Towards Decoding as Continuous Optimisation in Neural Machine Translation

Cong Duy Vu Hoang
University of Melbourne
Melbourne, VIC, Australia
vhoang2@student.unimelb.edu.au

Gholamreza Haffari
Monash University
Clayton, VIC, Australia
gholamreza.haffari@monash.edu

Trevor Cohn
University of Melbourne
Melbourne, VIC, Australia
t.cohn@unimelb.edu.au

*in EMNLP 2017

Neural Machine Translation

$$
\begin{aligned}
\mathrm{P}_{\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{x}) &= \sum_{i=1}^{|\boldsymbol{y}|} \log \mathrm{P}_{\boldsymbol{\Theta}}(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x}) \\
y_i|\boldsymbol{y}_{<i}, \boldsymbol{x} &\sim \mathrm{softmax}\left(\boldsymbol{f}(\boldsymbol{\Theta}, \boldsymbol{y}_{<i}, \boldsymbol{x})\right)
\end{aligned}
$$

In this paper:

$$
\begin{aligned}
\boldsymbol{f}(\boldsymbol{\Theta}, \boldsymbol{y}_{<i}, \boldsymbol{x}) = \boldsymbol{W}_o \cdot \mathrm{MLP}\left(\boldsymbol{c}_i, \boldsymbol{E}_T^{y_{i-1}}, \boldsymbol{g}_i\right) + \boldsymbol{b}_o \\
\boldsymbol{g}_i = \mathrm{RNN}_{dec}^{\phi}\left(\boldsymbol{c}_i, \boldsymbol{E}_T^{y_{i-1}}, \boldsymbol{g}_{i-1}\right)
\end{aligned}
$$

Training objective:

$$
\boldsymbol{\Theta}^* := \mathrm{argmax}_{\boldsymbol{\Theta}} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{D}} \log \mathrm{P}_{\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{x}) .
$$

Discrete decoding

- Decoding objective:

$$\boldsymbol{y}^* = \operatorname*{arg\,min}_{y_1,\ldots,y_\ell} \sum_{i=1}^{\ell} -\log \mathrm{P}_{\boldsymbol{\Theta}}\left(y_i \mid \boldsymbol{y}_{<i}, \boldsymbol{x}\right)$$
$$\text{s.t.} \quad \forall i \in \{1 \ldots \ell\} : y_i \in V_T.$$

- Re-write:

$$\operatorname*{arg\,min}_{\tilde{\boldsymbol{y}}_1,\ldots,\tilde{\boldsymbol{y}}_\ell} - \sum_{i=1}^{\ell} \tilde{\boldsymbol{y}}_i \cdot \log \operatorname{softmax}\left(\boldsymbol{f}\left(\boldsymbol{\Theta}, \tilde{\boldsymbol{y}}_{<i}, \boldsymbol{x}\right)\right)$$
$$\text{s.t.} \quad \forall i \in \{1 \ldots \ell\} : \tilde{\boldsymbol{y}}_i \in \mathbb{I}^{|V_T|} \qquad (5)$$

where $\tilde{\boldsymbol{y}}_i$ are vectors using the one-hot representation of the target words $\mathbb{I}^{|V_T|}$.

discrete optimization problem

Continuous decoding

$$\boxed{\begin{aligned}&\underset{\hat{\boldsymbol{y}}_1,\ldots,\hat{\boldsymbol{y}}_\ell}{\arg\min} - \sum_{i=1}^{\ell} \hat{\boldsymbol{y}}_i \cdot \log \mathrm{softmax}\left(\boldsymbol{f}\left(\boldsymbol{\Theta}, \hat{\boldsymbol{y}}_{<i}, \boldsymbol{x}\right)\right) \\ &\text{s.t.} \quad \forall i \in \{1 \ldots \ell\} : \hat{\boldsymbol{y}}_i \in \Delta_{|V_T|}\end{aligned}}$$

where $\Delta_{|V_T|}$ is the $|V_T|$-dimensional probability simplex, i.e., $\{\hat{\boldsymbol{y}}_i \in [0,1]^{|V_T|} : \|\hat{\boldsymbol{y}}_i\|_1 = 1\}$. Intuitively, this amounts to replacing $\boldsymbol{E}_T^{y_i}$ with the *expected* embedding of target language words $\mathbb{E}_{\hat{\boldsymbol{y}}_i(w)}[\boldsymbol{E}_T^{w}]$ under the distribution $\hat{\boldsymbol{y}}_i$ in the NMT model.

Y_i is a distribution over words
Generation words: choose the highest probability

Optimization method

- Exponentiated Gradient (EG)

Problem:

$$\arg\min_{\hat{\boldsymbol{y}}_1,\ldots,\hat{\boldsymbol{y}}_\ell} Q(\hat{\boldsymbol{y}}_1,\ldots,\hat{\boldsymbol{y}}_\ell)$$

$$\text{s.t.} \quad \forall i \in \{1\ldots\ell\} : \hat{\boldsymbol{y}}_i \in \Delta_{|V_T|}$$

where $Q(\hat{\boldsymbol{y}}_1,\ldots,\hat{\boldsymbol{y}}_\ell)$ is defined as

$$-\sum_{i=1}^{\ell} \hat{\boldsymbol{y}}_i \cdot \log \operatorname{softmax}\left(\boldsymbol{f}\left(\boldsymbol{\Theta}, \hat{\boldsymbol{y}}_{<i}, \boldsymbol{x}\right)\right).$$

Algorithm:

$$\forall w \in V_T : \quad \hat{\boldsymbol{y}}_i^t(w) = \frac{1}{Z_i^t}\hat{\boldsymbol{y}}_i^{t-1}(w)\exp\left(-\eta\nabla_{i,w}^{t-1}\right)$$

where η is the step size, $\nabla_{i,w}^{t-1} = \frac{\partial Q(\hat{\boldsymbol{y}}_1^{t-1},\ldots,\hat{\boldsymbol{y}}_\ell^{t-1})}{\partial \hat{\boldsymbol{y}}_i(w)}$ and Z_i^t is the normalisation constant

$$Z_i^t = \sum_{w \in V_T} \hat{\boldsymbol{y}}_i^{t-1}(w)\exp\left(-\eta\nabla_{i,w}^{t-1}\right).$$

Algorithm 1 The EG Algorithm for Decoding by Optimisation

1: For all i initialise $\hat{\boldsymbol{y}}_i^0 \in \Delta_{|V_T|}$
2: **for** $t = 1,\ldots,$ MaxIter **do** ▷ $Q(.)$ is defined as eqn (6)
3: For all i, w : calculate $\nabla_{i,w}^{t-1} = \frac{\partial Q(\hat{\boldsymbol{y}}_1^{t-1},\ldots,\hat{\boldsymbol{y}}_\ell^{t-1})}{\partial \hat{\boldsymbol{y}}_i(w)}$ ▷ using backpropagation
4: For all i, w : update $\hat{\boldsymbol{y}}_i^t(w) \propto \hat{\boldsymbol{y}}_i^{t-1}(w) \cdot \exp\left(-\eta\nabla_{i,w}^{t-1}\right)$ ▷ η is the step size
5: **return** $\arg\min_t Q(\hat{\boldsymbol{y}}_1^t,\ldots,\hat{\boldsymbol{y}}_\ell^t)$

Optimization method

- Stochastic Gradient Descent (SGD)

Make sure the simplex constraints:

$$\hat{\boldsymbol{y}}_i = \mathrm{softmax}\left(\hat{\boldsymbol{r}}_i\right).$$

$$\underset{\hat{\boldsymbol{r}}_1,\ldots,\hat{\boldsymbol{r}}_\ell}{\arg\min} - \sum_{i=1}^{\ell} \mathrm{softmax}\left(\hat{\boldsymbol{r}}_i\right) \cdot \log \mathrm{softmax}\left(\mathrm{f}\left(\boldsymbol{\Theta}, \hat{\boldsymbol{y}}_{<i}, \boldsymbol{x}\right)\right)$$

Algorithm 2 The SGD Algorithm for Decoding by Optimisation

1: For all i initialise $\hat{\boldsymbol{r}}_i^0$

2: **for** $t = 1, \ldots,$ MaxIter **do** $\triangleright$ $Q(.)$ is defined in eqn (6) and $\hat{\boldsymbol{y}}_i = \mathrm{softmax}(\hat{\boldsymbol{r}}_i)$

3: For all i, w : calculate $\nabla_{i,w}^{t-1} = \sum_{w' \in V_T} \frac{\partial Q(\hat{\boldsymbol{y}}_1^{t-1},\ldots,\hat{\boldsymbol{y}}_\ell^{t-1})}{\partial \hat{\boldsymbol{y}}_i(w')} \frac{\partial \hat{\boldsymbol{y}}_i(w')}{\partial \hat{\boldsymbol{r}}_i(w)}$ $\triangleright$ using backpropagation

4: For all i, w : update $\hat{\boldsymbol{r}}_i^t(w) = \hat{\boldsymbol{r}}_i^{t-1}(w) - \eta \nabla_{i,w}^{t-1}$ $\triangleright$ η is the step size

5: **return** $\arg\min_t Q(\mathrm{softmax}(\hat{\boldsymbol{r}}_1^t), \ldots, \mathrm{softmax}(\hat{\boldsymbol{r}}_\ell^t))$

Decoding in Extended NMT

- Allowing decoding for richer global models, for which there is no effective means of greedy decoding or beam search.

- Bidirectional Ensemble:

$$\begin{aligned}\mathcal{C}_{+\text{bidir}} :=& -\alpha \log \mathrm{P}_{\boldsymbol{\Theta}_{\leftarrow}}(\boldsymbol{y} \mid \boldsymbol{x}) \\ & -(1-\alpha) \log \mathrm{P}_{\boldsymbol{\Theta}_{\rightarrow}}(\boldsymbol{y} \mid \boldsymbol{x})\,;\end{aligned}$$

- Bilingual Ensemble:

$$\begin{aligned}\mathcal{C}_{+\text{biling}} :=& -\alpha \log \mathrm{P}_{\boldsymbol{\Theta}_{\mathrm{s}\rightarrow\mathrm{t}}}(\boldsymbol{y} \mid \boldsymbol{x}) \\ & -(1-\alpha) \log \mathrm{P}_{\boldsymbol{\Theta}_{\mathrm{s}\leftarrow\mathrm{t}}}(\boldsymbol{x} \mid \boldsymbol{y})\,;\end{aligned}$$

Experiment

- Main results:

	BTEC zh→en	TEDTalks de→en	WMT de→en
$\text{gdec}_{\text{left-to-right}}$	35.98	23.16	24.41
$\text{gdec}_{\text{right-to-left}}$	35.86	21.95	23.59
$\text{EGdec}_{\text{greedy init}}$	36.34	23.28	24.63
+bidirectional	**36.67**	**23.91**	**25.37**†
+bilingual	**36.88**†	**24.01**†	**25.21**
$\text{bdec}_{\text{left-to-right}}$	38.02	23.95	26.69
$\text{bdec}_{\text{right-to-left}}$	37.38	23.13	26.11
$\text{EGdec}_{\text{beam init}}$	38.38	24.02	26.66
+bidirectional	**39.13**†	**24.72**†	**27.34**†
+bilingual	38.25	**24.60**	26.82

Table 3: The BLEU evaluation results across evaluation datasets for EG algorithm variants against the baselines; **bold**: statistically significantly better than the best greedy or beam baseline, †: best performance on dataset.