RTM Stacking Results for Machine Translation Performance Prediction

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 with increased number of learning models in the set of results that are stacked to obtain a better mixture of experts prediction. We combine features extracted from the word-level predictions with the sentence- or document-level features, which significantly improve the results on the training sets but decrease the test set results.

1 Referential Translation Machines for Machine Translation Performance Predicion

Quality estimation task in WMT19 (Specia et al., 2019) (QET19) address machine translation performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence- and word- (Task 1), and document-levels (Task 2). The tasks contain subtasks involving English-German, English-Russian, and English-French machine translation (MT). The target to predict in Task 1 is HTER (human-targeted translation edit rate) scores (Snover et al., 2006) and binary classification of word-level translation errors and the target in Task 2 is multi-dimensional quality metrics (MQM) (Lommel, 2015). 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 RTM models (M for million).

We use referential translation machine (RTM) (Biçici, 2018; Biçici and Way, 2015) models for building our prediction models. RTMs predict data translation between the instances in the training set and the test set using interpretants, data close to the task instances. 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 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 by RTM models to provide context for the training and test sets improve as can be seen in the data statistics of `parfda` instance selection (Biçici, 2019). Figure 1 depicts RTMs and explains the model building process. RTMs use `parfda` for instance selection and machine translation performance prediction system (MTPPS) for obtaining the features, which includes additional features from word alignment and also from GLMd for word-level prediction.

			RTM interpretants	
Task	Train	Test	Training	LM
Task 1 (en-de)	14442	1000		
Task 1 (en-ru)	16089	1000	0.250M	5M
Task 2 (en-fr)	1468	180		

Table 1: Number of instances and interpretants used.

We use ridge regression, kernel ridge regression, k-nearest neighors, support vector regression, AdaBoost (Freund and Schapire, 1997), gradient tree boosting, gaussian process regressor, 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`.[1] We experiment with:

- including the statistics of the binary tags obtained as features extracted from word-level tag predictions for sentence-level prediction,
- using KNN to estimate the noise level for

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

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 3: Shared Task Papers (Day 2) pages 75–79

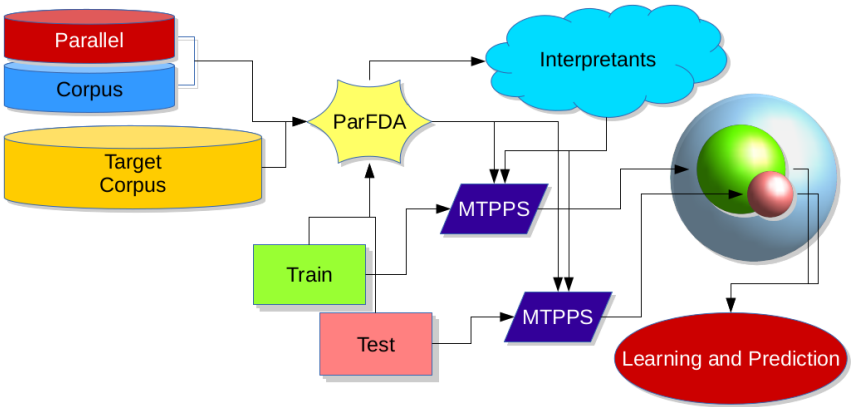

Figure 1: RTM depiction: parfda selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training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

SVR, which obtains accuracy with 5% error compared with estimates obtained with known noise level (Cherkassky and Ma, 2004) and set $\epsilon = \sigma/2$.

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. The neural network architecture they used is also hybrid with different types of layers: input word embedding use 64 dimensional vectors, the next three layers are two feedforward layers with 400 nodes and a bidirectional gated recurrent units layer with 200 units, followed by similar three layers with half nodes, followed by a feedforward layer with 50 nodes and a softmax layer.

We use Global Linear Models (GLM) (Collins, 2002) with dynamic learning (GLMd) (Biçici, 2018) for word- and phrase-level translation performance prediction. GLMd uses weights in a range $[a, b]$ to update the learning rate dynamically according to the error rate. Evaluation metrics listed are Pearson's correlation (r), mean absolute error (MAE), and root mean squared error (RMSE).

2 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:

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

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 only use the MIX prediction if 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 1 relative evaluation metric scores.

We also use stacking 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 build second level predictors.

For the document-level RTM model, instead of running separate MTPPS instances for each training or test document to obtain specific features for each document, we concatenate the sentences from each document to obtain a single sentence representing each and then run an RTM model. This conversion decreases the number of features and obtains close results (Biçici, 2018).

Before model combination, we further filter prediction results from different machine learn-

				r_P	MAE	RAE	MAER	MRAER
2019	sentence	en-de		0.4908	0.1102	0.8017	0.8721	0.7554
			+word tags	0.9608	0.0237	0.1725	0.1388	0.1823
		en-ru		0.2724	0.1548	0.8769	0.9064	0.7736
			+word tags	0.9481	0.028	0.1587	0.1541	0.1553
	document	en-fr		0.3959	17.982	0.8564	0.933	0.7908
			+word tags	0.478	17.1015	0.8144	0.8921	0.7564
2018	sentence	en-de SMT		0.4386	0.1368	0.8675	0.9103	0.8168
			+word tags	0.9424	0.0391	0.248	0.1716	0.2969
		en-de NMT		0.4613	0.1109	0.8066	0.8414	0.7347
			+word tags	0.9589	0.0244	0.1777	0.144	0.1901
		de-en SMT		0.5636	0.1355	0.7903	0.9173	0.7826
			+word tags	0.9276	0.0485	0.2828	0.2413	0.3378
		en-cs SMT		0.5397	0.1506	0.8084	0.8203	0.7886
			+word tags	0.9356	0.0477	0.256	0.1825	0.3021
		en-lv SMT		0.4006	0.1329	0.8832	0.9316	0.8059
			+word tags	0.9452	0.0342	0.2271	0.1768	0.2625
		en-lv NMT		0.5779	0.1441	0.7831	0.8679	0.7768
			+word tags	0.9571	0.0398	0.2163	0.1778	0.2573
	document	en-fr		0.2141	40.7359	0.9324	1.2074	0.7573
			+word tags	0.2254	41.6591	0.9535	1.0849	0.7783

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

ing 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 to include results that are better than the best RR model's results. In general, the combined model is better than the best model in the set and stacking achieves better results than MIX.

3 Results

We tokenize and truecase all of the corpora using Moses' (Koehn et al., 2007) processing tools.[2] LMs are built using `kenlm` (Heafield et al., 2013). The comparison of results on the training set are in Table 2 and the results on the test set we obtained after the competition are in Tables 3 and 5. Official competition results of RTMs are similar.

We convert MQM annotation to word-level tags to train GLMd models and obtain word-level predictions. Addition of the tagging features from the word-level prediction improves the training results significantly but does not improve the test results at the same rate, which indicates overfitting. The reason for the overfitting with the word-level features is due to their high correlation with the target. Table 4 lists some of the top individual feature correlations for en-ru in Task1. Top 26 highly correlated features belong to word-level features.

We also obtained new results on QET18 datasets and experimented adding features from word-level predictions on the QET18 sentence-level results. QET18 results in Table 3 are improved overall.

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 stacking.

Acknowledgments

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

References

Ergun Biçici. 2018. RTM results for predicting translation performance. In *Third Conf. on Statistical Machine Translation (WMT18)*, Brussels, Belgium.

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

				r_P	MAE	RAE	MAER	MRAER
2019	sentence	en-de		0.4617	0.1176	0.8066	0.7755	0.7338
			+word tags	0.1842	0.1612	1.1056	1.1472	1.1771
		en-ru		0.269	0.187	0.8468	0.7827	0.7232
			+word tags	0.2423	**0.1868**	**0.8461**	0.7919	0.7681
	document	en-fr		0.3064	21.6283	0.9044	1.3233	0.8565
			+word tags	0.2162	22.2011	0.9283	**1.2688**	0.8861
				r_P	r_S	MAE	RMSE	
2018	sentence	en-de SMT		0.4165 (11)	0.4236 (9)	0.1368 (10)	0.1734 (10)	
			+word tags	0.2689 (16)	0.2780 (12)	0.1659 (15)	0.2192 (15)	
			top	0.7397	0.7543	0.0937	0.1362	
		en-de NMT		0.4752 (3)	0.5556 (4)	0.1173 (3)	0.1753 (5)	
			+word tags	0.1645 (16)	0.3752 (10)	0.1501 (11)	0.2239 (14)	
			top	0.5129	0.6052	0.1114	0.1719	
		de-en SMT		0.5773 (9)	0.5144 (8)	0.1326 (10)	0.1687 (9)	
			+word tags	0.3936 (12)	0.3530 (9)	0.1603 (13)	0.2155 (13)	
			top	0.7667	0.7318	0.0945	0.1315	
		en-cs SMT		0.5007 (6)	0.5037 (5)	0.1544 (6)	0.1988 (6)	
			+word tags	0.4469 (8)	0.4384 (7)	0.1775 (11)	0.2331 (11)	
			top	0.6918	0.7105	0.1223	0.1693	
		en-lv SMT		0.3560 (7)	0.2884 (8)	0.1395 (5)	0.1867 (4)	
			+word tags	0.3097 (10)	0.2578 (8)	0.1598 (6)	0.2155 (8)	
			top	0.6188	0.5766	0.1202	0.1602	
		en-lv NMT		0.5394 (4)	0.4963 (4)	0.1533 (2)	0.2009 (2)	
			+word tags	0.4132 (7)	0.4007 (7)	0.1841 (7)	0.2466 (8)	
			top	0.6819	0.6665	0.1308	0.1747	
	document	en-fr		0.0068 (4)		58.4664 (4)	88.1198 (4)	
			+word tags	**0.0112** (4)		**58.0524** (4)	**86.3416** (4)	
			top	0.5337		56.2264	85.2319	

Table 3: RTM stacking results on the test set where **bold** indicate results that improve with the addition of features from word-level predictions. (#) indicates the rank. r_S is Spearman's correlation.

r_P train	r_P test	feature
0.937	0.2369	avg number of 1s in tags
0.5941	0.1838	std of the number of 1s in tags
0.0773	0.055	translation average BLEU

Table 4: Word-level prediction features are highly correlated with the target in the training set for en-ru in Task1.

Ergun Biçici. 2019. Machine translation with `parfda`, moses, `kenlm`, `nplm`, and pro. In *Fourth Conf. on Statistical Machine Translation (WMT19)*, Florence, Italy.

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*.

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

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *ACL-02 Conf. on Empirical methods in natural language processing*, EMNLP '02, pages 1–8, Stroudsburg, PA, USA.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

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.

		model	r_P	r_S	MAE	RMSE
sentence	en-de		0.4617	0.5279	0.1176	0.1757
		+word tags	0.1842	0.3308	0.1612	0.2334
		top	0.5718	0.6221		
	en-ru		0.2690	0.2677	0.187	0.2827
		+word tags	0.2423	0.1474	0.1868	0.3048
		top	0.5923	0.5388		
document	en-fr		0.3065	0.3642	21.6282	26.1010
		+word tags	0.2162	0.2460	22.2011	27.0249
		top	0.3744			

Table 5: RTM test results and the top result.

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 Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

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 Computational Linguistics Companion Volume Demo and Poster Sessions*, 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.

Arle Lommel. 2015. Multidimensional quality metrics (mqm) definition. *URL http://www. qt21. eu/mqm-definition/definition-2015-12-30. html*.

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 Computational Linguistics*, 5:205–218.

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, Varvara Logacheva, Ramón F. Astudillo, and André Martins. 2019. Findings of the wmt 2019 shared task on quality estimation. In *Fourth Conf. on Machine Translation*, Florence, Italy.

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.