEM) as an alternative to MIX to combine using weights and correlation of the errors, $C_{i,j}$, where GEM achieves smaller error than the best combined model (Perrone and Cooper, 1992):

$$\begin{aligned} \hat{\mathbf{y}}_{\text{GEM}} &= \sum_{i=1}^{L} w_i \psi_i(\mathbf{x}) = \mathbf{y} + \sum_{i=1}^{L} w_i \epsilon_i \\ C_{i,j} &= E[\epsilon_i, \epsilon_j] = (\psi_i(\mathbf{x}) - \mathbf{y})^T (\psi_i(\mathbf{x}) - \mathbf{y}) \\ w_i &= \frac{\sum_{j=1}^{L} C_{i,j}}{\sum_{k=1}^{L} \sum_{j=1}^{L} C_{k,j}} \end{aligned}$$

Super learner (Polley and van der Laan, 2010) is a stacking model on a library of L learning models that are V-fold cross-validated on the training set and constructs an $V \times L$ level 1 dataset. Theoretical results show that as the number of different predictors in the ensemble increase, the ensemble result gets closer to the oracle result (Dudoit and van der Laan, 2005). The function that minimize the empirical risk on the validation set will achieve lower error than the function that minimize the overall risk: $\frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\psi^*, y_i) - \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{\psi}, y_i) \geq 0$ (Vapnik, 1998).

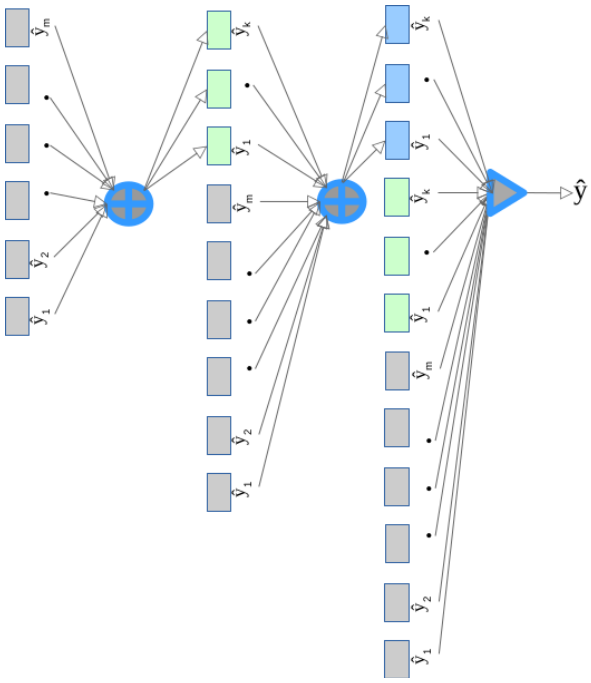

Figure 2: Model combination.

Model combination (Figure 2) selects top k combined predictions and adds them to the set of predictions where the next layer can use another model combination step or just pick the best model according to the results on the training set. We use a two layer combination where the second layer is a combination of all of the predictions obtained. The last layer is an $\arg\max$.

Our test set results using super learner are in Table 3. Before model combination, we further filter prediction results from different machine learning models based on the results on the training set to decrease the number of models combined and improve the results. A criteria that we use is MREAR ≥ 0.95 since MRAER computes the mean relative RAE score, which we want to be less than 1. In general, the combined model is better than the best model in the set. Super learner improve the results (Table 3).

The baseline deepQuest (Ive et al., 2018) use bidirectional gated recurrent unit type recurrent neural networks to model QET. RTM + deepQuest combination results in Task 2 use linear interpolation of RTM and deepQuest results with weights $0 \leq \lambda \leq 1$ and $1 - \lambda$ respectively as well as polynomial function fits to find the best combination model optimized on the development set. The most common function fit found is $f(x) =$

		r_P	MAE	RMSE
Task 1	en-de	**0.246**	0.5312	0.7699
	en-zh	**0.228**	4.4588	4.559
	et-en	0.13	2.9666	3.0942
	ne-en	0.087	**3.6449**	**3.8997**
	ro-en	0.376	3.1361	3.2656
	ru-en	0.347	0.9238	1.2276
	si-en	0.066	2.0869	2.3426
	en-cs	**0.053**	**7.0391**	**7.1159**
	en-ja	**-0.01**	**6.9076**	**6.9553**
	km-en	**0.032**	**5.6718**	**5.7694**
	ps-en	**-0.159**	**7.1563**	**7.27**
Task 2	en-de	0.125	0.1614	**0.237**
	en-zh	**-0.052**	**0.516**	**0.5648**
	et-en	**0.24**	0.2147	0.276
	ne-en	**0.299**	0.1797	0.2293
	ro-en	0.276	0.5562	0.603
	ru-en	0.143	0.2186	0.3197
	si-en	0.171	0.307	0.3713
	en-cs	**-0.108**	**0.7076**	**0.7535**
	en-ja	0.013	0.4636	0.5456
	km-en	0.008	0.5161	0.5928
	ps-en	**-0.064**	**0.4854**	**0.5671**

Table 3: RTM test results in sentence-level MTPP in tasks 1 and 2 using super learner. Improved results are shown in **bold**.

$a^x + bx^3 + cx^2 + dx + e$ (Table 4).

Task 3 results are in Table 5.

4 Conclusion

Referential translation machines pioneer a language independent approach and remove the need to access any task or domain specific information or resource and can achieve good results in automatic, accurate, and language independent prediction of translation scores. We present RTM ensemble results with super learner.

		r_P	MAE	RMSE
Task 2 deepQuest	en-de	0.225	0.1635	0.211
	en-zh	0.206	0.3033	0.3442
	et-en	0.39	0.1763	0.2256
	ne-en	0.366	0.1958	0.2441
	ro-en	0.558	0.1707	0.219
	ru-en	0.273	0.2062	0.2822
	si-en	0.338	0.2046	0.2542
Task 2 RTM + deepQuest	en-de	0.205	**0.1558**	0.2289
	en-zh	0.129	0.3786	0.4244
	et-en	**0.425**	**0.1684**	**0.2147**
	ne-en	0.353	**0.1653**	**0.2112**
	ro-en	0.397	0.5252	0.5692
	ru-en	-0.121	0.3633	0.4558
	si-en	0.277	0.2512	0.3111

Table 4: RTM test results in sentence-level MTPP in Task 2 using deepQuest and results combining deepQuest with super learner results.

		MCC	F1 BAD	F1 GOOD	F1 MULTI
Task3	en-cs	0.0508	0.81	0.24	0.1944
	en-de	0.0778	0.7874	0.2634	0.2074
	en-ja	-0.0523	0.1639	0.1418	0.0232
	en-zh	-0.0052	0.6059	0.2401	0.1455

Table 5: RTM test results in sentence-level MTPP in Task 3 using super learner.

References

Ergun Biçici. 2018. RTM results for predicting translation performance. In *Proc. of the Third Conf. on Machine Translation (WMT18)*, pages 765–769, Brussels, Belgium.

Ergun Biçici. 2019. Machine translation with parfda, Moses, kenlm, nplm, and PRO. In *Proc. of the Fourth Conf. on Machine Translation (WMT19)*, pages 122–128, Florence, Italy.

Ergun Biçici. 2020. RTM ensemble learning results at quality estimation task. In *Proc. of the Fifth Conf. on Machine Translation (WMT20)*, Online.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*, 27(3-4):171–192.

Ergun Biçici and Andy Way. 2015. Referential translation machines for predicting semantic similarity. *Language Resources and Evaluation*, pages 1–27.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

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

Sandrine Dudoit and Mark J. van der Laan. 2005. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2(2):131–154.

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.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *51st Annual Meeting of the Assoc. for Comp. Ling.*, pages 690–696, Sofia, Bulgaria.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proc. of the 27th Intl. Conf. on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA.

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 *45th Annual Meeting of the Assoc. for Comp. Ling.*, pages 177–180, Prague, Czech Republic.

Ludmila I. Kuncheva and Juan J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

Michael Perrone and Leon Cooper. 1992. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ. Providence RI Inst. for Brain and Neural Systems.

Eric C. Polley and Mark J. van der Laan. 2010. Super learner in prediction. Technical report, U.C. Berkeley Division of Biostatistics.

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

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proc. of the Sixth Conf. on Machine Translation*, Online. Association for Comp. Ling.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

S. Wold, A. Ruhe, H. Wold, and III Dunn, W. J. 1984. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5:735–743.