

# Towards Decoding as Continuous Optimisation in Neural Machine Translation

Cong Duy Vu Hoang<sup>†</sup> and Gholamreza Haffari<sup>‡</sup> and Trevor Cohn<sup>†</sup>

<sup>†</sup> University of Melbourne  
Melbourne, VIC, Australia

<sup>‡</sup> Monash University  
Clayton, VIC, Australia

vhoang2@student.unimelb.edu.au, gholamreza.haffari@monash.edu,  
t.cohn@unimelb.edu.au

## Abstract

We propose a novel decoding approach for neural machine translation (NMT) based on continuous optimisation. We reformulate decoding, a discrete optimization problem, into a continuous problem, such that optimization can make use of efficient gradient-based techniques. Our powerful decoding framework allows for more accurate decoding for standard neural machine translation models, as well as enabling decoding in intractable models such as intersection of several different NMT models. Our empirical results show that our decoding framework is effective, and can lead to substantial improvements in translations, especially in situations where greedy search and beam search are not feasible. Finally, we show how the technique is highly competitive with, and complementary to, reranking.

## 1 Introduction

Sequence to sequence learning with neural networks (Graves, 2013; Sutskever et al., 2014; Lip-ton et al., 2015) is typically associated with two phases: training and decoding (*a.k.a.* inference). Model parameters are learned by optimising the training objective, in order that the model can produce good translations when decoding unseen sentences. The majority of research has focused on the training paradigm or network architecture, however effective means of decoding have been under-investigated. Conventional heuristic-based approaches for approximate inference include greedy, beam, and stochastic search. Greedy and beam search have been empirically proved to be adequate for many sequence to sequence tasks, and are the standard methods for NMT decoding.

However, these inference approaches have several drawbacks. Firstly, although NMT models use a left-to-right generation which would appear to facilitate efficient search, the models themselves use a recurrent architecture, and accordingly are non-Markov. This prevents exact dynamic programming solutions, and moreover, limits the potential to incorporate additional global features or constraints. Global factors can be highly useful in producing better and more diverse translations. Secondly, the sequential decoding of symbols in the target sequence, the inter-dependencies among the target symbols are not fully exploited. For example, when decoding the words of the target sentence in a left-to-right manner, the right context is not exploited leading potentially to inferior performance (see Watanabe and Sumita (2002a) who apply this idea in traditional statistical MT). A natural way to capture this is to intersect left-to-right and right-to-left models, however the resulting model has no natural generation order, and thus standard decoding methods are unsuitable.

We introduce a novel decoding framework (§ 3) that relaxes this *discrete* optimisation problem into a *continuous* optimisation problem. This is akin to linear programming relaxation approach for approximate inference in graphical models with discrete random variables, where the exact inference is NP-hard (Sontag, 2010; Belanger and McCallum, 2016). The resulting continuous optimisation problem is challenging due to the non-linearity and non-convexity of the relaxed decoding objective. We make use of stochastic gradient descent (SGD) and exponentiated gradient (EG) algorithms for decoding based on our relaxation approach.<sup>1</sup> Our decoding framework is powerful and flexible, as it enables us to decode with global constraints involving intersection of multiple NMT

<sup>1</sup>Both methods are mainly used for training in prior work.

models (§4). We present experimental results on Chinese-English and German-English translation tasks, confirming the effectiveness of our relaxed optimisation method for decoding (§5).

## 2 Neural Machine Translation

We briefly review the attentional neural translation model proposed by Bahdanau et al. (2015) as a sequence-to-sequence neural model onto which we apply our decoding framework.

In neural machine translation (NMT), the probability of the target sentence  $\mathbf{y}$  given a source sentence  $\mathbf{x}$  is written as:

$$P_{\Theta}(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{|\mathbf{y}|} \log P_{\Theta}(y_i|\mathbf{y}_{<i}, \mathbf{x}) \quad (1)$$

$$y_i|\mathbf{y}_{<i}, \mathbf{x} \sim \text{softmax}(\mathbf{f}(\Theta, \mathbf{y}_{<i}, \mathbf{x}))$$

where  $\mathbf{f}$  is a non-linear function of the previously generated sequence of words  $\mathbf{y}_{<i}$ , the source sentence  $\mathbf{x}$ , and the model parameters  $\Theta$ . In this paper, we realise  $\mathbf{f}$  as follows:

$$\mathbf{f}(\Theta, \mathbf{y}_{<i}, \mathbf{x}) = \mathbf{W}_o \cdot \text{MLP}(\mathbf{c}_i, \mathbf{E}_T^{y_{i-1}}, \mathbf{g}_i) + \mathbf{b}_o$$

$$\mathbf{g}_i = \text{RNN}_{dec}^{\phi}(\mathbf{c}_i, \mathbf{E}_T^{y_{i-1}}, \mathbf{g}_{i-1})$$

where MLP is a single hidden layer neural network with tanh activation function, and  $\mathbf{E}_T^{y_{i-1}}$  is the embedding of the target word  $y_{i-1}$  in the embedding matrix  $\mathbf{E}_T \in \mathbb{R}^{n_e \times |V_T|}$  of the target language vocabulary  $V_T$  and  $n_e$  is the embedding dimension. The state  $\mathbf{g}_i$  of the decoder RNN is a function of  $y_{i-1}$ , its previous state  $\mathbf{g}_{i-1}$ , and the context  $\mathbf{c}_i = \sum_{j=1}^{|\mathbf{x}|} \alpha_{ij} \mathbf{h}_j$  summarises parts of the source sentence which are *attended* to, where

$$\alpha_i = \text{softmax}(e_i) \quad ; \quad e_{ij} = \text{MLP}(\mathbf{g}_{i-1}, \mathbf{h}_j)$$

$$\mathbf{h}_j = \text{biRNN}_{enc}^{\theta}(\mathbf{E}_S^{x_j}, \overrightarrow{\mathbf{h}}_{j-1}, \overleftarrow{\mathbf{h}}_{j+1})$$

In above,  $\overrightarrow{\mathbf{h}}_i$  and  $\overleftarrow{\mathbf{h}}_i$  are the states of the left-to-right and right-to-left RNNs encoding the source sentence, and  $\mathbf{E}_S^{x_j}$  is the embedding of the source word  $x_j$  in the embedding matrix  $\mathbf{E}_S \in \mathbb{R}^{n'_e \times |V_S|}$  of the source language vocabulary  $V_S$  and  $n'_e$  is the embedding dimension.

Given a bilingual corpus  $\mathcal{D}$ , the model parameters are learned by maximizing the conditional log-likelihood,

$$\Theta^* := \text{argmax}_{\Theta} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \log P_{\Theta}(\mathbf{y}|\mathbf{x}). \quad (2)$$

The model parameters  $\Theta$  include the weight matrix  $\mathbf{W}_o \in \mathbb{R}^{|V_T| \times n_h}$  and the bias  $\mathbf{b}_o \in \mathbb{R}^{|V_T|}$  – with  $n_H$  denoting the hidden dimension size – as well as the RNN encoder  $\text{biRNN}_{enc}^{\theta}$  / decoder  $\text{RNN}_{dec}^{\phi}$  parameters, word embedding matrices, and the parameters of the attention mechanism. The model is trained end-to-end by optimising the training objective using stochastic gradient descent (SGD) or its variants. In this paper, we focus on the decoding problem, which we turn to in the next section.

## 3 Decoding as Continuous Optimisation

In decoding, we are interested in finding the highest probability translation for a given source sentence:

$$\text{minimise}_{\mathbf{y}} \quad -P_{\Theta}(\mathbf{y}|\mathbf{x}) \quad \text{s.t.} \quad \mathbf{y} \in \mathcal{Y}_{\mathbf{x}} \quad (3)$$

where  $\mathcal{Y}_{\mathbf{x}}$  is the space of possible translations for the source sentence  $\mathbf{x}$ . In general, searching  $\mathcal{Y}_{\mathbf{x}}$  to find the highest probability translation is intractable due to the recurrent nature of eqn (1) which prevents dynamic programming for efficient search. This is problematic, as the space of translations is exponentially large with respect to the output length  $|\mathbf{y}|$ .

We now formulate this discrete optimisation problem as a continuous one, and then use standard algorithms for continuous optimisation for decoding. Let us assume that the maximum length of a possible translation for a source sentence is known and denote it as  $\ell$ . The best translation for a given source sentence solves the following optimisation problem:

$$\mathbf{y}^* = \text{arg min}_{y_1, \dots, y_{\ell}} \sum_{i=1}^{\ell} -\log P_{\Theta}(y_i|\mathbf{y}_{<i}, \mathbf{x}) \quad (4)$$

$$\text{s.t.} \quad \forall i \in \{1 \dots \ell\} : y_i \in V_T.$$

where we allow the translation to be padded with sentinel symbols to the right, which are ignored in computing the model probability. Equivalently, we can rewrite the above discrete optimisation problem as follows:

$$\text{arg min}_{\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_{\ell}} - \sum_{i=1}^{\ell} \tilde{\mathbf{y}}_i \cdot \log \text{softmax}(\mathbf{f}(\Theta, \tilde{\mathbf{y}}_{<i}, \mathbf{x}))$$

$$\text{s.t.} \quad \forall i \in \{1 \dots \ell\} : \tilde{\mathbf{y}}_i \in \mathbb{I}^{|V_T|} \quad (5)$$

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

---

**Algorithm 1** The EG Algorithm for Decoding by Optimisation
 

---

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

We now convert the optimisation problem (5) to a continuous one by dropping the integrality constraints  $\hat{\mathbf{y}}_i \in \mathbb{I}^{|V|}$  and require the variables to take values from the probability simplex:

$$\begin{aligned} \arg \min_{\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_\ell} & - \sum_{i=1}^{\ell} \hat{\mathbf{y}}_i \cdot \log \text{softmax}(\mathbf{f}(\Theta, \hat{\mathbf{y}}_{<i}, \mathbf{x})) \\ \text{s.t. } & \forall i \in \{1 \dots \ell\} : \hat{\mathbf{y}}_i \in \Delta_{|V_T|} \end{aligned}$$

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

After solving the above constrained continuous optimisation problem, there is no guarantee that the resulting solution  $\{\hat{\mathbf{y}}_i^*\}_{i=1}^{\ell}$  will comprise one-hot vectors, i.e., target language words. Instead it can find *fractional* solutions, that require ‘rounding’ in order to resolve them to lexical items. To solve this problem, we take the  $\arg \max$ ,<sup>2</sup> i.e., take the highest scoring word for each position  $\hat{\mathbf{y}}_i^*$ . We leave exploration of more elaborate projection techniques to the future work.

In the context of graphical models, the above relaxation technique gives rise to linear programming for approximate inference (Sontag, 2010; Belanger and McCallum, 2016). However, our decoding problem is much harder due to the non-linearity and non-convexity of the objective function operating on high dimensional space for deep models. We now turn our attention to optimisation algorithms to effectively solve the decoding optimisation problem.

### 3.1 Exponentiated Gradient (EG)

Exponentiated gradient (Kivinen and Warmuth, 1997) is an elegant algorithm for solving optimisation problems involving simplex constraints. Re-

call our constrained optimisation problem:

$$\begin{aligned} \arg \min_{\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_\ell} & Q(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_\ell) \\ \text{s.t. } & \forall i \in \{1 \dots \ell\} : \hat{\mathbf{y}}_i \in \Delta_{|V_T|} \end{aligned}$$

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

$$- \sum_{i=1}^{\ell} \hat{\mathbf{y}}_i \cdot \log \text{softmax}(\mathbf{f}(\Theta, \hat{\mathbf{y}}_{<i}, \mathbf{x})). \quad (6)$$

EG is an iterative algorithm, which updates each distribution  $\hat{\mathbf{y}}_i^t$  in the current time-step  $t$  based on the distributions of the previous time-step as follows:

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

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

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

The partial derivatives  $\nabla_{i,w}$  are calculated using the back propagation algorithm treating  $\{\hat{\mathbf{y}}_i\}_{i=1}^{\ell}$  as *parameters* and the original parameters of the model  $\Theta$  as constants. Adapting EG to our decoding problem leads to Algorithm 1.

It can be shown that the EG algorithm is a gradient descent algorithm for minimising the following objective function subject to the simplex constraints:

$$\begin{aligned} Q(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_\ell) & - \gamma \sum_{i=1}^{\ell} \sum_{w \in V_T} \hat{\mathbf{y}}_i(w) \log \frac{1}{\hat{\mathbf{y}}_i(w)} \\ & = Q(\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_\ell) - \gamma \sum_{i=1}^{\ell} \text{Entropy}(\hat{\mathbf{y}}_i) \end{aligned} \quad (7)$$

In other words, the algorithm looks for the maximum entropy solution which also maximizes the

<sup>2</sup>Ties are broken arbitrarily.

log likelihood under the model. There are intriguing parallels with the maximum entropy formulation of log-linear models (Berger et al., 1996). In our setting, the entropy term acts as a prior which discourages overly-confident estimates in the absence of sufficient evidence.

### 3.2 Stochastic Gradient Descent (SGD)

To be able to apply SGD to our optimisation problem, we need to make sure that the simplex constraints are enforced. One way to achieve this is by reparameterising using the softmax transformation, i.e.  $\hat{\mathbf{y}}_i = \text{softmax}(\hat{\mathbf{r}}_i)$ . The resulting *unconstrained* optimisation problem, now over  $\hat{\mathbf{r}}_i$ , becomes

$$\arg \min_{\hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_\ell} - \sum_{i=1}^{\ell} \text{softmax}(\hat{\mathbf{r}}_i) \cdot \log \text{softmax}(f(\Theta, \hat{\mathbf{y}}_{<i}, \mathbf{x}))$$

where  $\mathbf{E}_T^{y_i}$  is replaced with the expected embedding of the target words under the distribution resulted from the  $\mathbb{E}_{\text{softmax}(\hat{\mathbf{r}}_i)}[\mathbf{E}_T^w]$  in the model.

To apply SGD updates, we need the gradient of the objective function with respect to the new variables  $\hat{\mathbf{r}}_i$  which can be derived with the backpropagation algorithm based on the chain rule:

$$\frac{\partial Q}{\partial \hat{\mathbf{r}}_i(w)} = \sum_{w' \in V_T} \frac{\partial Q(\cdot)}{\partial \hat{\mathbf{y}}_i(w')} \frac{\partial \hat{\mathbf{y}}_i(w')}{\partial \hat{\mathbf{r}}_i(w)}$$

The resulting SGD algorithm is summarized in Algorithm 2.

## 4 Decoding in Extended NMT

Our decoding framework allows us to effectively and flexibly add additional global factors over the output symbols during inference. This enables decoding for richer global models, for which there is no effective means of greedy decoding or beam search. We outline several such models, and their corresponding relaxed objective functions for optimisation-based decoding.

**Bidirectional Ensemble.** Standard NMT generates the translation in a left-to-right manner, conditioning each target word on its left context. However, the joint probability of the translation can be decomposed in a myriad of different orders; one compelling alternative would be to condition each target word on its right context, i.e., generating the target sentence from right-to-left. We would not expect a right-to-left model to outperform a left-to-right, however, as the left-to-right ordering

reflects the natural temporal order of spoken language. However, the right-to-left model is likely to provide a complementary signal in translation, as it will be bringing different biases and making largely independent prediction errors to those of the left-to-right model. For this reason, we propose to use both models, and seek to find translations that have high probability according both models (this mirrors work on bidirectional decoding in classical statistical machine translation by Watanabe and Sumita (2002b).) Decoding under the ensemble of these models leads to an intractable search problem, not well suited to traditional greedy or beam search algorithms, which require a fixed generation order of the target words. This ensemble decoding problem can be formulated simply in our linear relaxation approach, using the following objective function:

$$\mathcal{C}_{+\text{bidir}} := -\alpha \log P_{\Theta_{\leftarrow}}(\mathbf{y} | \mathbf{x}) - (1 - \alpha) \log P_{\Theta_{\rightarrow}}(\mathbf{y} | \mathbf{x}); \quad (8)$$

where  $\alpha$  is an interpolation hyper-parameter, which we set to 0.5;  $\Theta_{\rightarrow}$  and  $\Theta_{\leftarrow}$  are the pre-trained left-to-right and right-to-left models, respectively. This bidirectional agreement may also lead to improvement in translation diversity, as shown in Li and Jurafsky (2016) in a re-ranking evaluation.

**Bilingual Ensemble.** Another source of complementary information is in terms of the translation direction, that is forward translation from the source to the target language, and reverse translation in the target to source direction. Decoding must find a translation which scores well under both the forward and reverse translation models. This is inspired by the direct and reverse feature functions commonly used in classical discriminative SMT (Och and Ney, 2002) which have been shown to offer some complementary benefits (although see Lopez and Resnik (2006)). More specifically, we decode for the best translation in the intersection of the source-to-target and target-to-source models by minimizing the following objective function:

$$\mathcal{C}_{+\text{biling}} := -\alpha \log P_{\Theta_{s \rightarrow t}}(\mathbf{y} | \mathbf{x}) - (1 - \alpha) \log P_{\Theta_{s \leftarrow t}}(\mathbf{x} | \mathbf{y}); \quad (9)$$

where  $\alpha$  is an interpolation hyper-parameter to be fine-tuned; and  $\Theta_{s \rightarrow t}$  and  $\Theta_{s \leftarrow t}$  are the pre-trained

---

**Algorithm 2** The SGD Algorithm for Decoding by Optimisation

---

- 1: For all  $i$  initialise  $\hat{r}_i^0$
  - 2: **for**  $t = 1, \dots, \text{MaxIter}$  **do**  $\triangleright Q(\cdot)$  is defined in eqn (6) and  $\hat{y}_i = \text{softmax}(\hat{r}_i)$
  - 3:   For all  $i, w$  : calculate  $\nabla_{i,w}^{t-1} = \sum_{w' \in V_T} \frac{\partial Q(\hat{y}_1^{t-1}, \dots, \hat{y}_\ell^{t-1})}{\partial \hat{y}_i(w')} \frac{\partial \hat{y}_i(w')}{\partial \hat{r}_i(w)}$   $\triangleright$  using backpropagation
  - 4:   For all  $i, w$  : update  $\hat{r}_i^t(w) = \hat{r}_i^{t-1}(w) - \eta \nabla_{i,w}^{t-1}$   $\triangleright \eta$  is the step size
  - 5: **return**  $\arg \min_t Q(\text{softmax}(\hat{r}_1^t), \dots, \text{softmax}(\hat{r}_\ell^t))$
- 

	# tokens	# types	# sents
<b>BTEC zh→en</b>			
train	422k / 454k	3k / 3k	44,016
dev	10k / 10k	1k / 1k	1,006
test	5k / 5k	1k / 1k	506
<b>TED Talks de→en</b>			
train	4m / 4m	26k / 19k	194,181
dev-test2010	33k / 35k	4k / 3k	1,565
test2014	26k / 27k	4k / 3k	1,305
<b>WMT 2016 de→en</b>			
train	107m / 108m	90k / 78k	4m
dev-test2013&14	154k / 152k	20k / 13k	6003
test2015	54k / 54k	10k / 8k	2169

Table 1: Statistics of the training and evaluation sets; token and types are presented for both source/target languages.

source-to-target and target-to-source models, respectively. Decoding for the best translation under the above objective function leads to an intractable search problem, as the reverse model is global over the target language, meaning there is no obvious means of search with a greedy algorithm or similar.

**Discussion.** There are two important considerations on how best to initialise the relaxed optimisation in the above settings, and how best to choose the step size. As the relaxed optimisation problem is, in general, non-convex, finding a plausible initialisation is likely to be important for avoiding local optima. Furthermore, a proper step size is a key in the success of the EG-based and SGD-based optimisation algorithms, and there is no obvious method how to best choose its value. We may also adaptively change the step size using (scheduled) annealing or via the line search. We return to this considerations in the experimental evaluation.

## 5 Experiments

### 5.1 Setup

**Datasets.** We conducted our experiments on datasets with different scales, translating between Chinese→English using the BTEC corpus, and German→English using the IWSLT 2015 TED

Talks(Cettolo et al., 2014) and WMT 2016<sup>3</sup> corpora. The statistics of the datasets can be found in Table 1.

**NMT Models.** We implemented our continuous-optimisation based decoding method on top of the Mantidae toolkit<sup>4</sup> (Cohn et al., 2016), and using the *dynet* deep learning library<sup>5</sup> (Neubig et al., 2017). All neural network models were configured with 512 input embedding and hidden layer dimensions, and 256 alignment dimension, with 1 and 2 hidden layers in the source and target, respectively. We used a LSTM recurrent structure (Hochreiter and Schmidhuber, 1997) for both source and target RNN sequences. For the vocabulary, we use word frequency cut-off of 5, and words rarer than this were mapped to a sentinel. For the large-scale WMT dataset, we applied byte-pair encoding (BPE) method (Sennrich et al., 2016) to better handle unknown words.<sup>6</sup> For training our neural models, we use early stopping based on development perplexity, which usually occurs after 5-8 epochs.

**Evaluation Metrics.** We evaluated in terms of search error, measured using the model score of the inferred solution (either continuous or discrete), as well as measuring the end translation quality with case-insensitive BLEU (Papineni et al., 2002). The continuous cost measures  $-\frac{1}{|\hat{y}|} \log P_{\Theta}(\hat{y} | x)$  under the model  $\Theta$ ; the discrete model score has the same formulation, albeit using the discrete rounded solution  $y$  (see §3). Note the cost can be used as a tool for selecting the best inference solution, as well as assessing convergence, as we illustrate below.

---

<sup>3</sup><http://www.statmt.org/wmt16/translation-task.html>

<sup>4</sup><https://github.com/duyvuleo/Mantidae>

<sup>5</sup><https://github.com/clab/dynet>

<sup>6</sup>With BPE, the out of vocabulary rates on heldout data are  $< 1\%$ .

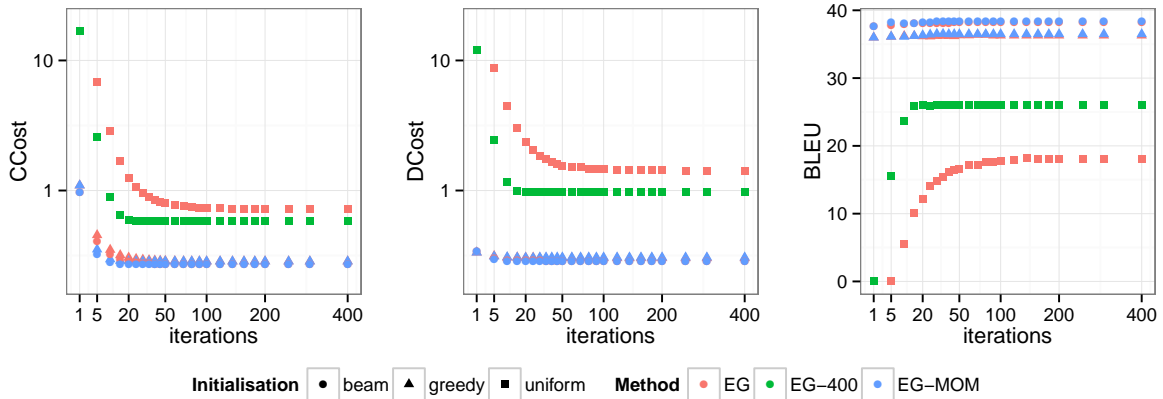


Figure 1: Analysis on effects of initialisation states (uniform vs. greedy vs. beam), step size annealing, momentum mechanism from BTEC zh→en translation. **EG-400**: EG algorithm with step size  $\eta = 400$  (otherwise  $\eta = 50$ ); **EG-MOM**: EG algorithm with momentum.

## 5.2 Results and Analysis

**Initialisation and Step Size.** As our relaxed optimisation problems are non-convex, local optima are likely to be a problem. We test this empirically, focusing on the effect that initialisation and step size,  $\eta$ , have on the inference quality.

For plausible initialisation states, we evaluate different strategies: *uniform* in which the relaxed variables  $\hat{\mathbf{y}}$  are initialised to  $\frac{1}{|V_T|}$ ; and *greedy* or *beam* whereby  $\hat{\mathbf{y}}$  are initialised based on an already good solution produced by a baseline decoder with greedy (gdec) or beam (bdec). Instead of using the Viterbi outputs as a one-hot representation, we initialise to the probability prediction vectors,<sup>7</sup> which serves to limit attraction of the initialisation condition, which is likely to be a local (but not global) optima.

Figure 1 illustrates the effect of initialisation on the EG algorithm, in terms of search error (left and middle) and translation quality (right), as we vary the number of iterations of inference. There is clear evidence of non-convexity: all initialisation methods can be seen to converge using all three measures, however they arrive at highly different solutions. Uniform initialisation is clearly not a viable approach, while greedy and beam initialisation both yield much better results. The best initialisation, beam, outperforms both greedy and beam decoding in terms of BLEU.

Note that the EG algorithm has fairly slow convergence, requiring at least 100 iterations, irrespective of the initialisation. To overcome this,

<sup>7</sup>Here, EG uses softmax normalization whereas SGD uses the pre-softmax vector.

we use momentum (Qian, 1999) to accelerate the convergence by modifying the term  $\nabla_{i,w}^t$  in Algorithm 1 with a weighted moving average of past gradients:

$$\nabla_{i,w}^{t-1} = \gamma \nabla_{i,w}^{t-2} + \eta \frac{\partial Q(\hat{\mathbf{y}}_1^{t-1}, \dots, \hat{\mathbf{y}}_\ell^{t-1})}{\partial \hat{\mathbf{y}}_i(w)}$$

where we set the momentum term  $\gamma = 0.9$ . The EG with momentum (**EG-MOM**) converges after fewer iterations (about 35), and results in marginally better BLEU scores. The momentum technique is usually used for SGD involving additive updates; it is interesting to see it also works in EG with multiplicative updates.

The step size,  $\eta$ , is another important hyperparameter for gradient based search. We tune the step size using line search over  $[10, 400]$  over the development set. Figure 1 illustrates the effect of changing step size from 50 to 400 (compare **EG** and **EG-400** with **uniform**), which results in a marked difference of about 10 BLEU points, underlining the importance of tuning this value. We found that EG with momentum had less of a reliance on step size, with optimal values in  $[10, 50]$ ; we use this setting hereafter.

**Continuous vs Discrete Costs.** Another important question is whether the assumption behind continuous relaxation is valid, i.e., if we optimise a continuous cost to solve a discrete problem, do we improve the discrete output? Although the continuous cost diminishes with inference iterations (Figure 1 left), and appears to converge, it is not clear whether this corresponds to a better discrete output (note that the discrete cost and BLEU

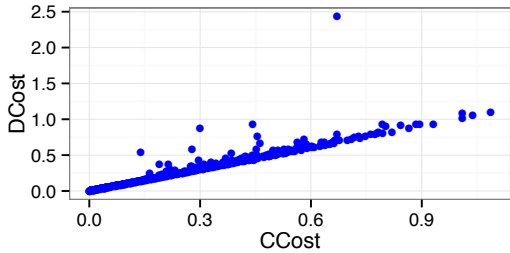


Figure 2: Comparing discrete vs continuous costs from BTEC zh→en translation, using the EG algorithm with momentum,  $\eta = 50$ . Each point corresponds to a sentence.

	BLEU	AvgLen
bdec <sub>left-to-right</sub>	26.69	20.73
filtered rerank	26.84	20.66
EGdec w/ beam init	27.34	20.73
full rerank	27.34	21.76
EGdec w/ rerank init	27.78	21.70

Table 2: The BLEU evaluation results with EG algorithm against 100-best reranking on WMT evaluation dataset.

scores do show improvements Figure 1 centre and right.) Figure 2 illustrates the relation between the two cost measures, showing that in most cases the discrete and continuous costs are identical. Linear relaxation fails only for a handful of cases, where the nearest discrete solution is significantly worse than it would appear using the continuous cost.

**EG vs SGD.** Both the EG and SGD algorithms are iterative methods for solving the relaxed optimisation problem with simplex constraints. We measure empirically their difference in terms of quality of inference and speed of convergence, as illustrated in Figure 3. Observe that SGD requires 150 iterations for convergence, whereas EG requires many fewer (50). This concurs with previous work on learning structured prediction models with EG (Globerson et al., 2007). Further, the EG algorithm consistently produces better results in terms of both model cost and BLEU.

**EG vs Reranking.** Reranking is an alternative method for integrating global factors into the existing NMT systems. We compare our EG decoding algorithm against the reranking approach with bidirectional factor where the N-best outputs of a left-to-right decoder is re-scored with the forced decoder operating in a right-to-left fashion. The

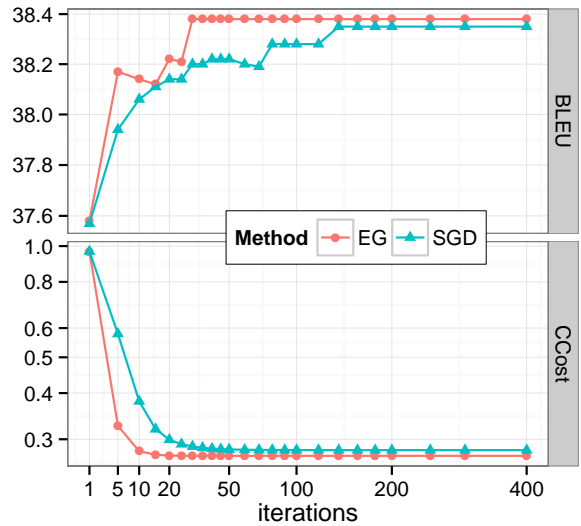


Figure 3: Analysis on convergence and performance comparing SOFTMAX and EG algorithms from BTEC zh→en translation. Both algorithms use momentum and step size 50.

results are shown in Table 2. Our EG algorithm initialised with the reranked output achieves the best BLEU score. We also compare reranking with EG algorithm initialised with the beam decoder, where for direct comparison we filter out sentences with length greater than that of the beam output in the  $k$ -best lists. These results show that the EG algorithm is capable of effectively exploiting the search space.

Beyond achieving similar or better translations to re-ranking, note that EG is simpler in implementation, as it does not require  $k$ best lists, weight tuning and so forth. Instead this is replaced with iterative gradient descent. The run-time of the two methods are comparable, when reranking uses modest  $k$ , however EG can be considerably faster when  $k$  is large, as is typically done to extract the full benefit from re-ranking. This performance difference is a consequence of GPU acceleration of the dense vector operations in EG inference.

**Computational Efficiency.** We also quantify the computational efficiency of the proposed decoding approach. Benchmarking on a GPU Titan X for decoding BTEC zh→en, the average time per sentence is 0.02 secs for greedy, 0.07s for beam=5, 0.11s for beam=10, and 3.1s for relaxed EG decoding, which uses an average of 35 EG iterations. The majority of time in the EG algorithm is in the forward and backward passes, taking 30% and 67% of the time, respectively. Our imple-

	BTEC	TEDTalks	WMT
	zh → en	de → en	de → en
$gdec_{\text{left-to-right}}$	35.98	23.16	24.41
$gdec_{\text{right-to-left}}$	35.86	21.95	23.59
$EGdec_{\text{greedy init}}$	36.34	23.28	24.63
+bidir.	<b>36.67</b>	<b>23.91</b>	<b>25.37<sup>†</sup></b>
+bilingual	<b>36.88<sup>†</sup></b>	<b>24.01<sup>†</sup></b>	<b>25.21</b>
$bdec_{\text{left-to-right}}$	38.02	23.95	26.69
$bdec_{\text{right-to-left}}$	37.38	23.13	26.11
$EGdec_{\text{beam init}}$	38.38	24.02	26.66
+bidir.	<b>39.13<sup>†</sup></b>	<b>24.72<sup>†</sup></b>	<b>27.34<sup>†</sup></b>
+bilingual	38.25	<b>24.60</b>	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, <sup>†</sup>: best performance on dataset.

mentation was not optimised thoroughly, and it is likely that it could be made significantly faster, which we defer to future research.

**Main Results.** Table 3 shows our experimental results across all datasets, evaluating the EG algorithm and its variants.<sup>8</sup> For the EG algorithm with greedy initialisation (top), we see small but consistent improvements in terms of BLEU. Beam initialisation led to overall higher BLEU scores, and again demonstrating a similar pattern of improvements, albeit of a lower magnitude, over the initialisation values.

Next we evaluate the capability of our inference method with extended NMT models, where approximate algorithms such as greedy or beam search are infeasible. With the **bidirectional** ensemble, we obtained the statistically significant BLEU score improvements compared to the unidirectional models, for either greedy or beam initialisation. This is interesting in the sense that the unidirectional right-to-left model always performs worse than the left-to-right model. However, our method with bidirectional ensemble is capable of combining their strengths in a unified setting. For the **bilingual** ensemble, we see similar effects, with better BLEU score improvements in most cases, albeit of a lower magnitude, over the bidirectional one. This is likely to be due to a disparity with the training condition for the models,

<sup>8</sup>Due to the space constraints, we report results for the EG algorithm only. See also translation examples in Figure 4.

which were learned independently of one another.

Overall, decoding in extended NMT models leads to performance improvements compared to baseline methods. This is one of the main findings in this work, and augurs well for its extension to other global model variants.

## 6 Related Work

Decoding (inference) for neural models is an important task; however, there is limited research in this space perhaps due to the challenging nature of this task, with only a few works exploring some extensions to improve upon them. The most widely-used inference methods include sampling (Cho, 2016), greedy and beam search (Sutskever et al., 2014; Bahdanau et al., 2015, *inter alia*), and reranking (Birch, 2016; Li and Jurafsky, 2016).

Cho (2016) proposed to perturb the neural model by injecting noise(s) in the hidden transition function of the conditional recurrent neural language model during greedy or beam search, and execute multiple parallel decoding runs. This strategy can improve over greedy and beam search; however, it is not clear how, when and where noise should be injected to be beneficial. Recently, Wiseman and Rush (2016) proposed beam search optimisation while *training* neural models, where the model parameters are updated in case the gold standard falls outside of the beam. This exposes the model to its past incorrect predicted labels, hence making the training more robust. This is orthogonal to our approach where we focus on the decoding problem with a pre-trained model.

Reranking has also been proposed as a means of global model combination: Birch (2016) and Li and Jurafsky (2016) re-rank the left-to-right decoded translations based on the scores of a right-to-left model, learning to more diverse translations. Related, Li et al. (2016) learn to adjust the beam diversity with reinforcement learning.

Perhaps most relevant is Snelleman (2016), performed concurrently to this work, who also proposed an inference method for NMT using linear relaxation. Snelleman’s method was similar to our SGD approach, however he did not manage to outperform beam search baselines with an encoder-decoder. In contrast we go much further, proposing the EG algorithm, which we show works much more effectively than SGD, and demonstrate how this can be applied to inference in an attentional



BTEC zh→en	
Source	我确定我昨天给旅馆打过电话并且做了预定。
Reference	i am sure that i called the hotel <b>yesterday</b> and made a reservation .
beam dec (l2r)	i 'm sure i called the hotel <b>reservation</b> and i made a reservation .
beam dec (r2l)	i 'm sure i <b>made this hotel reservation</b> and made a reservation .
rerank +bidir.	i 'm sure i called the hotel <b>reservation</b> and i made a reservation .
rerank +biling.	i 'm sure i called the hotel <b>reservation</b> and i made a reservation .
EGdec	i 'm sure i called the hotel <b>yesterday</b> and i made a reservation .
+bidir.	i 'm sure i called the hotel <b>yesterday</b> and i made a reservation .
+biling.	i 'm sure i called the hotel <b>yesterday</b> and i made a reservation .

---

TED Talks de→en	
Source	wir sind doch alle gute bürger der sozialen medien , bei denen die währung neid ist . stimmt ' s ?
Reference	i mean , we 're all <b>good citizens</b> of social media , are n't we , where the currency is envy ?
beam dec (l2r)	we 're all <b>great UNK</b> of social media , where the currency is envy . right ?
beam dec (r2l)	we 're all <b>good citizens</b> in social media , which is where that is envy . right ?
rerank +bidir.	we 're all <b>good citizens</b> of social media , where the currency is envy . right ?
rerank +biling.	we 're all <b>good citizens</b> of social media , where the currency is envy . right ?
EGdec	we 're all <b>great UNK</b> of social media , where the currency is envy . right ?
+bidir.	we 're all <b>good UNK</b> of social media , where the currency is envy . right ?
+biling.	we 're all <b>good citizens</b> of social media , where the currency is envy . right ?

---

WMT de→en	
Source	neben dem wm-titel 2007 und dem gewinn der champions league 2014 holte er 2008 ( hsg nordhorn ) und 2010 ( tbv lemgo ) den ehf-pokal .
Reference	besides the <b>2007 world championship</b> he also won the champions league in may and the ehf-cup in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) .
beam dec (l2r)*	in addition to the <b>title 2007 in 2007</b> and the win of the champions league 2014 in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) , he won the ehf cup .
beam dec (r2l)	in addition to the <b>world championship title 2007</b> and winning the champions league in 2014 , he won the ehf-cup in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) .
EGdec	in addition to the <b>title 2007 in 2007</b> and the win of the champions league 2014 in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) , he won the ehf cup .
+bidir.	in addition to the title <b>championship in 2007</b> and the win of the champions league 2014 in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) , he won the ehf cup .
+biling.	in addition to the <b>world title title 2007</b> and the win of the champions league 2014 in 2008 ( hsg nordhorn ) and 2010 ( tbv lemgo ) , he won the ehf cup .

Figure 4: Translation examples generated by the models. \*: reranking with bidirectional (+bidir.) and bilingual (+biling.) produced the same translation string.

encoder-decoder. Moreover, we demonstrate the utility of related optimisation for inference over global ensembles of models, resulting in consistent improvements in search error and end translation quality.

Recently, relaxation techniques have been applied to deep models for training and inference in text classification (Belanger and McCallum, 2016; Belanger et al., 2017), and fully differentiable training of sequence-to-sequence models with scheduled-sampling (Goyal et al., 2017). Our work has applied the relaxation technique specifically for *decoding* in NMT models.

## 7 Conclusions

This work presents the first attempt in formulating decoding in NMT as a continuous optimisation problem. The core idea is to drop the integrality (i.e. one-hot vector) constraint from the prediction variables and allow them to have soft assignments within the probability simplex while minimising the loss function produced by the neural

model. We have provided two optimisation algorithms – exponentiated gradient (EG) and stochastic gradient descent (SGD) – for optimising the resulting contained optimisation problem, where our findings show the effectiveness of EG compared to SGD. Thanks to our framework, we have been able to decode when intersecting left-to-right and right-to-left as well as source-to-target and target-to-source NMT models. Our results show that our decoding framework is effective and leads to substantial improvements in translations generated from the intersected models, where the typical greedy or beam search algorithms are not applicable.

This work raises several compelling possibilities which we intend to address in future work, such as improving decoding speed, integrating additional constraints such as word coverage and fertility into *decoding*,<sup>9</sup> and applying our method to other intractable structured prediction problems.

<sup>9</sup>These constraints have only been used for training in the previous works (Cohn et al., 2016; Mi et al., 2016).

## Acknowledgments

We thank the reviewers for valuable feedbacks and discussions. Cong Duy Vu Hoang is supported by Australian Government Research Training Program Scholarships at the University of Melbourne, Australia. This work was supported by the Australian Research Council through their Discovery and Future Fellowship programmes.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. of 3rd International Conference on Learning Representations (ICLR2015)*.
- D. Belanger, B. Yang, and A. McCallum. 2017. [End-to-End Learning for Structured Prediction Energy Networks](#). In *Proceedings of the 34th International Conference on International Conference on Machine Learning - Volume 48, ICML'17*.
- David Belanger and Andrew McCallum. 2016. [Structured prediction energy networks](#). In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16*, pages 983–992. JMLR.org.
- Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. [A maximum entropy approach to natural language processing](#). *Comput. Linguist.*, 22(1):39–71.
- Rico Sennrich; Barry Haddow; Alexandra Birch. 2016. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers. Berlin, Germany*.
- M. Cettolo, J. Niehues, S. Stuker, L. Bentivogli, and M. Federico. 2014. Report on the 11th IWSLT Evaluation Campaign. In *Proc. of The International Workshop on Spoken Language Translation (IWSLT)*.
- K. Cho. 2016. [Noisy Parallel Approximate Decoding for Conditional Recurrent Language Model](#). *ArXiv e-prints*.
- Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating Structural Alignment Biases into an Attentional Neural Translation Model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.
- Amir Globerson, Terry Y. Koo, Xavier Carreras, and Michael Collins. 2007. Exponentiated Gradient Algorithms for Log-linear Structured Prediction. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pages 305–312, New York, NY, USA. ACM.
- Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2017. [Differentiable scheduled sampling for credit assignment](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL'17, Vancouver, Canada.
- A. Graves. 2013. [Generating Sequences With Recurrent Neural Networks](#). *ArXiv e-prints*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.
- Jyrki Kivinen and Manfred K. Warmuth. 1997. Exponentiated Gradient Versus Gradient Descent for Linear Predictors. *Inf. Comput.*, 132(1):1–63.
- J. Li and D. Jurafsky. 2016. [Mutual Information and Diverse Decoding Improve Neural Machine Translation](#). *ArXiv e-prints*.
- J. Li, W. Monroe, and D. Jurafsky. 2016. [A Simple, Fast Diverse Decoding Algorithm for Neural Generation](#). *ArXiv e-prints*.
- Z. C. Lipton, J. Berkowitz, and C. Elkan. 2015. [A Critical Review of Recurrent Neural Networks for Sequence Learning](#). *ArXiv e-prints*.
- Adam Lopez and Philip Resnik. 2006. Word-based alignment, phrase-based translation: Whats the link. In *Proceedings of 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*.
- Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage Embedding Models for Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.
- G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji, L. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta, and P. Yin. 2017. [DyNet: The Dynamic Neural Network Toolkit](#). *ArXiv e-prints*.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings*

of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145 – 151.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Emanuel Snelleman. 2016. Decoding neural machine translation using gradient descent. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden.

David Sontag. 2010. *Approximate Inference in Graphical Models using LP Relaxations*. Ph.D. thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14*, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Taro Watanabe and Eiichiro Sumita. 2002a. Bidirectional Decoding for Statistical Machine Translation. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1, COLING '02*, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taro Watanabe and Eiichiro Sumita. 2002b. Bidirectional decoding for statistical machine translation. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.