Energy-Based Reranking: Improving Neural Machine Translation Using Energy-Based Models

Sumanta Bhattacharyya, Amirmohammad Rooshenas*
Department of Computer Science, College of Computing and Informatics
University of North Carolina Charlotte
{sbhatta9,rooshenas}@uncc.edu

Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum
College of Information and Computer Science, University of Massachusetts Amherst
{snaskar,simeng,miyyer,mccallum}@cs.umass.edu

Abstract

The discrepancy between maximum likelihood estimation (MLE) and task measures such as BLEU score has been studied before for autoregressive neural machine translation (NMT) and resulted in alternative training algorithms (Ranzato et al., 2016; Norouzi et al., 2016; Shen et al., 2016; Wu et al., 2018). However, MLE training remains the de facto approach for autoregressive NMT because of its computational efficiency and stability. Despite this mismatch between the training objective and task measure, we notice that the samples drawn from an MLE-based trained NMT support the desired distribution – there are samples with much higher BLEU score comparing to the beam decoding output. To benefit from this observation, we train an energy-based model to mimic the behavior of the task measure (i.e., the energy-based model assigns lower energy to samples with higher BLEU score), which is resulted in a re-ranking algorithm based on the samples drawn from NMT: energy-based re-ranking (EBR). We use both marginal energy models (over target sentence) and joint energy models (over both source and target sentences). Our EBR with the joint energy model consistently improves the performance of the Transformer-based NMT: +3.7 BLEU points on IWSLT'14 German-English, +3.37 BELU points on Sinhala-English, +1.4 BLEU points on WMT'16 English-German tasks.

1 Introduction

Autoregressive models are widely used for neural machine translation (NMT) (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). The autoregressive factorization provides a tractable likelihood computation as well as efficient sampling. The former results in the effective maximum likelihood estimation (MLE) for training the parameters of NMT models. However, optimizing likelihood does not guarantee an improvement in task-based measures such as the BLEU score, which has motivated directly optimizing task measures with reinforcement learning (Ranzato et al., 2016; Norouzi et al., 2016; Shen et al., 2016; Bahdanau et al., 2017; Wu et al., 2018). However, for NMT, these training algorithms are often used in conjunction with MLE training (Wu et al., 2018) or as fine-tuning (Choshen et al., 2020).

Interestingly, we observe that samples drawn from an NMT model trained using MLE may have higher quality (measured with BLEU) than the outputs of beam search. In particular, we draw 100 target samples for each source sentence from an NMT model trained using MLE on the IWSLT'14 German-English task, and observe that an oracle ranker – i.e. $\text{argmax}_{\mathbf{y}\sim P_{\text{NMT}}(\mathbf{y}|\mathbf{x})}\ \text{BLEU}(., \mathbf{y}^*)$, where $(\mathbf{x}, \mathbf{y}^*)$ is the pair of source and gold target sentence – achieves the high score of 67.54, while the beam decoding achieves 33.87. We also look at the distribution of the Spearman rank correlation coefficient of the drawn samples with respect to the log probability score of the baseline NMT (BaseNMT). Figure 1 shows that there is no strong correlation between the BLEU score ranking of samples and the log probability score ranking for the majority of source sentences; thus, maximum a priori (MAP) decoding is incapable of finding the desired output. In parallel to our study, Eikema and Aziz (2020) also report that the mismatch regarding MLE training of autoregressive models is attributable to the distribution of the probability mass rather than the parameter estimation, resulting in a poor MAP decoding.

Instead of looking for an alternate algorithm for parameter estimation, these results motivate us to explore training a parametric approximation of the metric, here BLEU score: $\omega_\theta(\mathbf{y}, \mathbf{x}) \approx \text{BLEU}(\mathbf{y}, \mathbf{y}^*)$. Therefore the decoding becomes:

*Amirmohammad Rooshenas is the corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4528–4537
August 1–6, 2021.

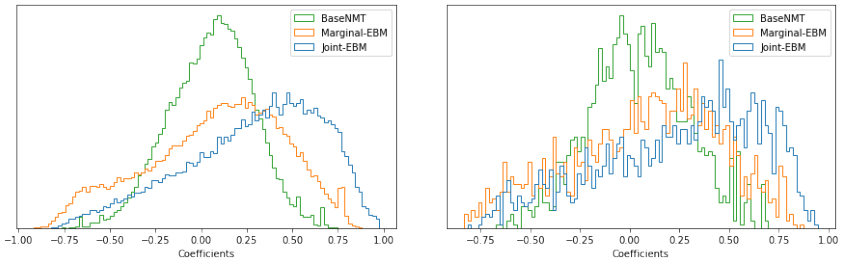

Figure 1: Distribution of the Spearman rank-order correlation coefficients for the training data (left) and test data (right) of the IWSLT'14 German-English task.

$\text{argmax}_{\mathbf{y}\sim P_{\text{NMT}}(.|\mathbf{x})}\, \omega_\theta(\mathbf{y}, \mathbf{x})$.

We use energy-based models (EBMs) to parameterize $\omega_\theta(\mathbf{y}, \mathbf{x})$. EBMs (LeCun et al., 2006) are general parametric models that assign a scalar energy value to each configuration of input variables, thus defining an unnormalized probability distribution. Although computing the partition function is intractable for general EBMs, we only require the relative energy of the sampled sentences from the BaseNMT model, thus canceling out the normalization constant. In this paper we use two different energy-based models: marginal energy model (Marginal-EBM) defined only over target sentences and joint energy model (Joint-EBM) defined over both source and target sentences.

Figure 1 also shows the correlation coefficient of the energy ranking and BLEU score using both Marginal-EBM and Joint-EBM. The shift in the coefficient distribution suggests that decoding based on energy scores results in better BLEU scores compared to decoding based on the log probability scores of the BaseNMT model. Also we observe that Joint-EBM works better than using Marginal-EBM as Joint-EBM better captures the correlation of source and target sentences, while Marginal-EBM is not directly conditioned on the source sentence.

In this paper, we describe how to train EBMs[1] to achieve the desired ranking. Our energy ranker consistently improves the performance of Transformer-based NMT on German-English, Romanian-English and Italian-English tasks from IWSLT'14, the French-English task from IWSLT'17, German-English task from WMT'14, and English-German task from WMT'16, as well as the low-resource Sinhala-English and Nepali-English tasks described in the FLoRes dataset (Guzmán et al., 2019).

[1]The code is available at https://github.com/rooshenas/ebr_mt

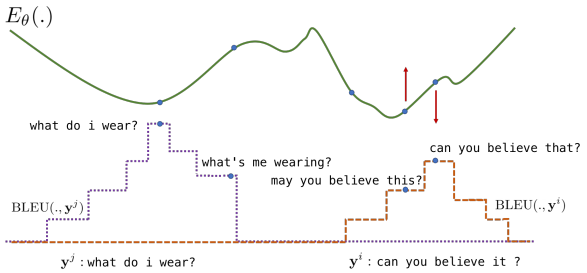

Figure 2: The EBM is trained such that its energy landscape is consistent with the BLEU score. Marginal-EBM is not conditioned on the source sentence, thus each local region is trained to have similar ranking as that BLEU score for the samples in the region.

2 Energy-Based Reranking

Using EBM E_θ to reweight the samples from an NMT defines a new probability distribution over the output sentences (see Grover et al. (2019)): $P_\theta(\mathbf{y}|\mathbf{x}) \propto P_{\text{NMT}}(\mathbf{y}|\mathbf{x}) \exp(\frac{-E_\theta(\mathbf{y},\mathbf{x})}{T})$, where T is temperature. The ideal re-ranker requires an EBM with the energy function $E_\theta(\mathbf{y}, \mathbf{x})$ such that $P_\theta(\mathbf{y}|\mathbf{x})$ and $\text{BLEU}(\mathbf{y}, \mathbf{y}^i)$ have similar modes for all $(\mathbf{x}^i, \mathbf{y}^i) \in \mathcal{D}$, where $\mathcal{D}$ is an empirical data distribution. To train θ we use rank-based training (Rohanimanesh et al., 2011; Rooshenas et al., 2018, 2019). Rank-based training enforces that the samples from $P_\theta(.)$ have similar ranking with respect to both the energy score and task measure (see Figure 2).

To sample from $P_\theta(\mathbf{y}|\mathbf{x})$, we sample k sentences from $P_{\text{NMT}}(\mathbf{y}|\mathbf{x})$ using multinomial sampling from locally normalized distributions over the output and reweight the samples based on the energy network $\exp(\frac{-E_\theta(\mathbf{y},\mathbf{x})}{T})$. Then we resample two sentences, $\mathbf{y}_1$ and $\mathbf{y}_2$, from the renormalized set, which defines a conditional distribution: $P^i(\mathbf{y}|\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{y},\mathbf{x})/T)}{\sum_k \exp(-E_\theta(\mathbf{y}_k,\mathbf{x})/T)}$ (a similar sampling approach has been used in Deng et al. (2020)). Now we train the energy model such that the ranking of $\mathbf{y}_1$ and $\mathbf{y}_2$ with respect to the energy model

is consistent with their ranking with respect to the task metric, BLEU score.

In general, we assume $\mathbf{y}_h$ is the sentence with the higher BLEU score and $\mathbf{y}_l$ is the sentence with with the lower BLEU score. Therefore, the training objective of $E_\theta(\mathbf{y}, \mathbf{x})$ becomes:

$$M = \alpha(\text{BLEU}(\mathbf{y}_h, \mathbf{y}_i) - \text{BLEU}(\mathbf{y}_l, \mathbf{y}_i))$$
$$\xi(\mathbf{y}_i, \mathbf{x}_i) = M + E_\theta(\mathbf{y}_h, \mathbf{x}_i) - E_\theta(\mathbf{y}_l, \mathbf{x}_i)$$
$$\min_\theta \sum_{(\mathbf{y}_i, \mathbf{x}_i) \in \mathcal{D}} \max(\xi(\mathbf{y}_i, \mathbf{x}_i), 0). \quad (1)$$

Where $\xi(\mathbf{y}_i, \mathbf{x}_i)$ is the margin violation and α is the margin weight. Algorithm 1 outlines the whole training procedure.

If we define the energy only over sentences of the target language, $E_\theta(\mathbf{y})$, we can share the energy-model among multiple language pairs with the same target language. In this case we have to, first, sample the language l from our language set and then sample a sentence pair from the selected language training set $\mathcal{D}_l$. The probability of selecting a language is proportional to the number of sentences in its training set.

Algorithm 1 Rank-Based Training of EBM

$P_{\text{NMT}}(y|x) \leftarrow$ Pretrained NMT
$E_\theta(\mathbf{y}, \mathbf{x}) \leftarrow$ Energy based models for target sentences
repeat
  $\mathcal{L} \leftarrow 0$.
  for batch size **do**
    Sample $(\mathbf{x}_i, \mathbf{y}_i)$ from $\mathcal{D}$
    $Y_i \leftarrow$ collect k samples from $P_{\text{NMT}}(.|\mathbf{x}_i)$
    $P^i(\mathbf{y}) \leftarrow \frac{\exp(-E_\theta(\mathbf{y},\mathbf{x})/T)}{\sum_{\mathbf{y} \in Y_i} \exp(-E_\theta(\mathbf{y},\mathbf{x})/T)}$ for $\mathbf{y} \in Y_i$
    $\mathbf{y}_1, \mathbf{y}_2 \leftarrow$ samples from $P_i(\mathbf{y})$
    $\mathbf{y}_h \leftarrow \text{argmax}_{\mathbf{y}_1,\mathbf{y}_2}\{\text{BLEU}(\mathbf{y}_1, \mathbf{y}_i), \text{BLEU}(\mathbf{y}_2, \mathbf{y}_i)\}$
    $\mathbf{y}_l \leftarrow \text{argmin}_{\mathbf{y}_1,\mathbf{y}_2}\{\text{BLEU}(\mathbf{y}_1, \mathbf{y}_i), \text{BLEU}(\mathbf{y}_2, \mathbf{y}_i)\}$
    $M \leftarrow \alpha(\text{BLEU}(\mathbf{y}_h, \mathbf{y}_i) - \text{BLEU}(\mathbf{y}_l, \mathbf{y}_i))$
    $\mathcal{L} \leftarrow \mathcal{L} + \max(M + E_\theta(\mathbf{y}_h, \mathbf{x}_i) - E_\theta(\mathbf{y}_l, \mathbf{x}_i), 0)$
  end for
  $\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}$ // λ is learning rate
until Convergence

In this paper, we use BERT (Devlin et al., 2019) to parameterize both $E_\theta(\mathbf{y}, \mathbf{x})$ and $E_\theta(\mathbf{y})$. Section 4.3 and 4.4 discuss the construction of E_θ in detail.

3 Related Work

Grover et al. (2019) show that importance weights can be used to make generative models better fit the desired data distribution: $p_\theta(\mathbf{y}) \propto q(\mathbf{y})\omega_\theta(\mathbf{y})$, where $q(\mathbf{y})$ is a generative model that we can efficiently take samples from and $\omega_\theta(\mathbf{y})$ is the importance weight function. The importance weights can be determined using a discriminator that differentiates the generated samples from the target data. Rosenfeld et al.; Parshakova et al. (2001; 2019) define $q(\mathbf{y})$ as autoregressive model and $\omega_\theta(\mathbf{y})$ using a log-linear model: $\omega_\theta(\mathbf{y}) = \exp(\theta^T \phi(\mathbf{y}))$, where $\phi(\mathbf{y})$ is the vector of sufficient statistics (features) evaluated at $\mathbf{y}$. The log-linear model simplifies training the parameters θ: $\nabla_\theta p_\theta(\mathbf{y}) = \sum_{\mathbf{y} \in \mathcal{D}} \phi(\mathbf{y}) - \mathbb{E}_{\hat{\mathbf{y}} \sim p_\theta(.)} \phi(\hat{\mathbf{y}})$. The expectation term can be estimated using rejecting sampling or importance sampling given the proposal distribution q. Deng et al. (2020) extend this approach for text generation by using unrestricted EBMs instead of log-linear models: $\omega_\theta(\mathbf{y}) = \exp(-E_\theta(\mathbf{y}))$. They train the EBM using noise contrastive estimation (Gutmann and Hyvärinen, 2010). We find this less suitable for re-ranking in the translation tasks (see Section 4).

Discriminative re-ranking was first introduced by Shen et al. (2004) for improving the performance of machine translation (MT). They have trained a linear separator using the perceptron learning algorithm to distinguish the top r translations from the rest of the translations in the n-best possible outputs. The features for the discriminator are extracted from both source and target sentences. Mizumoto and Matsumoto (2016) combine the score of MT and the linear model using more complex syntactical features to re-rank the target sentences. Here, we rely on the features learned by BERT, and given the high capacity of the energy model, we train the energy model to respect the ranking of every pair of samples.

Gulcehre et al. (2017) describe using language model (LM) to improve the performance of NMT using shallow and deep fusion. Shallow models combine the marginal probability of predicting each word in NMT and LM: $\log P_{\text{NMT}}(y_i|y_{<i}) + \lambda \log P_{\text{LM}}(y_i|y_{<i})$, while deep fusion concatenates the hidden states of two models before predicting each word and uses parallel data to fine-tune the weights. Similar to deep fusion, Domhan and Hieber (2017) feed the unnormalized output of LM to the decoder of NMT. Domhan and Hieber (2017) jointly train the LM and NMT using monolingual target-side data and parallel data, respectively. Sennrich et al. (2016a) augment the parallel training data with monolingual data with the target language and back-translation.

Re-ranking with LM has also been explored by Ng et al. (2019), where they decode the output

based on $\log p(y|x) + \lambda_1 \log p(x|y) + \lambda_2 \log p(y)$, where $p(y|x)$ is the direct model provided by NMT, $p(x|y)$ is computed via back-translation and $p(y)$ is an LM. Our approach differs from the previous methods that use LMs for re-ranking as we train our energy-based model to be consistent with the task measure instead of using pre-trained LMs. In our experiments, we only explore the effect of using the direct model plus LM, nevertheless, back-translation can also be added into our model for further improvement.

Recently, Salazar et al. (2020) use masked language models (MLM) such as BERT to score hypotheses from NMT. Salazar et al. (2020) describe the score of a MLM as pseudo-log-likelihood score (PLL). To calculate PLL score of a sentence, each token w_i in the sentence is sequentially masked, which allows the calculation of $\log p(w_i|\mathbf{w}_{\setminus i})$ from the output of the MLM. The normalized pseudo-log-probability of the sentence is the average of log-probability of the masked words given the rest of the words in the sentence: $\frac{1}{N}\sum_{i=1}^{N} \log p(w_i|\mathbf{w}_{\setminus i})$, where N is the length of the sentence. We use this approach as one of our baselines.

In parallel to our work, Guo et al. (2020) proposes using two different BERT models as an encoder of the source language (X-BERT) and a decoder of the target language (Y-BERT). Guo et al. (2020) add an extra trainable encoder-decoder adaption module followed by a feed-forward module to each layer of the decoder and a feed-forward module to each layer of the encoder. (Please see Guo et al. (2020) for more detail on the architecture.) For fine-tuning XY-BERT for translation tasks, Guo et al. (2020) keep all XY-BERT's parameters fixed except the parameters of the new modules, and use mask-predict decoding (Ghazvininejad et al., 2019) for running test-time inference. Guo et al. (2020) report a significant improvement over prior non-autoregressive models and superior performance comparing to autoregressive methods on IWSLT'14 German-English task. Their finding is consistent with our improvement using the pretained BERT model. However, our Joint-EBM model is a different way of using BERT for translation, which does not require separate BERT models for source and target language. Please see Section 4.9 for a detailed comparison.

Finally, other works also discuss using BERT to improve the performance of NMT. Clinchant et al. (2019) describe initializing the embedding or the whole encoder with BERT's parameters. Zhu et al. (2020) use an attention model to incorporate the output of BERT into encoder and decoder of NMT. In our approach, we use BERT as an external energy-based ranker.

4 Experiments

4.1 Datasets

We use German-English (De→En), Romanian-English (Ro→En) and Italian-English (It→En) from IWSLT'14 datasets and French-English (Fr→En) from IWSLT'17 translation tasks. We also use IWSLT'14 English-German (En→De) to show that the proposed method can be expanded to translation tasks with a different target language. All sentences were preprocessed using byte-pair-encoding (Sennrich et al., 2016b). For all language pairs in IWSLT'14 and IWSLT'17, we merge the test datasets tst2010, tst2011, tst2012 and report BLEU on the merged dataset. We also use German-English (De→En) from the WMT'14 and English-German (En→De) from WMT'16 translation tasks.

Finally, we use low-resource translation tasks Nepali-English (Ne→En) and Sinhala-English (Si→En) from FLoRes (Guzmán et al., 2019) translation tasks. We follow dataset distribution and preprocessing steps described in Guzmán et al. (2019) using the FLoRes implementation. FLoRes dataset contains development (dev), devtest and test dataset for both language pairs. Similar to Guzmán et al. (2019) we use the devtest dataset for all our evaluations.

4.2 Base Model

We use the Transformer[2](Vaswani et al., 2017) as our BaseNMT. Our Transformer architecture includes six encoder and six decoder layers, and the number of attention heads, embedding dimension and inner-layer dimension are 8, 512 and 4096, respectively. We use dropout, weight decay, label smoothing to regularize our models. We use layer normalization and early stopping. Models are optimized using Adam (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-8}$ and we use the same learning rate scheduler as Ott et al. (2019). We trained our models on 1 Nvidia TITANX GPU.

[2]We use the implementation in Opennmt (Klein et al., 2017) and Fairseq (Ott et al., 2019) toolkits.

Table 1: BLEU score comparison for IWSLT, FLoRes, and WMT (indicated using *) tasks.

	De→En	Fr→En	It→En	Ro→En	Si→En	Ne→En	En→De	De→En*	En→De*
BaseNMT + Beam	33.87	31.50	32.08	33.21	7.10	6.07	28.83	30.13	28.84
BaseNMT + Sample	33.98	31.59	32.22	33.64	7.19	6.44	28.85	30.28	28.89
BaseNMT + LM	34.25	31.56	32.52	33.01	7.11	6.02	28.91	30.31	28.93
BaseNMT + MLM	34.42	32.13	33.68	33.85	7.70	7.21	30.12	30.61	28.98
NCE-EBR	34.47	32.00	32.89	32.23	7.98	7.36	28.22	31.42	29.03
Marginal-EBR	35.68	33.77	34.00	34.48	8.62	7.26	30.82	31.65	29.14
Shared-EBR	35.75	33.80	34.14	34.65	10.29	9.25	-	-	-
Conditional-EBM	**37.58**	**35.02**	**36.05**	**37.19**	**10.47**	**9.82**	**30.97**	**32.21**	**30.23**
Oracle	67.54	68.43	71.77	73.95	14.71	11.91	52.14	50.89	45.15

Table 2: Shared-EBR performance for Si→En by training with difference sets of language pairs.

BaseNMT	+ Si→En	+ De→En	+ Fr→En	all
7.10	8.62	9.30	9.76	**10.29**

4.3 Marginal-EBM

To construct the energy network over the sentences of the target language, we use a pretrained BERT (Devlin et al., 2019) from Huggingface (Wolf et al., 2019) as our pretrained language model and project the hidden state of BERT for each output token into a scalar value and define the energy value of the target sentence as the average of the scalar values. We use the *BERT-base uncased* model with 12 encoder layers, 768 hidden state dimension, 12 attention heads and 110M parameters. For the projection layer, we use a 2-layer MLP with 256 hidden variables. In our experiments, we only train the parameters of the projection layer and the rest of BERT's parameters remain frozen. We use margin weight of $\alpha = 10$ and temperature $T = 1000$ for our experiments. We regularize the projection layer using L2 regularization. Models are optimized using Adam (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-8}$ and a learning rate of 0.01. We run all experiments on 1 Nvidia TESLA M40 GPU.

4.4 Joint-EBM

Joint-EBM must assign a score to a pair of sentences from source and target languages, so to construct the Joint-EBM, similar to Marginal-EBM, we need a Joint-BERT. We feed the sentence pairs from source and target languages jointly to BERT, thus the name Joint-BERT. Since Joint-BERT has not been pre-trained to accept pairs of sentences from two different languages, we fine-tune it for 12 epochs using the input format of `[CLS]Source[SEP]Target[SEP]` with the pairs of source and target sentences for each translation task. For fine-tuning, we only mask the tokens of the target sentence. For all translation tasks we use the BERT-Base, Multilingual Cased model with 12 encoder layers, 768 hidden state dimension, 12 attention heads and 110M parameters. After fine-tuning Joint-BERT, we follow the same architecture as Marginal-EBM for the Joint-EBM.

4.5 Methods

As the main baseline, we run beam decoding with a beam size of five over the trained BaseNMT (BaseNMT+Beam). We also use the samples drawn from the BaseNMT and report the BLEU score of the sample with the highest log-probability score on BaseNMT (BaseNMT+Sample). For all methods we use 100 target samples for each source sentence. BaseNMT+LM draws samples from the BaseNMT and uses $\log P_{\text{NMT}}(\mathbf{y}|\mathbf{x}) + \lambda \log P_{LM}(\mathbf{y})$ to rank the samples ($\lambda = 0.01$ out of the set of $\{0.001, 0.01, 0.1\}$ results in the best performance).

In our BaseNMT+LM baseline, we use pretrained language model to calculate $\log P_{LM}(\mathbf{y})$. For the {De, Fr, It, Ro, Si, Ne}→En tasks, we use a pretrained Transformer-XL (Dai et al., 2019) *transfo-xl-wt103* and for the En→De task we use a pretrained XLM (Lample and Conneau, 2019) *xlm-mlm-ende-1024* from Huggingface (Wolf et al., 2019). BaseNMT+MLM is similar to BaseNMT+LM but it uses $\log P_{\text{NMT}}(\mathbf{y}|\mathbf{x}) + \lambda \log P_{MLM}(\mathbf{y})$, where P_{MLM} is the average pseudo-log-probability of sample $\mathbf{y}$ calculated using BERT. We use the same architecture of BERT as Marginal-EBM, but we fine-tuned BERT for MLM over the target sentences in training sets for 10 epochs. We tuned λ similar to BaseNMT+LM.

EBR is our method that uses rank-based training for EBMs. We explore EBR with Marginal-EBM (Marginal-EBR) and Joint-EBM (Conditional-EBR). We also use noise-contrastive estimation to train our Marginal-EBM, similar to Deng et al. (2020), which we refer to as NCE-EBR. Next,

we have Shared-EBR that trains single Marginal-EBM for the tasks with the same target language. Shared-EBR is only trained on IWSLT and FLoRes tasks with English target. For this method, we first sample a translation task and then sample a batch from that task and follow Algorithm 1 for the training of the Marginal-EBM. Finally, as an upper bound for the best achievable result, we also extract the translations from the sample that are closest to the gold data (based on BLEU score).

4.6 Results

Table 1 shows the performance of the described methods for IWSLT, FLoRes, and WMT translation tasks.[3] BaseNMT+Sample achieves a better score than beam decoding suggesting that our multinomial sampling supports the modes of the distribution defined by the BaseNMT. Similarly, oracle values are high, indicating that the samples also support the desired distribution. This satisfies the necessary condition for $P_\theta(\mathbf{y}|\mathbf{x}) \propto P_{\text{NMT}}(\mathbf{y}|\mathbf{x}) \exp(-E_\theta(\mathbf{y}, \mathbf{x})/T)$ to be closer to the desired distribution. Re-ranking with a language model using BaseNMT+LM improves over BaseNMT+Sample for De→En, Fr→En, It→En, and En→De, but fails on Ro→En, Si→En, and Ne→En. However, in all of these tasks, the difference between BaseNMT+Sample and BaseNMT+LM is not substantial. BaseNMT+MLM is consistently better than BaseNMT+LM. The performance of BaseNMT+MLM is attributable to PLL scoring, as the encoder has the global information over the sentence. Marginal-EBR performs considerably better than BaseNMT+{Beam, Sample, LM, MLM} and better than NCE-EBR on all tasks except on Ne→En, where NCE-EBR outperforms Marginal-EBR. The main advantage of Marginal-EBR over NCE-EBR is the use of only sampled data instead of gold data for training. See Section 4.7 for detailed discussion.

Shared-EBR has a significant improvement over the Marginal-EBR, especially it improves the low-resource task of Si→En by more than 2 BLEU points. For this task, we also show that how using more language pairs in training improves performance (Table 2).

Conditional-EBR outperforms Shared-EBR on all tasks. The performance of Conditional-EBR is due to the use of Joint-EBM model, which enables the model to define different energy landscapes for different source sentences. Therefore, samples from the target language are more separable given the source sentence, while Marginal-EBM may not distinguish target sentences for different source sentences.

The translation improvement of using EBR on IWSLT and FLoRes translation tasks are more considerable than the improvement of using EBR on WMT tasks. We believe that pre-trained BERT helps low-resource tasks more than large-scale translation tasks.

[3] We use SacreBLEU (Post, 2018) as a consistent BLEU implementation for all of our experiments.

Table 3: The effect of using gold data in the ranking objective for Marginal-EBR.

γ	0.0	0.25	0.75	1.0
De→En	35.68	35.00	34.20	33.75
Fr→En	33.77	33.15	31.65	30.82

Table 4: Effect of Entropy Regularization on IWSLT'14 DE-EN

	Regularization	No Regularization
BaseNMT + Beam	33.96	33.87
Conditional-EBR	**37.88**	37.58
Oracle	68.21	67.54

4.7 Effect of Using Gold Data

Noise-contrastive estimation (NCE) trains the energy model using a discriminative training to distinguish gold data from the sampled data (Gutmann and Hyvärinen, 2010; Deng et al., 2020). In contrast to the NCE-EBR, EBR does not directly use gold data in the training of the EBM, but only exploit it to determine the rank of two points as well as the margin. To show that our approach is effective, we introduce parameter γ as the percentage of the time that we can use gold data as one of the points (for example, $\mathbf{y}_h$ in Algorithm 1). Table 3 shows the results for both De→En and Fr→En tasks using Marginal-EBR. As we increase the value of γ, the performance of Marginal-EBR drops. The main reason is that BaseNMT rarely produces the exact correct translation in the sample set, thus learning the ranking with respect to the gold data is not very informative. When the γ is zero, the Marginal-EBM learns to re-rank the samples with respect to their distance to the gold data.

4.8 Regularized Training

We hypothesize that the performance of EBR improves as we increase the support of the base distribution toward the mode of the true distribution. To show that we add an entropy regularization term to the likelihood training of BaseNMT:

$$\max_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{i} \log p(y_i|\mathbf{y}_{<i},\mathbf{x}) - \beta \sum_{i} p(y_i)\log p(y_i). \quad (2)$$

Entropy regularization improves the diversity of samples, and as a result, Oracle's score increases by 0.67 BLEU points. While BaseNMT only benefits less than 0.1 BLEU points from the regularization, Conditional-EBR improves by 0.3 BLEU points (see Table 4). For this study we explored β from $\{0.01, 0.1\}$, and reported results use $\beta = 0.01$ selected based on the validation set. BaseNMT trained with $\beta = 0.1$ has the Oracle score of 65.76 on the test set (comparing to the Oracle score of 68.21 for $\beta = 0.01$), which indicates that stronger regularization reduces the sample quality.

4.9 Using XY-BERT for Joint-EBM

To explore the effect of a different way of conditioning on the source language, we compare the EBM constructed using the Joint-BERT model with EBM constructed using recently introduced XY-BERT (Guo et al., 2020). To construct EBM from XY-BERT, we remove the output layer and project each hidden-state of the final layer to a scalar energy value similar to how we build EBM from BERT. We compare these two models on IWSLT'14 De→En task. For XY-BERT we use German BERT for the encoder and English BERT for the decoder, following Guo et al. (2020). Our Joint-BERT uses Multilingual BERT because we feed both source and target sentences to BERT jointly. Conditional-EBR with XY-BERT achieves 38.33 BLEU score, which is 0.75 BLEU points higher than Conditional-EBR with Joint-BERT and improves the performance of XY-BERT with mask-predict decoding (Ghazvininejad et al., 2019) by 1.84 BLEU points.[4] We believe that the improvement in Conditional-EBR using XY-BERT is mostly attributable to using specialized BERT models. Moreover, XY-BERT has extra trainable modules, so we could fine-tune XY-BERT on the translation task for 60 epochs, while keeping the rest of the parameters fixed without causing catastrophic forgetting. Joint-BERT, on the other hand, does not have any extra parameters, so we fine-tuned all parameters for only 15 epochs. Further training of Joint BERT resulted in poor performance. We leave adding extra modules for better fine-tuning of Joint BERT for future studies.

[4] Guo et al. (2020) report 36.49 BLEU score using XY-BERT with 10 iterations of mask-predict decoding.

4.10 Maximizing Expected Score

As another comparison, we train our models by directly maximizing the expected BLEU score (compared to rank-based training):

$$\max_{\theta} \mathbb{E}_{\mathbf{y}_p\sim p_\theta(.|\mathbf{x})}[\text{BLEU}(\mathbf{y}_p,\mathbf{y}^*)] \quad (3)$$

We use log-trick to calculate the gradient of the above objective:

$$\begin{aligned}&\nabla_\theta \mathbb{E}_{p_\theta}[\text{BLEU}(\mathbf{y}_p,\mathbf{y}^*)] \\ &\quad= \mathbb{E}_{\mathbf{y}_p\sim p_\theta}[\text{BLEU}(\mathbf{y}_p,\mathbf{y}^*)[-\nabla_\theta E_\theta(\mathbf{y}_p,\mathbf{x}) \\ &\quad+ \mathbb{E}_{\mathbf{y}'\sim p_\theta}[\nabla_\theta E(\mathbf{y}',\mathbf{x})]]]. \quad (4)\end{aligned}$$

We use self-normalized importance sampling to draw samples from the energy-based model. We use one sample to approximate the outer expectation and 10 samples to approximate the inner expectation. We train both Marginal-EBM and Joint-EBM by maximizing the expected BLEU score on IWSLT'14 DE-EN. The former obtains a score of 34.20 BLEU and the latter achieves 34.77 BLEU points. Both models underperform rank-based training.

4.11 Inference Time

We compare the inference latency of EBR variations with BaseNMT (Table 5). We use 100 samples for re-ranking using Marginal-EBR, Conditional-EBR with Joint-BERT and Conditional EBR with XY-BERT (Guo et al., 2020). Inference on Marginal-EBR takes on average about 170 milliseconds per sentence more than inference in BaseNMT as we have to sample 100 sentences from BaseNMT and evaluate them on the energy model. We evaluate the Marginal-EBR only on the target sentences, while we evaluate Conditional-EBR for sequences from both source and target language, so the input sequence of Conditional-EBR is longer, thus having higher latency comparing to Marginal-EBR. We also measure the latency of Conditional-EBR when we use XY-BERT architecture to construct Joint-EBM. In this case, we have

Table 5: Average inference time per sentence (milliseconds), baseline transformer uses beam width of 5 and EBR uses 100 samples per sentence.

Method	De→En	En→De
Base-NMT	572	577
Marginal-EBR	749	756
Conditional-EBR (Joint BERT)	836	838
Conditional-EBR (XY-BERT)	921	929

two separate BERT models for source and target languages, increasing the number of parameters by 3.3 million and latency by about 90 milliseconds per sentence compared to Conditional-EBR that uses the Joint-BERT model.

5 Analysis

In this section, we study the sentence preference of Marginal-EBR created by the energy ranking.

5.1 Qualitative Analysis

We qualitatively investigate how the output of Marginal-EBR differs from that of BaseNMT model. On the IWSLT'14 test set, we examined 200 examples on which Marginal-EBR did better than NMT and 200 examples where BaseNMT is better. We find that about 30% of the time, the Marginal-EBR model chooses a translation with changed pronoun. Another frequent 'preference' Marginal-EBR makes compared to BaseNMT is to use the contraction form. Since this IWSLT data set is from TED talk, we conjecture that the energy model favors the translations that are in more oral style. Besides, it is also common for the Marginal-EBR model to prefer rephrases, for example, instead of using 'will' as used in BaseNMT, Marginal-EBR chooses the form 'am going to'. Finally, we find, for some pairs, Marginal-EBR chooses a different tense compared to the BaseNMT model (from MAP decoding).

Table 6 presents quintessential examples we find after examining 400 examples on IWSLT'14 De→En test set. It is worth to mention that examples do not strictly land in only one category. For example, the sentences we show in the 'Rephrase' type will also be counted as the change of pronouns. With this in mind, we compute statistics over the 400 sentences and find each of the 'Pronoun', 'Contraction' and 'Rephrase' appears approximately 30% of the time while 10% of the sentences change 'Tense'. The other less frequent types are changing of determiners, prepositions and deletion (comparing the MAP decoding of BaseNMT and preferred output by Marginal-EBR).

Type	Example
Pronoun	N: to us , he meant the freedom . E: for us , it meant freedom .
Contraction	N: they are exotic ; they are experimental . E: they are exotical . they 're experimental .
Rephrase	N: and it 's our unseen reality . E: that 's our invisible reality .
Tense	N: a new life has been born . E: and a new life was born .

Table 6: Typical examples on IWSLT'14 test set, categorized by the difference between BaseNMT and Marginal-EBR. 'N' stands for BaseNMT and 'E' stands for Marginal-EBR introduced in this paper.

Table 7: BLEU scores by length on IWSLT'14 test set. Sentences are divided into 3 groups according to reference length: less than or equal to 5 , in the range between 5 and 10, greater than 10.

	(0, 5]	(5, 10]	(10,)
NMT	23.78	33.22	34.77
Marginal-EBR	26.38	35.20	35.68

5.2 BLEU Gains by Length

Besides the qualitative analysis, we are also curious to see whether the improvement is affected by length. Table 7 shows the BLEU scores on the IWSLT'14 test set, which is divided into three bins according to the target length. Shorter sentences have the largest increase in BLEU, and the gain is decreasing as length increases. We reckon that it is easier for EBR to cover larger training space for sentences of shorter length and thus has the largest improvement in BLEU for these sentences.

5.3 Random Sentences

In the absence of access to the source sentence, the energy model ranks the outputs purely according to the features of target sentences. We hypothesize that the energy model is better at differentiating incoherent and coherent sentences and manage to show that through the following analysis. We apply two kinds of shuffle on IWSLT'14 test set targets: (1) global shuffle: tokens in the sentence are randomly shuffled (2) local shuffle: we first randomly select a token and randomly shuffle the tokens within a local window of three. Then we compute the energy scores of these shuffled sentences as well as the untouched ones. The energy scores are listed in Table 8. (The energy model assign a lower energy to its preference.) We find 87%

Table 8: Energy scores of randomly shuffled sentences as well as original targets on IWSLT'14 De→En test set.

Shuffle Type	Average Energy Scores
Local	-0.013
Global	0.002
Original	-0.037

of the time, the energy model is able to distinguish the original sentence from a local shuffled one, and 90.5% from the global shuffled one. This supports our hypothesis that the energy model is capable of capturing the fluency of generated candidates.

6 Conclusion and Future Work

We introduce energy-based re-ranking (EBR) to improve the performance of autoregressive neural machine translation. Despite its superior performance, EBR suffers from high latency because of its dependency on sampling from an autoregressive model. Directly sampling from the underlying EBM can speed up the inference, which is our future direction in order to benefit from the power of energy-based models for machine translation.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *Proc. of ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2020. On the weaknesses of reinforcement learning for neural machine translation. In *Proc. of ICLR*.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *Proc. of ICLR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pages 4171–4186.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proc. of EMNLP*, pages 1500–1505.

Bryan Eikema and Wilker Aziz. 2020. Is map decoding all you need? the inadequacy of the mode in neural machine translation. *arXiv:2005.10283*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. of ICML*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. 2019. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems 32*, pages 11058–11070.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems 33*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proc. of AISTATS*, pages 297–304.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. In *arXiv preprint arXiv:1902.01382*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. of ACL*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Tomoya Mizumoto and Yuji Matsumoto. 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In *Proc. of NAACL-HLT*, pages 1133–1138.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL-HLT 2019: Demonstrations*.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. 2019. Global autoregressive models for data-efficient sequence learning. In *Proc. of CoNLL*, pages 900–909.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proc. of ICLR*.

Khashayar Rohanimanesh, Kedar Bellare, Aron Culotta, Andrew McCallum, and Michael L Wick. 2011. Samplerank: Training factor graphs with atomic gradients. In *Proc. of ICML*, pages 777–784.

Amirmohammad Rooshenas, Aishwarya Kamath, and Andrew McCallum. 2018. Training structured prediction energy networks with indirect supervision. In *Proc. of NAACL-HLT*.

Amirmohammad Rooshenas, Dongxu Zhang, Gopal Sharma, and Andrew McCallum. 2019. Search-guided, lightly-supervised training of structured prediction energy networks. In *Advances in Neural Information Processing Systems 32*, pages 13522–13532.

Ronald Rosenfeld, Stanley F Chen, and Xiaojin Zhu. 2001. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Computer Speech & Language*, 15(1):55–73.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proc. of ACL*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pages 1715–1725.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proc. of NAACL-HLT*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proc. of ACL*, pages 1683–1692.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. A study of reinforcement learning for neural machine translation. In *Proc. of EMNLP*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating BERT into neural machine translation. In *Proc. of ICLR*.