A APPENDIX

A.1 DERIVATION FOR THE GRADIENT OF ψ

Notations. We present the derivation of the gradient of our backward model $P(\mathbf{x}|y;\psi)$ that we stated in 8. Throughout the derivation, we use the standard Jacobian notations. Specifically, if $f(x_1, x_2, ..., x_m) : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a smooth function, then $\frac{\partial f}{\partial x} \in \mathbb{R}^{n\times m}$ is the Jacobian matrix, where the entry at row i^{th} and column j^{th} is $\frac{\partial f_i}{\partial x_j}$. In the special case that $n = 1$, $\frac{\partial f}{\partial x}$ is the *transpose* of the gradient vector $\nabla_x f$. Additionally, per standard conventions, vectors are column vectors unless otherwise specified. To avoid confusions, we annotate the dimensions of the matrices and vectors in our equations.

Derivation. At training step t^{th}, the forward model's parameter from the previous step was $\theta^{(t-1)}$, and the backward model's parameter was $\psi^{(t-1)}$. Based on $\psi^{(t-1)}$, and receiving a sentence $y \sim P_T(\mathbf{y})$ in the target language T, the backward model samples a pseudo source sentence $\widehat{x} \sim P(\mathbf{x}|y;\psi^{(t-1)})$. Using $(\widehat{x}, y)$, the forward model computes the gradient and updates its parameter θ. For simplicity, we consider the case where the forward model is trained with SGD with learning rate η. This leads to the following update:

$$\theta^{(t)} = \theta^{(t-1)} - \eta\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big) \tag{10}$$

In MetaBT, we update $\psi^{(t-1)}$ into $\psi^{(t)}$ such that the loss of the forward model on the development *at the expected parameter* $\theta^{(t)}$ is minimized. We compute the gradient ∇_ψ according to this goal. The expected parameter $\theta^{(t)}$ is:

$$\begin{aligned}\overline{\theta}^{(t)} &= \mathbb{E}_{\widehat{x}\sim P(\mathbf{x}|y;\psi^{(t-1)})}\Big[\theta^{(t-1)} - \eta\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)\Big] \\ &= \theta^{(t-1)} - \eta\sum_{\widehat{x}} P\Big(\widehat{x}\Big|y;\psi^{(t-1)}\Big)\,\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)\end{aligned} \tag{11}$$

Here, the summation is taken over *all* possible sequences of tokens x. Note that under regulatory conditions of the distribution $P\big(\mathbf{x}\big|y;\psi^{(t-1)}\big)$, this summation converges.

Now, for simplicity, let us denote the loss of the forward model at $\overline{\theta}^{(t)}$ on the development set D_{dev} as $J_{\text{dev}}\Big(\overline{\theta}^{(t)}\Big)$. We apply the chain rule to compute $\nabla_\psi J_{\text{dev}}$ as follows:

$$[\nabla_\psi J_{\text{dev}}]^\top = \underbrace{\frac{\partial J_{\text{dev}}}{\partial \psi}}_{1\times|\psi|} = \underbrace{\frac{\partial J_{\text{dev}}}{\partial \overline{\theta}^{(t)}}}_{1\times|\theta|} \cdot \underbrace{\frac{\partial \overline{\theta}^{(t)}}{\partial \psi}}_{|\theta|\times|\psi|} \tag{12}$$

We will approximate the first factor in 12 using *a single sample* $\theta^{(t)}$, which is calculated according to the $\widehat{x}$ that we sample as discussed in 10, that is:

$$\frac{\partial J_{\text{dev}}}{\partial \overline{\theta}^{(t)}} \approx \frac{\partial J_{\text{dev}}}{\partial \theta^{(t)}} \tag{13}$$

Now we expand the second factor in 12 as follows:

$$\begin{aligned}\underbrace{\frac{\partial J_{\text{dev}}}{\partial \overline{\theta}^{(t)}}}_{|\theta|\times|\psi|} &= \underbrace{\frac{\partial \theta^{(t-1)}}{\partial \psi}}_{\approx 0 \text{ (Markov)}} - \eta\sum_x \underbrace{\nabla_\theta \ell\Big(x, y; \theta^{(t-1)}\Big)}_{|\theta|\times 1} \cdot \underbrace{\frac{\partial P\big(x\big|y;\psi^{(t-1)}\big)}{\partial \psi}}_{1\times|\psi|} && \text{(Markov assumption)} \\ &= -\eta\sum_x \underbrace{\nabla_\theta \ell\Big(x, y; \theta^{(t-1)}\Big)}_{|\theta|\times 1} \cdot \underbrace{\frac{\partial \log P\big(x\big|y;\psi^{(t-1)}\big)}{\partial \psi}}_{1\times|\psi|} \cdot \underbrace{P\Big(x\Big|y;\psi^{(t-1)}\Big)}_{\text{scalar}} && \text{(log-gradient trick)} \\ &= -\eta\mathbb{E}_{x\sim P\left(\mathbf{x}\middle|y;\psi^{(t-1)}\right)}\left[\nabla_\theta \ell\Big(x, y; \theta^{(t-1)}\Big) \cdot \frac{\partial \log P\big(x\big|y;\psi^{(t-1)}\big)}{\partial \psi}\right]\end{aligned} \tag{14}$$

Once again, we approximate this resulting expectation via *a single sample* $\widehat{x} \sim P\big(\mathbf{x}\big|y;\psi^{(t-1)}\big)$, that is:

$$\underbrace{\frac{\partial J_{\text{dev}}}{\partial \overline{\theta}^{(t)}}}_{|\theta|\times|\psi|} \approx -\eta \underbrace{\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)}_{|\theta|\times 1} \cdot \underbrace{\frac{\partial \log P\big(\widehat{x}\big|y;\psi^{(t-1)}\big)}{\partial \psi}}_{1\times|\psi|} \tag{15}$$

Putting 13, 15, and 12 together, we have the final approximating gradient $\nabla_\psi J_{\text{dev}}$:

$$\big[\nabla_\psi J_{\text{dev}}\big]^\top \approx -\eta \cdot \underbrace{\frac{\partial J_{\text{dev}}}{\partial \theta^{(t)}}}_{1\times|\theta|} \cdot \underbrace{\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)}_{|\theta|\times 1} \cdot \underbrace{\frac{\partial \log P\big(\widehat{x}\big|y;\psi^{(t-1)}\big)}{\partial \psi}}_{1\times|\psi|} \tag{16}$$

Using associativity of matrix multiplications, we can group the first two factors which result in a scalar. Then, by transposing both sides, we obtain the final result:

$$\underbrace{\nabla_\psi J_{\text{dev}}}_{|\psi|\times 1} \approx -\eta \cdot \Big[\underbrace{\nabla_\theta J_{\text{dev}}\Big(\theta^{(t)}\Big)^\top}_{1\times|\theta|} \cdot \underbrace{\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)}_{|\theta|\times 1}\Big] \cdot \underbrace{\nabla_\psi \log P\Big(\widehat{x}\Big|y;\psi^{(t-1)}\Big)}_{|\psi|\times 1} \tag{17}$$

This final result is *almost* what we stated in 8. In 8, we do not have the learning rate term $-\eta$, since η is a scalar and can be absorbed into the learning rate of the backward model. Thus, our derivation is complete.

It is worth noting that our derivation above assumes that the forward model parameters θ is updated with vanilla stochastic gradient descent. In reality, we either use Adam (Kingma & Ba, 2015) or LAMBOptimizer to update θ. In that case, the derivation of MetaBT stays almost the same, except that at Eq. 11, we will have a slightly different update:

$$\overline{\theta}^{(t)} = \mathbb{E}_{\widehat{x}\sim P(\mathbf{x}|y;\psi^{(t-1)})}\Big[\theta^{(t-1)} - \eta \cdot h\Big(\nabla_\theta \ell\Big(\widehat{x}, y; \theta^{(t-1)}\Big)\Big)\Big], \tag{18}$$

where h is the function specified by the optimizer. If we assume that all moving averages and momentums of the optimizer are independent of θ and ψ, then we can simply replace $\nabla_\theta \ell\big(\widehat{x}, y; \theta^{(t-1)}\big)$ with $h\big(\nabla_\theta \ell\big(\widehat{x}, y; \theta^{(t-1)}\big)\big)$ and use follow the same derivation.

It is also worth noting that in our derivations, we made two strong approximations about computing an expectation via a single sample, namely at 13 and 15, which could potentially lead to a high variance in our approximation. However, since the backward model $P(\mathbf{x}|y;\psi)$ is pre-trained to convergence, most of the samples $\widehat{x}$ from it will concentrate around the correct pseudo source sentence, and hence the variance of these approximations are reasonable. It is hard to measure such variance and confirm our hypothesis here. Nevertheless, the fact that our training procedure does not diverge empirically suggests that our approximations have acceptable variances.

A.2 Training Details

Here we list some other training details of the standard setting:

- We use the Transformer-Base architecture from Vaswani et al. (2017). All initialization follow the paper.
- We share all embeddings and softmax weights between the encoder and the decoder.
- We use a batch size of 2048 sentences for the forward model, and a batch size of 1024 sentences for the backward model. We use a smaller batch size of 512 for the validation batches that are sampled from D_{dev}.
- We train for 200,000 update steps, where each update step counts as one update for the forward model and one update for the backward model, as we described in Section 2.

The training details of the multilingual NMT setting are as follows:

- We use the transformer model with word embedding of dimension 512, and feed-forward dimension of 1024. It has 6 layers and 4 attention heads for both the encoder and the decoder.

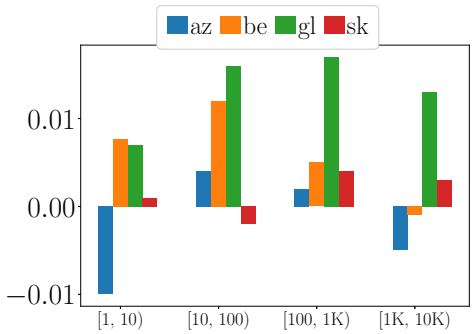

Figure 6: Gain in target word F1 measures of MBT compared to MLE. Words are bucketed from left to right based on increasing frequency in the S'-T data. MBT brings more gains on target words have middle frequency in the related language data.

- We share all embeddings between the encoder and the decoder.
- Since the dataset is relatively small, we ran each experiment 4 times with different random seeds and record the average.
- To optimize the backward model, we use a baseline to stabilize training. We keep a moving average baseline of the gradient dot-product as in 8 (the forward model's gradient alignment), and subtract the baseline from the current reward before each update.

A.3 Effect of MBT on Multilingual Transfer

To further demonstrate the effect of these improvements in vocabulary coverage, we compare the word prediction accuracy for target words in the training data of S'-T. We bucket the target words in the test set according to their frequency in S'-T , and then calculate the word F-1 scores for each bucket[2]. The difference of F-1 score between MBT and MLE for the four languages in the multilingual setting are plotted in 6. We can see that MBT generally has higher word accuracy than MLE, and the gains are most significant for middle-frequency words in the related language, probably because high-frequency words may be covered well already by the training data in the low-resource language. The improved word accuracy indicates that MBT can make better use of the data from the related language.

A.4 Example Translations

Src	As the town ' s contribution to the 150th anniversary of the Protestant Church in Haigerloch , the town ' s Office of Culture and Tourism is to dedicate the last of this year ' s public thematic tours on Sunday 27 October to the Abendsmahlskirche (Church of the Holy Communion) .
Ref	Als Beitrag der Stadt zum 150 - jährigen Bestehend der Evangelischen Kirche in Haigerloch widmet das Kultur - und Tourismusbüro der Stadt die letzte ihrer diesjährigen öffentlichen Themenführungen am Sonntag , 27 . Oktober , der Abendmahlskirche .
MLE	Als Beitrag der Stadt zum 150.
MBT	Als Beitrag der Stadt zum 150 - jährigen Bestehen der evangelischen Kirche in Haigerloch widmet das Amt für Kultur und Tourismus am Sonntag , 27 . Oktober , der Abendsmahlskirche die letzten öffentlichen Themenführungen .

Table 2: Examples of en-de translations.

[2]We use compare-mt for analysis (Neubig et al., 2019)