Kernel Regression Based Machine Translation

Zhuoran Wang and **John Shawe-Taylor**
Department of Computer Science
University College London
London, WC1E 6BT
United Kingdom
{z.wang, jst}@cs.ucl.ac.uk

Sandor Szedmak
School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ
United Kingdom
ss03v@ecs.soton.ac.uk

Abstract

We present a novel machine translation framework based on kernel regression techniques. In our model, the translation task is viewed as a string-to-string mapping, for which a regression type learning is employed with both the source and the target sentences embedded into their kernel induced feature spaces. We report the experiments on a French-English translation task showing encouraging results.

1 Introduction

Fig. 1 illustrates an example of phrase alignment for statistical machine translation (SMT). A rough linear relation is shown by the co-occurences of phrases in bilingual sentence pairs, which motivates us to introduce a novel study on the SMT task:

If we define the feature space $\mathcal{H}_x$ of our source language $\mathcal{X}$ as all its possible phrases (i.e. informative blended word *n*-grams), and define the mapping $\Phi_x : \mathcal{X} \to \mathcal{H}_x$, then a sentence $x \in \mathcal{X}$ can be expressed by its feature vector $\Phi_x(x) \in \mathcal{H}_x$. Each component of $\Phi_x(x)$ is indexed by a phrase with the value being the frequency of it in x. The definition of the feature space $\mathcal{H}_y$ of our target language $\mathcal{Y}$ can be made in a similar way, with corresponding mapping $\Phi_y : \mathcal{Y} \to \mathcal{H}_y$. Now in the machine translation task, given $S = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, \ldots, m\}$, a set of sample sentence pairs where y_i is the translation of x_i, we are trying to learn $\mathbf{W}$ a matrix represented linear operator, such that:

$$\Phi_y(y) = f(x) = \mathbf{W}\Phi_x(x) \quad (1)$$

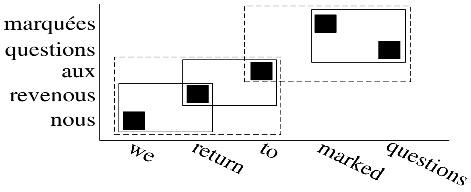

Figure 1: Phrase alignment in SMT

to predict the translation y for a new sentence x.

Comparing with traditional methods, this model gives us a theoretical framework to capture higher-dimensional dependencies within the sentences. To solve the multi-output regression problem, we investigate two models, least squares regression (LSR) similar to the technique presented in (Cortes et al., 2005), and maximum margin regression (MMR) introduced in (Szedmak et al., 2006).

The rest of the paper is organized as follows. Section 2 gives a brief review of the regression models. Section 3 details the solution to the pre-image problem. We report the experimental results in Section 4, with discussions in Section 5.

2 Kernel Regression with Vector Outputs

2.1 Kernel Induced Feature Space

In the practical learning process, only the inner products of the feature vectors are needed (see Section 2.2, 2.3 and 3), so we can perform the so-called kernel trick to avoid dealing with the very high-dimensional feature vectors explicitly. That is, for $x, z \in \mathcal{X}$, a kernel function is defined as:

$$\kappa_x(x, z) = \langle \Phi_x(x), \Phi_x(z) \rangle = \Phi_x(x)^\top \Phi_x(z) \quad (2)$$

Proceedings of NAACL HLT 2007, Companion Volume, pages 185–188,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

Similarly, a kernel function $\kappa_y(\cdot, \cdot)$ is defined in $\mathcal{H}_y$.

In our case, the blended *n*-spectrum string kernel (Lodhi et al., 2002) that compares two strings by counting how many (contiguous) substrings of length from 1 up to *n* they have in common, is a good choice for the kernel function to induce our feature spaces $\mathcal{H}_x$ and $\mathcal{H}_y$ implicitly, even though it brings in some uninformative features (word n-grams) as well, when compared to our original definition.

2.2 Least Squares Regression

A basic method to solve the problem in Eq. 1 is least squares regression that seeks the matrix $\mathbf{W}$ minimizing the squared loss in $\mathcal{H}_y$ on the training set S:

$$\min \|\mathbf{W}\mathbf{M}_x - \mathbf{M}_y\|_F^2 \qquad (3)$$

where $\mathbf{M}_x = [\Phi_x(x_1), ..., \Phi_x(x_m)]$, $\mathbf{M}_y = [\Phi_y(y_1), ..., \Phi_y(y_m)]$, and $\| \cdot \|_F$ denotes the Frobenius norm.

Differentiating the expression and setting it to zero gives:

$$\begin{aligned} & 2\mathbf{W}\mathbf{M}_x\mathbf{M}_x^\top - 2\mathbf{M}_y\mathbf{M}_x^\top = 0 \\ \Rightarrow \quad & \mathbf{W} = \mathbf{M}_y\mathbf{K}_x^{-1}\mathbf{M}_x^\top \end{aligned} \qquad (4)$$

where $\mathbf{K}_x = \mathbf{M}_x^\top\mathbf{M}_x = (\kappa_x(x_i, x_j)_{1\leq i,j\leq m})$ is the Gram matrix.

2.3 Maximum Margin Regression

An alternative solution to our regression learning problem is proposed in (Szedmak et al., 2006), called maximum margin regression. If L2-normalized feature vectors are used in Eq. 1, denoted by $\bar{\Phi}_x(\cdot)$ and $\bar{\Phi}_y(\cdot)$, MMR solves the following optimization:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{W}\|_F^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \langle \bar{\Phi}_y(y_i), \mathbf{W}\bar{\Phi}_x(x_i)\rangle_{\mathcal{H}_y} \geq 1 - \xi_i, \\ & \xi_i > 0, i = 1, \ldots, m. \end{aligned} \qquad (5)$$

where $C > 0$ is the regularization coefficient, and ξ_i are the slack variables. The Lagrange dual form with dual variables α_i gives:

$$\begin{aligned} \min \quad & \sum_{i,j=1}^{m}\alpha_i\alpha_j\bar{\kappa}_x(x_i, x_j)\bar{\kappa}_y(y_i, y_j) - \sum_{i=1}^{m}\alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \ i = 1, \ldots, m. \end{aligned} \qquad (6)$$

where $\bar{\kappa}_x(\cdot, \cdot)$ and $\bar{\kappa}_y(\cdot, \cdot)$ denote the kernel functions associated to the respective normalized feature vectors.

This dual problem can be solved efficiently with a perceptron algorithm based on an incremental subgradient method, of which the bounds on the complexity and achievable margin can be found in (Szedmak et al., 2006).

Then according to Karush-Kuhn-Tucker theory, $\mathbf{W}$ is expressed as:

$$\mathbf{W} = \sum_{i=1}^{m}\alpha_i\bar{\Phi}_y(y_i)\bar{\Phi}_x(x_i)^\top \qquad (7)$$

In practice, MMR works better when the distribution of the training points are symmetrical. So we center the data before normalizing them. If $\Phi_{S_x} = \frac{1}{m}\sum_{i=1}^{m}\Phi_x(x_i)$ is the centre of mass of the source sentence sample set $\{x_i\}$ in the feature space, the new feature map is given by $\hat{\Phi}_x(\cdot) = \Phi_x(\cdot) - \Phi_{S_x}$. The similar operation is performed on $\Phi_y(\cdot)$ to obtain $\hat{\Phi}_y(\cdot)$. Then the L2-normalizations of $\hat{\Phi}_x(\cdot)$ and $\hat{\Phi}_y(\cdot)$ yield our final feature vectors $\bar{\Phi}_x(\cdot)$ and $\bar{\Phi}_y(\cdot)$.

3 Pre-image Solution

To find the pre-image sentence $y = f^{-1}(x)$ can be achieved by seeking y_t that has the minimum loss between its feature vector $\Phi_y(y_t)$ and our prediction $f(x)$. That is (Eq. 8: LSR, Eq. 9: MMR):

$$\begin{aligned} y_t &= \arg\min_{y\in\mathcal{Y}(x)} \|\mathbf{W}\Phi_x(x) - \Phi_y(y)\|^2 \\ &= \arg\min_{y\in\mathcal{Y}(x)} \kappa_y(y, y) - 2k_y(y)\mathbf{K}_x^{-1}k_x(x) \end{aligned} \qquad (8)$$

$$\begin{aligned} y_t &= \arg\min_{y\in\mathcal{Y}(x)} 1 - \langle \bar{\Phi}_y(y), \mathbf{W}\bar{\Phi}_x(x)\rangle_{\mathcal{H}_y} \\ &= \arg\max_{y\in\mathcal{Y}(x)} \sum_{i=1}^{m}\alpha_i\bar{\kappa}_y(y_i, y)\bar{\kappa}_x(x_i, x) \end{aligned} \qquad (9)$$

where $\mathcal{Y}(x) \subset \mathcal{Y}$ is a finite set covering all potential translations for the given source sentence x, and $k_x(\cdot) = (\kappa_x(\cdot, x_i)_{1\leq i\leq m})$ and $k_y(\cdot) = (\kappa_y(\cdot, y_i)_{1\leq i\leq m})$ are $m \times 1$ column matrices.

A proper $\mathcal{Y}(x)$ can be generated according to a lexicon that contains possible translations for every component (word or phrase) in x. But the size of it will grow exponentially with the length of x, which poses implementation problem for a decoding algorithm.

In earlier systems, several heuristic search methods were developed, of which a typical example is Koehn (2004)'s beam search decoder for phrase-based models. However, in our case, because of the $\kappa_y(y, y)$ item in Eq. 8 and the normalization operation in MMR, neither the expression in Eq. 8 nor the one in Eq. 9 can be decomposed into a sum of subfunctions each involving feature components in a local area only. It means we cannot estimate exactly how well a part of the source sentence is translated, until we obtain a translation for the entire sentence, which prevents us doing a straightforward beam search similar to (Koehn, 2004).

To simplify the situation, we restrict the reordering (distortion) of phrases that yield the output sentences by only allowing adjacent phrases to exchange their positions. (The discussion of this strategy can be found in (Tillmann, 2004).) We use $x_{[i:j]}$ and $y_{[i:j]}$ to denote the substrings of x and y that begin with the ith word and end with the jth. Now, if we go back to the implementation of a beam search, the current distortion restriction guarantees that in each expansion of the search states (hypotheses) we have $x_{[1:l_x]}$ translated to a $y_{[1:l_y]}$, either like state (a) or like state (b) in Fig. 2, where l_x is the number of words translated in the source sentence, and l_y is the number of words obtained in the translation.

We assume that if y is a good translation of x, then $y_{[1:l_y]}$ is a good translation of $x_{[1:l_x]}$ as well. So we can expect that the squared loss $\|\mathbf{W}\Phi_x(x_{[1:l_x]}) - \Phi_y(y_{[1:l_y]})\|^2$ in the LSR is small, or the inner product $\langle \bar{\Phi}_y(y_{[1:l_y]}), \mathbf{W}\bar{\Phi}_x(x_{[1:l_x]})\rangle_{\mathcal{H}_y}$ in the MMR is large, for the hypothesis yielding a good translation. According to Eq. 8 and Eq. 9, the hypotheses in the search stacks can thus be reranked with the following score functions (Eq. 10: LSR, Eq. 11: MMR):

$$Score(x_{[1:l_x]}, y_{[1:l_y]}) = \kappa_y(y_{[1:l_y]}, y_{[1:l_y]}) - 2k_y(y_{[1:l_y]})\mathbf{K}_x^{-1}k_x(x_{[1:l_x]}) \tag{10}$$

$$Score(x_{[1:l_x]}, y_{[1:l_y]}) = \sum_{i=1}^{m} \alpha_i \bar{\kappa}_y(y_i, y_{[1:l_y]})\bar{\kappa}_x(x_i, x_{[1:l_x]}) \tag{11}$$

Therefore, to solve the pre-image problem, we just employ the same beam search algorithm as (Koehn, 2004), except we limit the derivation of new hypotheses with the distortion restriction mentioned above. However, our score functions will bring more runtime complexities when compared with traditional probabilistic methods. The time complexity of a naive implementation of the blended *n*-spectrum string kernel between two sentences s_i and s_j is $O(n|s_i||s_j|)$, where $|\cdot|$ denotes the length of the sentence. So the score function in Eq. 11 results in an average runtime complexity of $O(mnl_y l)$, where l is the average length of the sentences y_i in the training set. Note here $\bar{\kappa}_x(x_{[1:l_x]}, x_i)$ can be pre-computed for l_x from 1 to $|x|$ before the beam search, which calls for $O(m|x|)$ space. The average runtime complexity of the score function in Eq. 10 will be the same if we pre-compute $\mathbf{K}_x^{-1}k_x(x_{[1:l_x]})$.

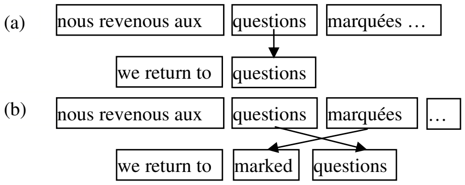

Figure 2: Search states with the limited distortion.

4 Experimental Results

4.1 Resource Description

Baseline System To compare with previous work, we take Pharaoh (Koehn, 2004) as a baseline system, with its default settings (translation table size 10, beam size 100). We train a trigram language model with the SRILM toolkit (Stocke, 2002). Whilst, the parameters for the maximum entropy model are developed based on the minimum error rate training method (Och, 2003).

In the following experiments, to facilitate comparison, each time we train our regression models and the language model and translation model for Pharaoh on a common corpus, and use the same phrase translation table as Pharaoh's to decode our systems. According to our preliminary experiments, with the beam size of 100, the search errors of our systems can be limited within 1.5%.

Corpora To evaluate our models, we randomly take 12,000 sentences from the French-English portion of the 1996–2003 Europarl corpus (Koehn, 2005) for scaling-up training, 300 for test (Test), and 300 for the development of Pharaoh (Dev). Some

	Vocabulary		Words		Perplexity	
	Fr	En	Fr	En	Dev	Test
4k	5084	4039	43k	39k	32.25	31.92
6k	6426	5058	64k	59k	30.81	29.03
8k	7377	5716	85k	79k	29.91	28.94
10k	8252	6339	106k	98k	27.55	27.09
12k	9006	6861	127k	118k	27.19	26.41

Table 1: Statistics of the corpora.

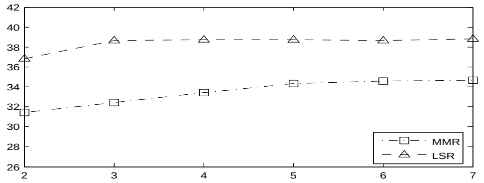

Figure 3: BLEU(%) versus *n*-spectrum

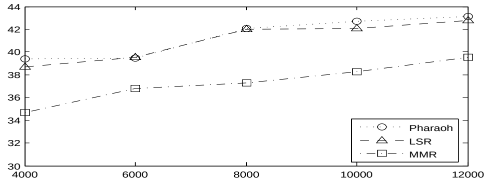

Figure 4: BLEU(%) versus training set size

characteristics of the corpora are summarized in Table 1.

4.2 Results

Based on the 4k training corpus, we test the performance of the blended *n*-spectrum string kernel in LSR and MMR using BLEU score, with n increasing from 2 to 7. Fig. 3 shows the results. It can be found that the performance becomes stable when *n* reaches a certain value. Finally, we choose the 3-spectrum for LSR, and the 5-spectrum for MMR.

Then we scale up the training set, and compare the performance of our models with Pharaoh in Fig. 4. We can see that the LSR model performs almost as well as Pharaoh, whose differences of BLEU score are within 0.5% when the training set is larger than 6k. But MMR model performs worse than the baseline. With the training set of 12k, it is outperformed by Pharaoh by 3.5%.

5 Discussions

Although at this stage the main contribution is still conceptual, the capability of our approach to be applied to machine translation is still demonstrated. Comparable performance to previous work is achieved by the LSR model.

But a main problem we face is to scale-up the training set, as in practice the training set for SMT will be much larger than several thousand sentences. A method to speed up the training is proposed in (Cortes et al., 2005). By approximating the Gram matrix with a $n \times m$ $(n \ll m)$ low-rank matrix, the time complexity of the matrix inversion operation can be reduced from $O(m^3)$ to $O(n^2m)$. But the space complexity of $O(nm)$ in their algorithm is still too expensive for SMT tasks. Subset selection techniques could give a solution to this problem, of which we will leave the further exploration to future work.

Acknowledgements

The authors acknowledge the support of the EU under the IST project No. FP6-033917.

References

C. Cortes, M. Mohri, and J. Weston. 2005. A general regression technique for learning transductions. In *Proc. of ICML'05*.

P. Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proc. of AMTA 2004*.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL'03*.

A. Stocke. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP'02*.

S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. 2006. Learning via linear operators: Maximum margin regression; multiclass and multiview learning at one-class complexity. Technical report, PASCAL, Southampton, UK.

C. Tillmann. 2004. A unigram orientation model for statistical machine translation. In *Proc. of HLT-NAACL'04*.