RTM Ensemble Learning Results at Quality Estimation Task

Ergun Biçici
ergun.bicici@boun.edu.tr
Electrical and Electronics Engineering Department, Boğaziçi University
orcid.org/0000-0002-2293-2031

Abstract

We obtain new results using referential translation machines (RTMs) with predictions mixed and stacked to obtain a better mixture of experts prediction. We are able to achieve better results than the baseline model in Task 1 subtasks. Our stacking results significantly improve the results on the training sets but decrease the test set results. RTMs can achieve to become the 5th among 13 models in ru-en subtask and 5th in the multilingual track of sentence-level Task 1 based on MAE.

1 Introduction

Quality estimation task in WMT20 (Specia et al., 2020) (QET20) address machine translation (MT) performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence- (Tasks 1 and 2), word- (Task 2), and document-levels (Task 3). Task 1 predicts the sentence-level direct assessment (DA) in 7 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), and
- low-resource, Sinhalese–English (si-en) and Nepalese–English (ne-en).

en-ru contains sentences from both Wikipedia and Reddit articles while others use only Wikipedia sentences with 7000 sentences for training, 1000 for development, and 1000 for testing. 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 (en-de)	8000	1000	bilingual	0.3 M	5 M
Task 1 (en-zh)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (si-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ne-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (et-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ro-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ru-en)	8000	1000	bilingual	0.2 M	4 M
Task 2 (en-de)	8000	1000	bilingual	0.3 M	5 M
Task 2 (en-zh)	8000	1000	monolingual en	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) and binary classification of word-level translation errors. We participated in sentence-level subtasks, which include English-German and English-Chinese in Task 2. 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, 2018; Biçici and Way, 2015) models (M for million).

We tokenize and truecase all of the corpora using Moses' (Koehn et al., 2007) processing tools.[1] 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, data selected close to the task instances in bilingual training settings or monolingual language model (LM) settings. Interpretants provide context for the prediction task and are used during the derivation of the features measuring the closeness of the test sentences to the

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

Proceedings of the 5th Conference on Machine Translation (WMT), pages 999–1003

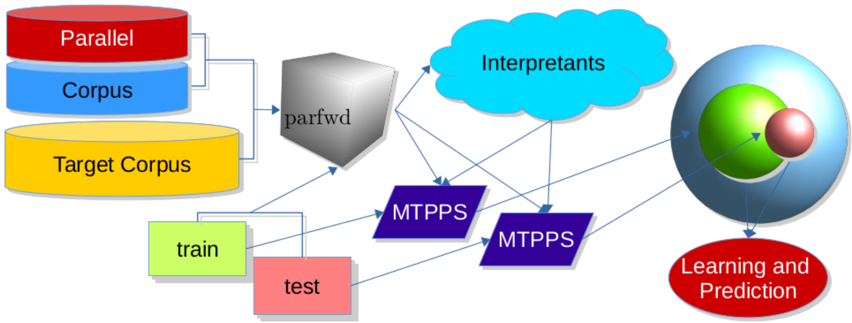

Figure 1: RTM depiction: parfwd selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training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

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, the capability of the interpretant datasets selected to provide context for the training and test sets improve as can be seen in the data statistics of `parfwd` instance selection, parallel feature weight decay (Biçici, 2019). RTMs use `parfwd` for instance selection and machine translation performance prediction system (MTPPS) (Biçici et al., 2013; Biçici and Way, 2015) for obtaining the features, which includes additional features from word alignment. Figure 1 depicts RTMs and explains the model building process.

Additionally, we included the sum, mean, standard deviation, minimum, and maximum of alignment word log probabilities as features in Task 1. In Task 2, we included word alignment displacement features including the average of source and target displacements relative to the length of the source or target sentences respectively and absolute displacement relative to the maximum of source and target sentence lengths.

Instead of resource based discernment, we treated en-de of Tasks 1 and 2 and ru-en as bilingual tasks where significant parallel corpora are available from WMT from previous years and the rest as monolingual, using solely English side of the corpora for deriving MTPP features. In accord, we treat en-de and ru-en as parallel MTPP and the rest as monolingual MTPP. RTM benefits from relevant data selection to be used as interpretants in both monolingual and bilingual settings. The related monolingual or bilingual datasets are used during feature extraction for the machine learning models of MT.

The machine learning models we use include ridge regression (RR), kernel ridge regression, support vector regression (SVR) (Boser et al., 1992), gradient tree boosting, extremely randomized trees (Geurts et al., 2006), and multi-layer perceptron (Bishop, 2006) as learning models 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`.[2] We experiment with:

- including the statistics of the binary tags obtained as features extracted from word-level tag predictions for sentence-level prediction,
- using 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-mix results are in Table 2 achieving 6th rank at best among 15 models in general.

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

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

		r_P	MAE	RMSE
Task 1	en-de	0.2622 (11)	0.5156 (8)	0.6828 (10)
	ru-en	0.6877 (8)	0.5138 (6)	0.6878 (7)
	en-zh	0.2310 (13)	0.5616 (6)	0.7298 (6)
	et-en	0.6067 (11)	0.5995 (8)	0.7284 (8)
	ne-en	0.5436 (11)	0.5308 (9)	0.6828 (9)
	si-en	0.5318 (10)	0.5003 (7)	0.6181 (7)
	ro-en	0.6990 (11)	0.5237 (8)	0.6574 (8)
Task 2	en-de	0.2289 (15)	0.1669 (15)	0.2081 (15)
	en-zh	0.3864 (15)	0.1585 (14)	0.1959 (15)

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

weighted average of the top k predictions, $\hat{y}$ with evaluation metrics indexed by $j \in J$ and weights with w:

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

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 results using r, RAE, MRAER, and MAER. We filter out those results with higher than 0.875 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):

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

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.

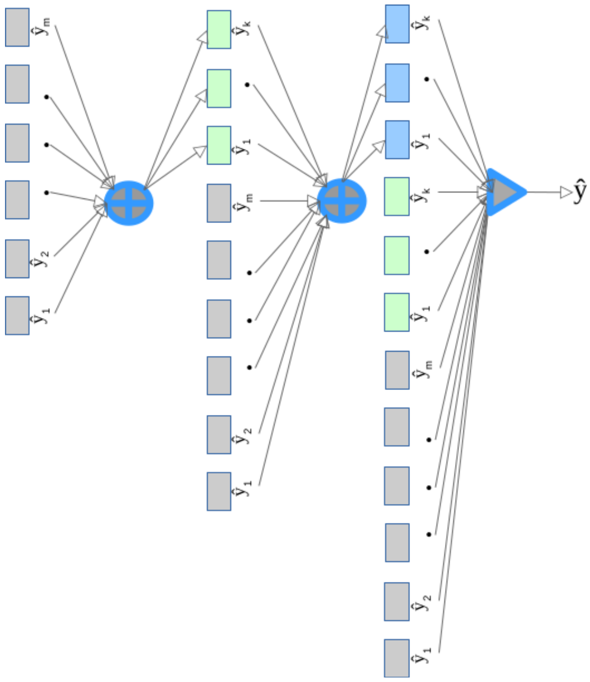

Figure 2: Model combination.

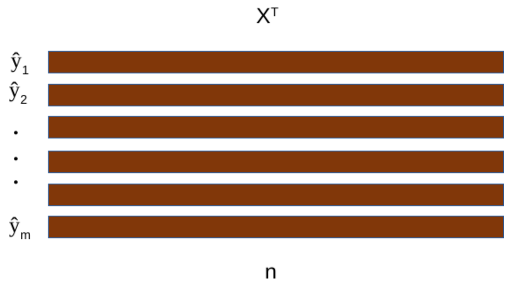

Figure 3: Stacking use predictions as features.

We also use stacking (STACK) to build higher level models using predictions from base prediction models where they can also use the probability associated with the predictions (Ting and Witten, 1999). The stacking models use the predictions from predictors as features and additional selected features and build second level predictors. Stacking with m predictors is depicted in Figure 3 where predictions are used as features for the predictors in the next level. Martins et al. (2017) used a hybrid stacking model to combine the word-level predictions from 15 predictors using neural networks with different initializations together with the previous features from a linear model. Our stacking results also use top features from the data similar to the pass through feature of the stacking regressor of sklearn.[3] For these features, we con-

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html

	r_P trans	GEM mix	STACK
Task 1	en-de	0.2205	0.4244
	en-zh	0.43	0.5426
	et-en	0.5518	0.6245
	ne-en	0.537	0.6182
	si-en	0.4984	0.5907
	ro-en	0.7025	0.7518
	ru-en	0.7245	0.7734
Task 2	en-de	0.4023	0.5153
	en-zh	0.4124	0.5193

Table 3: RTM train results in sentence-level MTPP in tasks 1 and 2. r_P is Pearson's correlation.

		r_P	MAE	RMSE
Task 1	en-de	0.2804 (10)	0.5139 (8)	0.6762 (7)
	ru-en	0.7009 (7)	0.4957 (5)	0.6776 (5)
	en-zh	0.2310 (13)	0.5616 (6)	0.7298 (6)
	et-en	0.6051 (11)	0.5998 (8)	0.7268 (8)
	ne-en	0.6186 (9)	0.4990 (9)	0.6422 (8)
	si-en	0.5493 (10)	0.4909 (6)	0.6055 (6)
	ro-en	0.7367 (10)	0.4967 (7)	0.6167 (7)
	multi	0.5063 (8)	0.5249 (5)	0.6628 (6)
Task 2	en-de	0.2631 (15)	0.1601 (14)	0.1983 (15)
	en-zh	0.4029 (15)	0.1574 (14)	0.1933 (15)

Table 4: RTM test results in sentence-level MTPP in tasks 1 and 2 using the best GEM mix + mix result.

sider at most the top 15% of the features selected with feature selection.

RTM can achieve better results than the baseline model in Task 1 in all tasks participated [4] where the baseline is a neural predictor-estimator approach implemented in OpenKiwi (Kepler et al.). Our training r_P results are in Table 3. Our test set results using GEM mix and MIX are in Table 4 where we obtain 5th rank among 11 submissions in the multilingual subtask according to MAE. Official evaluation metric is r_P.

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 $\geq$ 0.875 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 and stacking achieves better results than MIX on the training set. However, stacking models significantly improve the results on the training data but obtain decreased scores on the test set (Table 5).

[4] Task1: https://competitions.codalab.org/competitions/24447#results,Task2: https://competitions.codalab.org/competitions/24515#results

		r_P	MAE	RMSE
Task 1	en-de	0.2289 (15)	0.6319 (13)	0.7754 (13)
	ru-en	0.6057 (8)	0.7526 (10)	0.9917 (10)
	en-zh	0.1504 (15)	0.8043 (11)	1.0249 (11)
	et-en	0.4014 (13)	1.1209 (13)	1.3892 (13)
	ne-en	0.4856 (13)	0.5662 (10)	0.7688 (10)
	si-en	0.3720 (14)	1.1118 (14)	1.2967 (14)
	ro-en	0.5858 (15)	1.4448 (15)	1.7387 (15)
Task 2	en-de	0.2387 (18)	0.2305 (17)	0.2896 (18)
	en-zh	0.2701 (20)	0.5008 (19)	0.5391 (20)

Table 5: RTM test results in sentence-level MTPP in tasks 1 and 2 using stacking.

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 top performance in automatic, accurate, and language independent prediction of translation scores. We present RTM results with ensemble models and stacking.

Acknowledgments

The research reported here received financial support from the Scientific and Technological Research Council of Turkey (TÜBİTAK) and Boğaziçi University, Turkey.

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)*, Florence, Italy.

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 *Proceedings 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.

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.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T." Martins. OpenKiwi: An open source framework for quality estimation. In *Proc. of the 57th Annual Meeting of the Assoc. for Computational Linguistics: System Demonstrations", month = 7, year = 2019, address = Florence, Italy, publisher = Assoc. for Computational Linguistics, pages = 117–122,*.

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.

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

André F.T. Martins, Marcin Junczys-Dowmunt, Fabio N. Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Comp. Ling.*, 5:205–218.

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.

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, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proc. of the Fifth Conf. on Machine Translation: Shared Task Papers*, Online.

Kai Ming Ting and Ian H. Witten. 1999. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289.

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.