A Systematic Exploration of Diversity in Machine Translation
Supplementary Material

Kevin Gimpel[*] **Dhruv Batra**[†] **Chris Dyer**[‡] **Gregory Shakhnarovich**[*]
[*]Toyota Technological Institute at Chicago, Chicago, IL 60637, USA
[†]Virginia Tech, Blacksburg, VA 24061, USA
[‡]Carnegie Mellon University, Pittsburgh, PA 15213, USA
Corresponding author: kgimpel@ttic.edu

1 Structured Support Vector Machine Reranking

We now describe the reranking algorithm of Yadollahpour et al. (2013) and discuss how we applied it to MT.

Let $\mathbf{Y}_i = \{\mathbf{y}_i^{(1)} \ldots \mathbf{y}_i^{(M)}\}$ denote the set of M translations for source sentence x_i, and $\mathbf{Y}_i^R$ the set of reference translations.[1] Note that frequently it is the case that $\mathbf{Y}_i \cap \mathbf{Y}_i^R = \varnothing$. Let $\mathbf{y}_i^*$ denote the highest-quality translation in the set, i.e., $\mathbf{y}_i^* = \text{argmin}_{\mathbf{y} \in \mathbf{Y}_i} \ell(\mathbf{Y}_i^R, \mathbf{y})$, where $\ell(\mathbf{Y}_i^R, \mathbf{y})$ is the negated BLEU+1 score (Lin and Och, 2004) of $\mathbf{y}$. The quality of solution $\mathbf{y}_i^*$ leads to an upper-bound on the reranker performance since we must select one solution from $\mathbf{Y}_i$.

The reranking model assigns a score S_r to each translation in the set, i.e., $S_r(\mathbf{y}) = \boldsymbol{\alpha}^\intercal \boldsymbol{\psi}(x, \mathbf{y})$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}(x, \mathbf{y})$ are the reranker weights and features respectively. The reranking features can be quite complex, as decoding simply finds the highest scoring solution in the set: $\hat{\mathbf{y}}_i = \text{argmax}_{\mathbf{y} \in \mathbf{Y}_i} S_r(\mathbf{y})$.

The objective of the reranker (picking the best solution $\mathbf{y}_i^*$ from the set) is formulated as a structured SVM (Tsochantaridis et al., 2005). For training, we use the following loss function, defined for a hypothesis $\hat{\mathbf{y}}_i$:

$$\mathcal{L}(\mathbf{Y}_i^R, \hat{\mathbf{y}}_i) = \ell(\mathbf{Y}_i^R, \hat{\mathbf{y}}_i) - \ell(\mathbf{Y}_i^R, \mathbf{y}_i^*), \quad (1)$$

i.e., the negated BLEU+1 score of translation $\hat{\mathbf{y}}_i$ *relative* to that of the best translation in this set ($\mathbf{y}_i^*$). This relative loss forces the reranker to focus its effort on training instances where it is underperforming w.r.t. the set, rather than in absolute terms. For instance, consider two input sentences i, j with two translations each. The translations for sentence i have BLEU+1 scores of 65 and 45 while the translations for sentence j have scores 40 and 35. Using only absolute ℓ in the reranking objective would emphasize sentence j, since both translations for j have low BLEU compared to translations for i. Using the relative loss correctly shifts the focus to i because an incorrect choice in that set is much costlier (difference of 20 points) than an incorrect choice in set j (5 points).

We learn the reranker parameters $\boldsymbol{\alpha}$ by solving the following quadratic program:

$$\min_{\boldsymbol{\alpha}, \xi_i} \ \|\boldsymbol{\alpha}\|_2^2 + C \sum_{i \in [n]} \xi_i \quad (2a)$$

$$\text{s.t. } \boldsymbol{\alpha}^\intercal \Big(\boldsymbol{\psi}(x_i, \mathbf{y}_i^*) - \boldsymbol{\psi}(x_i, \mathbf{y}) \Big) \geq 1 - \frac{\xi_i}{\mathcal{L}(\mathbf{Y}_i^R, \mathbf{y})} \quad (2b)$$

$$\xi_i \geq 0, \qquad \forall \mathbf{y} \in \mathbf{Y}_i \setminus \mathbf{y}_i^*, \quad (2c)$$

Intuitively, (2b) maximizes the (soft) margin between the score of the oracle solution and all other solutions in the set. The violation in the margin ξ_i is scaled by the loss of the solution. Thus if in addition to $\mathbf{y}_i^*$ there are other good solutions in the set, the margin for such solutions will not be tightly enforced. On the other hand, the margin between $\mathbf{y}_i^*$ and bad solutions will be very strictly enforced. We solve (2) via the 1-slack cutting-plane algorithm of Joachims et al. (2009). We used OOQP (Gertz and Wright, 2003) to solve the quadratic program

[1] For conciseness, we do not explicitly show the derivation variable $\mathbf{h}_i^{(j)}$ associated with translation j, but it is always available for computing features for $\mathbf{y}_i^{(j)}$.

in the inner loop, which uses HSL, a collection of Fortran codes for large-scale scientific computation (`www.hsl.rl.ac.uk`).

References

E. M. Gertz and S. J. Wright. 2003. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29(1).

T. Joachims, T. Finley, and C. Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1).

C. Lin and F. J. Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proc. of COLING*.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. 2005. Large margin methods for structured and interdependent output variables. *JMLR*, 6.

P. Yadollahpour, D. Batra, and G. Shakhnarovich. 2013. Discriminative re-ranking of diverse segmentations. In *Proc. of CVPR*.