RegMT System for Machine Translation, System Combination, and Evaluation

Ergun Biçici
Koç University
34450 Sariyer, Istanbul, Turkey
ebicici@ku.edu.tr

Deniz Yuret
Koç University
34450 Sariyer, Istanbul, Turkey
dyuret@ku.edu.tr

Abstract

We present the results we obtain using our RegMT system, which uses transductive regression techniques to learn mappings between source and target features of given parallel corpora and use these mappings to generate machine translation outputs. Our training instance selection methods perform feature decay for proper selection of training instances, which plays an important role to learn correct feature mappings. RegMT uses L_2 regularized regression as well as L_1 regularized regression for sparse regression estimation of target features. We present translation results using our training instance selection methods, translation results using graph decoding, system combination results with RegMT, and performance evaluation with the F_1 measure over target features as a metric for evaluating translation quality.

1 Introduction

Regression can be used to find mappings between the source and target feature sets derived from given parallel corpora. Transduction learning uses a subset of the training examples that are closely related to the test set without using the model induced by the full training set. In the context of statistical machine translation, translations are performed at the sentence level and this enables us to select a small number of training instances for each test instance to guide the translation process. This also gives us a computational advantage when considering the high dimensionality of the problem as each sentence can be mapped to many features.

The goal in transductive regression based machine translation (RegMT) is both reducing the computational burden of the regression approach by reducing the dimensionality of the training set and the feature set and also improving the translation quality by using transduction.

We present translation results using our training instance selection methods, translation results using graph decoding, system combination results with RegMT, and performance evaluation with the F_1 measure over target features as a metric for evaluating translation quality. RegMT work builds on our previous regression-based machine translation results (Bicici and Yuret, 2010) especially with instance selection and additional graph decoding capability. We present our results to this year's challenges.

Outline: Section 2 gives an overview of the RegMT model. In section 3, we present our training instance selection techniques and WMT'11 results. In section 4, we present the graph decoding results on the Haitian Creole-English translation task. Section 5 presents our system combination results using reranking with the RegMT score. Section 6 evaluates the F_1 measure that we use for the automatic evaluation metrics challenge. The last section present our contributions.

2 Machine Translation Using Regression

Let X and Y correspond to the sets of tokens that can be used in the source and target strings, then, m training instances are represented as $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m) \in X^* \times Y^*$, where $(\mathbf{x}_i, \mathbf{y}_i)$ corresponds to a pair of source and target language

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 323–329,
Edinburgh, Scotland, UK, July 30–31, 2011.

token sequences for $1 \leq i \leq m$. Our goal is to find a mapping $f : X^* \rightarrow Y^*$ that can convert a source sentence to a target sentence sharing the same meaning in the target language (Figure 1).

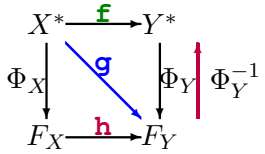

Figure 1: String-to-string mapping.

We define feature mappers $\Phi_X : X^* \rightarrow F_X = \mathbb{R}^{N_X}$ and $\Phi_Y : Y^* \rightarrow F_Y = \mathbb{R}^{N_Y}$ that map each string sequence to a point in high dimensional real number space. Let $\mathbf{M}_X \in \mathbb{R}^{N_X \times m}$ and $\mathbf{M}_Y \in \mathbb{R}^{N_Y \times m}$ such that $\mathbf{M}_X = [\Phi_X(\mathbf{x}_1), \ldots, \Phi_X(\mathbf{x}_m)]$ and $\mathbf{M}_Y = [\Phi_Y(\mathbf{y}_1), \ldots, \Phi_Y(\mathbf{y}_m)]$. The ridge regression solution using L_2 regularization is found by minimizing the following cost:

$$\mathbf{W}_{L_2} = \underset{\mathbf{W} \in \mathbb{R}^{N_Y \times N_X}}{\arg\min} \| \mathbf{M}_Y - \mathbf{W}\mathbf{M}_X \|_F^2 + \lambda \| \mathbf{W} \|_F^2 . \quad (1)$$

Two main challenges of the regression based machine translation (RegMT) approach are learning the regression function, $h : F_X \rightarrow F_Y$, and solving the *pre-image problem*, which, given the features of the estimated target string sequence, $h(\Phi_X(\mathbf{x})) = \Phi_Y(\hat{\mathbf{y}})$, attempts to find $\mathbf{y} \in Y^*$: $\mathbf{y} = \arg\min_{\mathbf{y} \in Y^*} ||h(\Phi_X(\mathbf{x})) - \Phi_Y(\mathbf{y})||^2$. Pre-image calculation involves a search over possible translations minimizing the cost function:

$$f(\mathbf{x}) = \underset{\mathbf{y} \in Y^*}{\arg\min} \| \Phi_Y(\mathbf{y}) - \mathbf{W}\Phi_X(\mathbf{x}) \|^2 . \quad (2)$$

2.1 L_1 Regularized Regression

String kernels lead to sparse feature representations and L_1 regularized regression is effective to find the mappings between sparsely observed features.We would like to observe only a few nonzero target coefficients corresponding to a source feature in the coefficient matrix. L_1 regularization helps us achieve solutions close to permutation matrices by increasing sparsity (Bishop, 2006) (page 145). In contrast, L_2 regularized solutions give us dense matrices.

$\mathbf{W}_{L_2}$ is not a sparse solution and most of the coefficients remain non-zero. We are interested in penalizing the coefficients better; zeroing the irrelevant ones leading to sparsification to obtain a solution that is closer to a permutation matrix. L_1 norm behaves both as a feature selection technique and a method for reducing coefficient values.

$$\mathbf{W}_{L_1} = \underset{\mathbf{W} \in \mathbb{R}^{N_Y \times N_X}}{\arg\min} \| \mathbf{M}_Y - \mathbf{W}\mathbf{M}_X \|_F^2 + \lambda \| \mathbf{W} \|_1 . \quad (3)$$

Equation 3 presents the *lasso* (Tibshirani, 1996) solution where the regularization term is now the L_1 matrix norm defined as $\|\mathbf{W}\|_1 = \sum_{i,j} |W_{i,j}|$. $\mathbf{W}_{L_2}$ can be found by taking the derivative but since L_1 regularization cost is not differentiable, $\mathbf{W}_{L_1}$ is found by optimization or approximation techniques. We use forward stagewise regression (FSR) (Hastie et al., 2006), which approximates *lasso* for L_1 regularized regression.

2.2 Related Work:

Regression techniques can be used to model the relationship between strings (Cortes et al., 2007). Wang et al. (2007) applies a string-to-string mapping approach to machine translation by using ordinary least squares regression and n-gram string kernels to a small dataset. Later they use L_2 regularized least squares regression (Wang and Shawe-Taylor, 2008). Although the translation quality they achieve is not better than Moses (Koehn et al., 2007), which is accepted to be the state-of-the-art, they show the feasibility of the approach. Serrano et al. (2009) use kernel regression to find translation mappings from source to target feature vectors and experiment with translating hotel front desk requests. Locally weighted regression solves separate weighted least squares problems for each instance (Hastie et al., 2009), weighted by a kernel similarity function.

3 Instance Selection for Machine Translation

Proper selection of training instances plays an important role for accurately learning feature mappings with limited computational resources. Coverage of the features is important since if we do not have the correct features in the training matrices, we will not be able to translate them. Coverage is measured by the percentage of target features of the test set found in the training set. For each test sentence, we pick a limited number of training instances designed to

improve the coverage of correct features to build a regression model.

We use two techniques for this purpose: (1) Feature Decay Algorithm (FDA), which optimizes source languge bigram coverage to maximize the target coverage, (2) *dice*. Feature decay algorithms (FDA) aim to maximize the coverage of the target language features (such as words, bigrams, and phrases) for the test sentences. FDA selects training instances one by one updating the coverage of the features already added to the training set in contrast to the features found in the test sentence.

We also use a technique that we call *dice*, which optimizes source language bigram coverage such that the difficulty of aligning source and target features is minimized. We define Dice's coefficient score as:

$$dice(x,y) = \frac{2C(x,y)}{C(x)C(y)}, \quad (4)$$

where $C(x,y)$ is the number of times x and y cooccurr and $C(x)$ is the count of observing x in the selected training set. Given a test source sentence, $S_{\mathcal{U}}$, we can estimate the goodness of a training sentence pair, (S,T), by the sum of the alignment scores:

$$\phi_{dice}(S_{\mathcal{U}},S,T) = \frac{\sum_{x\in X(S_{\mathcal{U}})}\sum_{j=1}^{|T|}\sum_{y\in Y(x)} dice(y,T_j)}{|T|\log|S|}, \quad (5)$$

where $X(S_{\mathcal{U}})$ stores the features of $S_{\mathcal{U}}$ and $Y(x)$ lists the tokens in feature x. The difficulty of word aligning a pair of training sentences, (S,T), can be approximated by $|S|^{|T|}$. We use a normalization factor proportional to $|T|\log|S|$.

The details of both of these techniques and further results can be found in (Bicici and Yuret, 2011).

3.1 Moses Experiments on the Translation Task

We have used FDA and *dice* algorithms to select training sets for the out-of-domain challenge test sets used in (Callison-Burch et al., 2011). The parallel corpus contains about 1.9 million training sentences and the test set contain 3003 sentences. We built separate Moses systems using all of the parallel corpus for the language pairs *en-de*, *de-en*, *en-es*, and *es-en*. We created training sets using all of the features of the test set to select training instances. The results given in Table 1 show that we can achieve similar BLEU performance using about 7% of the parallel corpus target words (200,000 instances) using *dice* and about 16% using FDA. In the out-of-domain translation task, we are able to reduce the training set size to achieve a performance close to the baseline. We may be able to achieve better performance in this out-of-domain task as well as explained in (Bicici and Yuret, 2011).

		en-de	*de-en*	*en-es*	*es-en*
BLEU	ALL	.1376	.2074	.2829	.2919
	FDA	.1363	.2055	.2824	.2892
	dice	.1374	.2061	.2834	.2857
words	ALL	47.4	49.6	52.8	50.4
	FDA	7.9	8.0	8.7	8.2
	dice	6.9	7.0	3.9	3.6
% ALL	FDA	17	16	16	16
	dice	14	14	7.4	7.1

Table 1: Performance for the out-of-domain task of (Callison-Burch et al., 2011). ALL corresponds to the baseline system using all of the parallel corpus. words list the size of the target words used in millions.

4 Graph Decoding for RegMT

We perform graph-based decoding by first generating a De Bruijn graph from the estimated $\hat{\mathbf{y}}$ (Cortes et al., 2007) and then finding Eulerian paths with maximum path weight. We use four features when scoring paths: (1) estimation weight from regression, (2) language model score, (3) brevity penalty as found by $e^{\alpha(l_R-|s|/|path|)}$ for l_R representing the length ratio from the parallel corpus and $|path|$ representing the length of the current path, (4) future cost as in Moses (Koehn et al., 2007) and weights are tuned using MERT (Och, 2003) on the *de-en dev* set.

We demonstrate that sparse L_1 regularized regression performs better than L_2 regularized regression. Graph based decoding can provide an alternative to state of the art phrase-based decoding system Moses in translation domains with small vocabulary and training set size.

4.1 Haitian Creole to English Translation Task with RegMT

We have trained a Moses system for the Haitian Creole to English translation task, cleaned corpus, us-

ing the options as described in section 3.1. Moses achieves 0.3186 BLEU on this task. We observed that graph decoding performs better where target coverage is high such that the bigrams used lead to a connected graph. To increase the connectivity, we have included Moses translations in the training set and performed graph decoding with RegMT. RegMT with L_2 regularized regression achieves 0.2708 BLEU with graph decoding and *lasso* achieves 0.26 BLEU.

Moses makes use of a number of distortion parameters and lexical weights, which are estimated using all of the parallel corpus. Thus, our Moses translation achieves a better performance than graph decoding with RegMT using 100 training instances for translating each source test sentence.

5 System Combination with RegMT

We perform experiments on the system combination task for the English-German, German-English, English-Spanish, and Spanish-English language pairs using the training corpus provided in WMT'11 (Callison-Burch et al., 2011). We have tokenized and lowercased each of the system outputs and combined these in a single N-best file per language pair. We use these N-best lists for reranking by RegMT to select the best translation model. Feature mappers used are 2-spectrum counting word kernels (Taylor and Cristianini, 2004).

We rerank N-best lists by a linear combination of the following scoring functions:

1. RegMT: Regression based machine translation scores as found by Equation 2.

2. CBLEU: Comparative BLEU scores we obtain by measuring the average BLEU performance of each translation relative to the other systems' translations in the N-best list.

3. LM: We calculate 5-gram language model scores for each translation using the language model trained over the target corpus provided in the translation task.

Since we do not have access to the reference translations nor to the translation model scores each system obtained for each sentence, we estimate translation model performance (CBLEU) by measuring the average BLEU performance of each translation relative to the other translations in the N-best list. Thus, each possible translation in the N-best list is BLEU scored against other translations and the average of these scores is selected as the CBLEU score for the sentence. Sentence level BLEU score calculation avoids singularities in n-gram precisions by taking the maximum of the match count and $\frac{1}{2|s_i|}$ for $|s_i|$ denoting the length of the source sentence s_i as used in (Macherey and Och, 2007).

Table 2 presents reranking results on all of the language pairs we considered, using RegMT, CBLEU, and LM scores with the same combination weights as above. We also list the performance of the best model (Max) as well as the worst (Min). We are able to achieve close or better BLEU scores in all of the listed systems when compared with the performance of the best translation system except for the *ht-en* language pair. The lower performance in the *ht-en* language pair may be due to having a single best translation system that outperforms others significantly. This happens for instance when an unconstrained model use external resources to achieve a significantly better performance than the second best model. 2^{nd} best in Table 2 lists the second best model's performance to estimate how much the best model's performance is better than the rest.

BLEU	*en-de*	*de-en*	*en-es*	*es-en*	*ht-en*
Min	.1064	.1572	.2174	.1976	.2281
Max	.1727	.2413	.3375	.3009	.3708
2^{nd} best	.1572	.2302	.3301	.2973	.3288
Average	.1416	.1997	.292	.2579	.2993
Oracle	.2529	.3305	.4265	.4233	.4336
RegMT	.1631	.2322	.3311	.3052	.3234

Table 2: System combination results.

RegMT model may prefer sentences with lower BLEU, which can sometimes cause it to achieve a lower BLEU performance than the best model. This is clearly the case for *en-de* with 1.6 BLEU points difference with the second best model performance and for *de-en* task with 1.11 BLEU points difference. Also this observation holds for *en-es* with 0.74 BLEU points difference and for *ht-en* with 4.2 BLEU points difference. For *es-en* task, there is 0.36 BLEU points difference with the second best model and these models likely to complement each other.

The existence of complementing SMT models is important for the reranking approach to achieve a performance better than the best model, as there is a need for the existence of a model performing better than the best model on some test sentences. We can use the competitive SMT model to achieve the performance of the best with a guarantee even when a single model is dominating the rest (Bicici and Kozat, 2010). For competing translation systems in an on-line machine translation setting adaptively learning of model weights can be performed based on the previous transaltion performance (Bicici and Kozat, 2010).

6 Target F_1 as a Performance Evaluation Metric

We use target sentence F_1 measure over the target features as a translation performance evaluation metric. We optimize the parameters of the RegMT model with the F_1 measure comparing the target vector with the estimate we get from the RegMT model. F_1 measure uses the 0/1-class predictions over the target feature with the estimate vector, $\Phi_Y(\hat{\mathbf{y}})$. Let TP be the true positive, TN the true negative, FP the false positive, and FN the false negative rates, we use the following measures for evaluation:

$$\text{prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{BER} = (\frac{\text{FP}}{\text{TN+FP}} + \frac{\text{FN}}{\text{TP+FN}})/2 \quad (6)$$

$$\text{rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F_1 = \frac{2\times\text{prec}\times\text{rec}}{\text{prec+rec}} \quad (7)$$

where BER is the balanced error rate, prec is precision, and rec is recall. The evaluation techniques measure the effectiveness of the learning models in identifying the features of the target sentence making minimal error to increase the performance of the decoder and its translation quality.

We use gapped word sequence kernels (Taylor and Cristianini, 2004) when using F_1 for evaluating translations since a given translation system may not be able to translate a given word but can correctly identify the surrounding phrase. For instance, let the reference translation be the following sentence:

```
a sound compromise has been reached
```

Some possible translations for the reference are given in Table 3 together with their BLEU (Papineni et al., 2001) and F_1 scores for comparison. F_1 score does not have a brevity penalty but a brief translation is penalized by a low recall value. We use up to 3 tokens as gaps. F_1 measure is able to increase the ranking of Trans_4 by using a gapped sequence kernel, which can be preferrable to Trans_3.

We note that a missing token corresponds to varying decreases in the n-gram precision used in the BLEU score. A sentence containing m tokens has m 1-grams, $m-1$ 2-grams, and $m-n+1$ n-grams. A missing token degrades the performance more in higher order n-gram precision values. A missing token decreases n-gram precision by $\frac{1}{m}$ for 1-grams and by $\frac{n}{m-n+1}$ for n-grams. Based on this observation, we use F_1 measure with gapped word sequence kernels to evaluate translations. Gapped features allows us to consider the surrounding phrase for a missing token as present in the translation.

Let the reference sentence be represented with `a b c d e f` where a-f, x, y, z correspond to tokens in the sentence. Then, Trans_3 has the form `a b x y f`, and Trans_4 has the form `a c y f`. Then, F_1 ranks Trans_4 higher than Trans_3 for orders greater than 3 as there are two consecutive word errors in Trans_3. F_1 can also prefer a missing token rather than a word error as we see by comparing Trans_4 and Trans_5 and it can still prefer contiguity over a gapped sequence as we see by comparing Trans_5 and Trans_6 in Table 3.

We calculate the correlation of F_1 with BLEU on the *en-de* development set. We use 5-grams with the F_1 measure as this increases the correlation with 4-gram BLEU. Table 4 gives the correlation results using both Pearson's correlation score and Spearman's correlation score. Spearman's correlation score is a better metric for comparing the relative orderings.

Metric	No gaps	Gaps
Pearson	.8793	.7879
Spearman	.9068	.8144

Table 4: F_1 correlation with 4-gram BLEU using blended 5-gram gapped word sequence features on the development set.

7 Contributions

We present the results we obtain using our RegMT system, which uses transductive regression techniques to learn mappings between source and tar-

	Format	BLEU	F_1		
Ref: a sound compromise has been reached	a b c d e f	4-grams	3-grams	4-grams	5-grams
Trans_1: a sound agreement has been reached	a b x d e f	.2427	.6111	.5417	.5
Trans_2: a compromise has reached	a c d f	.137	.44	.3492	.3188
Trans_3: a sound agreement is reached	a b x y f	.1029	.2	.1558	.1429
Trans_4: a compromise is reached	a c y f	.0758	.2	.1587	.1449
Trans_5: a good compromise is reached	a z c y f	.0579	.1667	.1299	.119
Trans_6: a good compromise is been	a z c y e	.0579	.2	.1558	.1429

Table 3: BLEU vs. F_1 on sample sentence translation task.

get features of given parallel corpora and use these mappings to generate machine translation outputs. We also present translation results using our training instance selection methods, translation results using graph decoding, system combination results with RegMT, and performance evaluation with F_1 measure over target features. RegMT work builds on our previous regression-based machine translation results (Bicici and Yuret, 2010) especially with instance selection and additional graph decoding capability.

References

Ergun Bicici and S. Serdar Kozat. 2010. Adaptive model weighting and transductive regression for predicting best system combinations. In *Proceedings of the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July. Association for Computational Linguistics.

Ergun Bicici and Deniz Yuret. 2010. L_1 regularized regression for reranking and system combination in machine translation. In *Proceedings of the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July. Association for Computational Linguistics.

Ergun Bicici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, Edinburgh, England, July.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. 2011. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, England, July.

Corinna Cortes, Mehryar Mohri, and Jason Weston. 2007. A general regression framework for learning string-to-string mappings. In Gokhan H. Bakir, Thomas Hofmann, and Bernhard Sch editors, *Predicting Structured Data*, pages 143–168. The MIT Press, September.

Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther. 2006. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, 2nd edition.

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 *Annual Meeting of the Assoc. for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June.

Wolfgang Macherey and Franz Josef Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 986–995, Prague, Czech Republic, June. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. *Association for Computational Linguistics*, 1:160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Nicolas Serrano, Jesus Andres-Ferrer, and Francisco Casacuberta. 2009. On a kernel regression approach to machine translation. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 394–401.

J. Shawe Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Robert J. Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288.

Zhuoran Wang and John Shawe-Taylor. 2008. Kernel regression framework for machine translation: UCL system description for WMT 2008 shared translation task. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 155–158, Columbus, Ohio, June. Association for Computational Linguistics.

Zhuoran Wang, John Shawe-Taylor, and Sandor Szedmak. 2007. Kernel regression based machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 185–188, Rochester, New York, April. Association for Computational Linguistics.