Normalizing Mutual Information for Robust Adaptive Training for Translation

Youngwon Lee[1]*, **Changmin Lee**[2], **Hojin Lee**[2], **Seung-won Hwang**[1]†
[1] Seoul National University, {ludaya,seungwonh}@snu.ac.kr
[2] Kakao Enterprise, South Korea, {louis.cm,lambda.lee}@kakaoenterprise.com

Abstract

Despite the success of neural machine translation models, tensions between fluency of optimizing target language modeling and source-faithfulness remain as challenges. Previously, Conditional Bilingual Mutual Information (CBMI), a scoring metric for the importance of target sentences and tokens, was proposed to encourage fluent and faithful translations. The score is obtained by combining the probability from the translation model and the target language model, which is then used to assign different weights to losses from sentences and tokens. Meanwhile, we argue this metric is not properly normalized, for which we propose Normalized Pointwise Mutual Information (NPMI). NPMI utilizes an additional language model on source language to approximate the joint likelihood of source-target pair and the likelihood of the source, which is then used for normalizing the score. We showed that NPMI better captures the dependence between source-target and that NPMI-based token-level adaptive training brings improvements over baselines with empirical results from En-De, De-En, and En-Ro translation tasks.

1 Introduction

Neural machine translation (NMT) models have achieved remarkable performance since Vaswani et al. (2017) introduced encoder-decoder architecture with self- and cross-attention mechanisms. However, they were also reported to generate hallucinations (Lee et al., 2018; Raunak et al., 2021) in some cases, failing to balance the dual goals of improving fluency while preserving faithfulness to the source.

Conditional Bilingual Mutual Information (CBMI), a metric for target tokens and sentences computed as the log quotient of the translation and the target-side language model probability, was proposed by Zhang et al. (2022) to guide the translation model with additional signal of importance of each target token or sentence, in combination with token-level adaptive training (Gu et al., 2020) in order to achieve this goal.

(a) Faithful translation

src Ich weiss nicht, ob das passieren wird oder nicht, aber ich bin optimist.
tgt I don't know whether that's going to happen or not, but I'm an optimist.

(b) Noisy translation

src Mich zum beispiel. (= "Me for example")
tgt It did mine.

	log P(src, tgt)	log P(src)	log P(tgt)	CBMI	NPMI
(a)	-39.6	-34.0	-36.0	1.38	**0.77**
(b)	-35.5	-21.0	-19.6	1.30	**0.15**

Figure 1: An example from IWSLT14 De-En train set. While our proposed sentence-level NPMI assigns a large score (near the upper bound 1) to the faithful source-target pair and a small score (near zero, which indicates neutrality) to a rather noisy pair, sentence-level CBMI scores for the two pairs are unable to achieve that. Note that the joint log likelihood values from the two pairs are comparable, while target lengths differ a lot.

While CBMI score is devised to pursue this joint goal of translation, we argue that it does not take source context into account which leads to its failure to provide a reliable measure of relevance for some cases. For example, Figure 1 illustrates faithful and noisy translations, where the former is expected to have a higher score. However, sentence-level CBMI score, which is defined as aggregated token-level CBMI scores simply divided by the length of the target sentence, fails to capture that the first pair is much more strongly connected in terms of the content than the second one.

We argue that this is because CBMI has a tendency to assign higher values for noisy or unlikely examples and vice versa due to the nature of pointwise mutual information (PMI) with an unbounded range.

Inspired by normalized pointwise mutual infor-

*Work done during internship at Kakao Enterprise.
†Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8008–8015
December 7-11, 2022

mation (NPMI), we propose to normalize by joint log likelihood (denoted as $\log P(\text{src}, \text{tgt})$ in Figure 1). Our derivation of using NPMI for sentence- and token-level weighting leads to a combination of log quotients of probabilities from translation model and source and target language models, unlike existing methods not considering the source-side language model. Figure 1 shows that our enhanced scoring, with a more founded derivation of paired normalization, can distinguish between faithful and noisy translations more clearly.

Our method is validated with WMT and IWSLT benchmarks with diverse language pairs and shows consistent improvements in COMET (Rei et al., 2020), a pretrained neural network based evaluation framework that reportedly correlates highly with human evaluation (Kocmi et al., 2021), as well as the widely-used BLEU scores.

2 Method

We first overview the concept of PMI and CBMI (Zhang et al., 2022) along with its proposed normalization. We then motivate why a new, and more systematic normalization is needed, and derive our proposed method.

2.1 Preliminary: PMI and adaptive training

Given two random variables X and Y, the pointwise mutual information (PMI) between the observations x and y is

$$\text{PMI}(x; y) = \log \left(\frac{p(x, y)}{p(x)p(y)} \right), \tag{1}$$

which is not bounded below and has an upper bound of $-\log p(x, y)$.

Token-level adaptive training, inspired by earlier approaches to fighting class imbalance problem in classification tasks, aims to assign static or dynamic weights to each of the tokens to further guide the translation model (Gu et al., 2020).

$$\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta) = \sum_j w_j \log p(y_j \mid \mathbf{x}, \mathbf{y}_{<j}; \theta) \tag{2}$$

2.2 Baseline: CBMI

Token-level CBMI, which is used to determine weights of loss from each target token, is the PMI between the target token considered and the whole source sentence $\mathbf{x}$, conditioned on the partially constructed target prefix $\mathbf{y}_{<j}$.

$$\begin{aligned} \text{CBMI}^t(\mathbf{x}; y_j) &:= \text{PMI}(\mathbf{x}; y_j \mid \mathbf{y}_{<j}) \\ &= \log \frac{p(\mathbf{x}, y_j \mid \mathbf{y}_{<j})}{p(\mathbf{x} \mid \mathbf{y}_{<j}) p(y_j \mid \mathbf{y}_{<j})} \\ &= \log \frac{p(y_j \mid \mathbf{x}, \mathbf{y}_{<j})}{p(y_j \mid \mathbf{y}_{<j})} = \log \frac{p_{\text{TM}}(j)}{p_{\text{tLM}}(j)} \end{aligned} \tag{3}$$

where $p_{\text{TM}}(j) = p(y_j \mid \mathbf{x}, \mathbf{y}_{<j}; \theta_{\text{TM}})$ is the translation model's output probability on the j-th target token and similarly $p_{\text{tLM}}(j) = p(y_j \mid \mathbf{y}_{<j}; \theta_{\text{tLM}})$ is the target-side language model's prediction on the same token.

For sentence-level scoring, token-level scoring is aggregated then normalized by the target sentence length $|\mathbf{y}|$ as follows:

$$\text{CBMI}^s(\mathbf{x}; \mathbf{y}) := \frac{1}{|\mathbf{y}|} \text{PMI}(\mathbf{x}; \mathbf{y}) \tag{4}$$

$$= \frac{1}{|\mathbf{y}|} \sum_j \text{CBMI}(\mathbf{x}; y_j). \tag{5}$$

Note that unlike token-level CBMI defined simply as the PMI between the source sentence and the target token considered by equation (3), sentence-level CBMI is defined as the PMI between the source and the target sentence *divided by the target sentence length* in equation (4). The derivation of (5) can be found in the appendix, or the reader might refer to the original paper (Zhang et al., 2022).

Then, to translate CBMI scores into token and sentence weights for adaptive training, CBMI^t and CBMI^s are further normalized by inter-token and inter-sentence statistics collected from a mini-batch, such as the mean μ and standard deviation σ of token and sentence scores in the batch, scaled by hyperparameter λ:

$$w_j^t = \max\left(0, 1 + \lambda^t \cdot \left(\text{CBMI}(\mathbf{x}; y_j) - \mu^t\right) / \sigma^t\right)$$

sentence-level weight w^s is similarly determined from μ^s, σ^s and λ^s, then aggregated into the final token weights w_j as follows:

$$w_j = w_j^t \cdot w^s$$

2.3 Motivation: normalization

Our critiques for their proposed normalization are as follows:

- Source-agnostic: When the pair of the source and the target has relatively low (or high) likelihood, both token- and sentence-level CBMI

scores can be over- (or under-) estimated as Figure 1 shows. While target length is (negatively) correlated with target sentence likelihood, it may diverge from the joint likelihood in some cases, resulting in faithful translations with relatively longer targets penalized and noisy translations with relatively shorter targets not penalized.

- Mapping: λ and σ together determine how much the final weights of tokens with different CBMI scores will diverge, while the former is an empirically determined constant and the latter may vary across batches, or over time as training proceeds and the model output changes.

3 Proposed: Source-aware Normalization

We first propose an alternative normalization, inspired by NPMI, then discuss how this score guides adaptive training.

3.1 Balanced normalization

Our first contribution is to propose a better founded normalization used in Bouma (2009), which normalizes PMI with the absolute value of the logarithm of the joint probability.

$$-1 < \text{NPMI}(x;y) = \frac{\text{PMI}(x;y)}{-\log p(x,y)} \leq 1.$$

This normalization bounds scoring within the range $(-1, 1]$ and the sign of scoring can also be interpreted: 0 for independence, 1 for complete co-occurrence, and -1 for no co-occurrence. However, this requires the estimation of $p(x, y)$, which we derive to obtain from the source-side language model (sLM) as below:

$$\begin{aligned}
\text{NPMI}^s(\mathbf{x};\mathbf{y}) &= \frac{\log\left(p(\mathbf{x},\mathbf{y})/p(\mathbf{x})p(\mathbf{y})\right)}{-\log p(\mathbf{x},\mathbf{y})} \\
&= \frac{\log\left(p(\mathbf{y}\,|\,\mathbf{x})p(\mathbf{x})/(p(\mathbf{x})p(\mathbf{y}))\right)}{-\log\left(p(\mathbf{y}\,|\,\mathbf{x})p(\mathbf{x})\right)} \\
&= \frac{\log\left(p(\mathbf{y}\,|\,\mathbf{x})/p(\mathbf{y})\right)}{-\log\left(p(\mathbf{y}\,|\,\mathbf{x})p(\mathbf{x})\right)} \\
&= \frac{\log p(\mathbf{y}\,|\,\mathbf{x}) - \log p(\mathbf{y})}{-\left(\log p(\mathbf{y}\,|\,\mathbf{x}) + \log p(\mathbf{x})\right)} \\
&= \frac{\sum_j \log p_{\text{tLM}}(j) - \sum_j \log p_{\text{TM}}(j)}{\sum_i \log p_{\text{sLM}}(i) + \sum_j \log p_{\text{TM}}(j)}
\end{aligned}$$

In the same way, we can derive token-level NPMI:

$$\begin{aligned}
&\text{NPMI}^t(\mathbf{x};y_j) = \text{NPMI}(\mathbf{x};y_j\,|\,\mathbf{y}_{<j}) \\
&= \frac{\text{PMI}(\mathbf{x};y_j\,|\,\mathbf{y}_{<j})}{-\log p(\mathbf{x},y_j\,|\,\mathbf{y}_{<j})} \\
&= \frac{\log p(y_j\,|\,\mathbf{x},\mathbf{y}_{<j}) - \log p(y_j\,|\,\mathbf{y}_{<j})}{-\log p(\mathbf{x},y_j\,|\,\mathbf{y}_{<j})} \\
&= \frac{-(\log p_{\text{TM}}(j) - \log p_{\text{tLM}}(j))}{\log p(y_j\,|\,\mathbf{x},\mathbf{y}_{<j}) + \log p(\mathbf{y}_{<j}\,|\,\mathbf{x}) + \log \frac{p(\mathbf{x})}{p(\mathbf{y}_{<j})}} \\
&= \frac{-q(j)}{\log p_{\text{TM}}(j) + \sum\limits_i \log p_{\text{sLM}}(i) + \sum\limits_{k<j} q(k)}
\end{aligned}$$

where $q(j) := \log p_{\text{TM}}(j) - \log p_{\text{tLM}}(j)$.

While this derivation is more complex than that of CBMI, it can still be computed efficiently in one forward pass.

3.2 Adaptive training

With NPMI normalization bounding its range to ± 1, we no longer require λ or σ for rescaling, but simply multiplying source- and token-level relative scores:

$$w_j^t = \frac{\text{NPMI}^t(\mathbf{x};y_j)^+}{\mu^t}$$

$$w^s = \frac{\text{NPMI}^s(\mathbf{x};y_j)^+}{\mu^s}$$

where $x^+ := \max(x, 0)$, to honor the design of "positive" NPMI values, by selectively weighing pairs with cooccurrences, and μ is the average of positive NPMI values that helps center the weights at 1.

The weight w_j, relying on translation and language models themselves, is less reliable in earlier stages of training, when it can be better off resorting to unweighted loss. This estimation gradually gets better in later stages.

We thus adopt dynamic smoothing, between weighting all tokens as 1, and by w_j, where c increase over time during training.

$$w_j' = (1-c) + c \cdot w_j^t \cdot w_s$$

Compared to CBMI, which solved the same problem through training the translation and the language model for some steps with the unweighted negative log-likelihood loss then applying the weighting of tokens to the translation model afterward, we increased the value of c over training steps so that it exponentially approaches a targeted

value. The former approach of skipping the weighting in the earlier stage of training can be viewed as setting $c = 0$ for some steps then fixing $c = 1$ for the rest. In contrast, by gradually increasing the ratio c, the model is allowed to be guided by the faithfulness measure relatively earlier in training, preventing it from being fully affected by detrimental training examples. We also note that mixing the unweighted and weighted loss with time-varying ratio c essentially has an effect of dynamically manipulating the scale hyperparameter λ in CBMI.

We only use language models for assisting the translation model during the training, that is, at inference time only the translation model is used for decoding.

4 Experiments

4.1 Datasets

We conducted experiments on three translation datasets, namely (1) WMT14 English-German (En-De) dataset which consists of approximately 4.5M training examples, (2) WMT16 English-Romanian (En-Ro) dataset comprising of about 610k examples, and (3) IWSLT14 De-En dataset for spoken language, which comes with 160k training examples. Following previous work, we used joined vocabulary of size 32k constructed using byte-pair encoding (Sennrich et al., 2016) for WMT14 En-De. We used BPE joined vocabulary of size 40k for WMT16 En-Ro and 10k for IWSLT14 De-En.

4.2 Results

Table 1 shows the tokenized BLEU scores with compound split following previous work (Vaswani et al., 2017; Zhang et al., 2022) and COMET (Rei et al., 2020) scores on the three translation benchmarks. Our method shows improvements over baselines, with larger margins under relatively low resource settings. We provide the detailed configuration for all experiments in the appendix A.

5 Analysis

Here we present more detailed analyses of our method regarding how the different levels of weighting and dynamic weight smoothing over time affected the performance. All the results are from experiments conducted on IWSLT14 De-En and evaluated on the test set.

5.1 Ablation study

Method	COMET
Transformer	40.58
+ sentence-level NPMI	40.63
+ token-level NPMI	41.14
+ both	41.40

Table 2: Effect of token- and sentence-level weights.

First, we inspect the effects of token- and sentence-level weighting separately. Table 2 shows that jointly using the two together leads to synergy, which gives the best result.

Method	COMET
CBMI	40.04
CBMI + weight smoothing	37.49
NPMI − weight smoothing	41.26
NPMI	41.40

Table 3: Effect of weight smoothing on token-level adaptive training.

Next, we examine the effect of dynamic weight smoothing on CBMI and NPMI. Table 3 shows the effect of dynamic weight smoothing applied to CBMI and removed from NPMI. CBMI performs worse with weight smoothing added; we conjecture that this is because the unstable (highly variant) nature of CBMI values is especially harmful to adaptive training in the very beginning. Though CBMI scores might fit with other strategies of scheduling the mix ratio c, we leave it as an open problem. Meanwhile, the score drop on NPMI shows that dynamic smoothing is effective at preventing the model from being affected by noisy examples and NPMI can provide more reliable, helpful weights even if language models are yet to be converged.

5.2 Training with single LM

Method	COMET
NPMI (two LMs)	41.40
NPMI (shared LM)	41.26

Table 4: Effect of single LM training for NPMI.

We present the option for training with a single, unified LM that models both the source and the target language if the vocabulary is constructed jointly as done in our experiments. Table 4 shows that our method maintains comparable performance

Dataset	WMT14 En→De		WMT16 En→Ro		IWSLT14 De→En	
Model	BLEU	COMET	BLEU	COMET	BLEU	COMET
Transformer	27.95	49.10	34.41	56.02	35.27	40.58
CBMI (Zhang et al., 2022)	**28.10**	49.29	34.56	56.67	35.16	40.04
NPMI (ours)	28.09	**49.30**	**34.62**	**57.32***	**35.28**	**41.40***

Table 1: BLEU and COMET scores on translation tasks. * refers to statistical significance.

in this setting. While the perplexity of the shared language model on both train and validation set was slightly higher than that of the separate language models, due to the well-equipped normalization scheme its detrimental effect on the performance was limited.

6 Related work

6.1 Leveraging target-side LMs

Under encoder-decoder seq2seq framework, the decoder is responsible for both capturing the embedded content of the source sequence and generating a fluent target sequence that faithfully reflects the captured information. As an example of work tried to relieve this burden through the use of LM on the target language, Stahlberg et al. (2018) combined the prediction of the NMT model and the language model through elementwise product to separate the role of the models so that NMT model could focus on modeling the source sequence faithfully while the language model accounts for the fluency. Target-side LMs may be used for measuring importance of target tokens and sentences, as described in the following paragraph.

6.2 Token-level adaptive training in NMT

Inspired by previous work in vision field, Gu et al. (2020) suggested a token-level adaptive objective, weighting loss of each token differently based on frequency, to fight data imbalance which is basically to help the model to learn embeddings of rare tokens. One can reappropriate the aforementioned approach of leveraging additional target-side LM in NMT to help the translation model distinguish which tokens require source context heavily and which tokens do not, then weight their losses based on that score as Miao et al. (2021) first suggested to consider the difference ('margin') between the output probabilities of translation model and the target-side language model. Zhang et al. (2022) made a similar approach, adjusting the importance of each target token or sentence according to the pointwise mutual information between the source sequence and target token, conditioned on the target prefix. Our distinction is to pose the necessity and also to present the advantage of introducing source-side LM to training of translation models from scratch.

7 Conclusion

In this paper, we propose a source-aware metric for target tokens and sentences based on normalized pointwise mutual information (NPMI) that effectively captures the dependence between the source and the target for translation task. With this score, the model can figure out how much specific tokens require the source context for proper translation and how faithful a given source-target pair is, thereby putting more focus on examples with higher adequacy or importance. We also devise a new token-level adaptive training strategy based on NPMI score, which dynamically adjusts the participation of weighted loss over time to gracefully overcome the limitation of imprecise approximation of model output probabilities in the earlier training stage.

Experimental results on translation benchmarks show that our proposed NPMI, combined with dynamic weight smoothing, performs well over various datasets and languages. We also validated through ablation experiments that our methods offer the best results when they are used together. We leave (1) the search for the best way of scheduling for weight smoothing and (2) leveraging powerful pretrained language models, rather than language models trained from scratch as future work.

Acknowledgements

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2020-0-01789), and by Kakao Enterprise.

Limitations

Assuring that the model pays more attention to the source sequence when necessary might not be enough for successful translation. Our model sometimes generated word-for-word translations, which are firmly rooted in the source sequence but not necessarily revealing the true meaning of the idiomatic phrase it contained. Hopefully, it might be alleviated given access to additional training data.

Also, due to the additional source-side LM, our method requires additional GPU memory, which could be a burden. In the case of using joined vocabulary, this can be relieved via using a unified language model for both the source and the target language, with minimal performance degradation as described in 5.2.

References

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. 2020. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1035–1046, Online. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 478–494. Association for Computational Linguistics.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. *Interpretability and Robustness in Audio, Speech, and Language (IRASL) Workshop at 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Mengqi Miao, Fandong Meng, Yijin Liu, Xiao-Hua Zhou, and Jie Zhou. 2021. Prevent the language model from being overconfident in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3456–3468, Online. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

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.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. *CoRR*, abs/1809.00125.

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*, volume 30. Curran Associates, Inc.

Songming Zhang, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, Jian Liu, and Jie Zhou. 2022. Conditional bilingual mutual information based adaptive training for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2377–2389. Association for Computational Linguistics.

A Experimental Settings in Detail

Here we provide the detailed experimental settings. We used the same configuration including vocabulary for the translation model and the language models.

We tokenize all the datasets with byte-pair encoding (BPE), with the dictionary being jointly constructed upon both the source and the target language. The resulting vocabulary size was 10k for IWSLT, 32k for WMT14 En-De and 35k for WMT16 En-Ro. Label smoothing with $\epsilon = 0.1$ was applied for all experiments. Also, we used Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ and the `inverse_sqrt` learning rate scheduler with default learning rate $7 \cdot 10^{-4}$, initial learning rate 10^{-7}, and warm-up steps 4000 for all tasks, with the sole exception of using the default learning rate of $5 \cdot 10^{-4}$ for IWSLT experiments.

For IWSLT, we used `transformer_iwslt_de_en` architecture, which has 6 layers in both the encoder and the decoder with embedding size 512, feed-forward network hidden dimension of 1024, 4 attention heads and applied dropout rate of 0.3. Lastly, we used batch size of 4k.

For the others, we used `transformer` (base) architecture, which has 6 layers, embedding size of 512, feed-forward network hidden dimension of 2048 and 8 attention heads. Dropout rate of 0.1 was used for WMT14 En-De while 0.3 was used for WMT16 En-Ro. Following the original settings from Zhang et al. (2022), we also applied attention dropout and activation dropout of rate 0.1 for training CBMI model on WMT14 En-De. We used effective batch size of 32k for these datasets.

We applied compound split to compute the (tokenized) BLUE scores for reporting performance on the test sets, and used detokenized BLEU scores for validation and choosing the best checkpoints. We used the average checkpoint over the 5 last checkpoints for WMT14 En-De and the 5 best checkpoints for others for evaluation. Following legacy settings, beam search was adopted as the decoding strategy with the beam size of 4 along with length penalty of 0.6 for WMT14 En-De, and beam size of 5 and length penalty of 1.0 for others.

For training CBMI models, we used the same scale hyperparameters $\lambda^s = 0.3$ and $\lambda^t = 0.1$ as suggested by Zhang et al. (2022), and ran experiments with different number of training steps for the pretraining stage with a fixed total number of training steps set as the same as the other models, then chose the best performing one.

For weight smoothing, we used the following formula to increase the value of c towards a fixed targeted value c_0 exponentially:

$$c(t) = (1 - r^{t/\tau}) \cdot c_0$$

where t is the training step. We set c_0 as 0.3 for WMT14 En-De and 0.6 for the others. Then, we searched for the value of r and τ which basically determines how fast we want the weighted loss to participate in training, especially in earlier stage. Since increasing r and lowering τ have the same effect and vice versa, we fixed $r = 0.99$ and searched for τ. The values chosen were $\tau = 4000$ for WMT14 En-De, $\tau = 400$ for WMT16 En-Ro, and $\tau = 800$ for IWSLT14 De-En.

B Complete Results

WMT14 En→De	BLEU	COMET
Transformer	$27.95 \pm .01$	$49.10 \pm .27$
CBMI	$28.10 \pm .07$	$49.29 \pm .07$
NPMI	$28.09 \pm .09$	$49.30 \pm .30$
WMT16 En→Ro	BLEU	COMET
Transformer	$34.41 \pm .10$	$56.02 \pm .14$
CBMI	$34.56 \pm .11$	$56.67 \pm .41$
NPMI	$34.62 \pm .06$	$57.32 \pm .16$
IWSLT14 De→En	BLEU	COMET
Transformer	$35.27 \pm .03$	$40.58 \pm .30$
CBMI	$35.16 \pm .05$	$40.04 \pm .18$
NPMI	$35.28 \pm .04$	$41.40 \pm .12$

Table 5: The complete results including standard error for each of the values presented previously in Table 1.

C Changes in NPMI Values During Training

As training proceeds, translation and language models produce better approximations for the probabilities of unknown true data distribution. We empirically observed that the average NPMI values determined by the model (for the training samples) increase over time. Similarly, for the examples in the validation set, mean NPMI values tend to increase then saturate or start to decrease over time, which we believe to be another signal indicating overfitting other than the rebound in validation loss. The peak mean sentence-level NPMI values on the validation set for IWSLT14 De-En was approximately 0.44. This behavior was consistent among different settings for scheduling the c value, and the peak value did not tend to fluctuate a lot. This can be viewed as models with slightly different configurations reach a sort of consensus on how faithful examples a given dataset provides are, which implies that although we are using relatively smaller models trained from scratch on a smaller dataset, the estimated probabilities are quite reliable and that our proposed NPMI has potential as a metric for evaluating source-target faithfulness to be used for purposes other than token-level adaptive training.

D Derivation of sentence-level CBMI

Here we repeat the proof from Zhang et al. (2022) that the pointwise mutual information between the source and the target sentences $\text{PMI}(\mathbf{x}, \mathbf{y})$ equals

the sum of the pointwise mutual information between the source sentence and each target token conditioned on the target prefix preceding that token $\mathrm{PMI}(\mathbf{x}, y_j \mid \mathbf{y}_{<j})$.

$$
\begin{aligned}
\mathrm{CBMI}^s(\mathbf{x};\mathbf{y}) &:= \frac{1}{|\mathbf{y}|}\mathrm{PMI}(\mathbf{x};\mathbf{y}) \\
&= \frac{1}{|\mathbf{y}|}\log\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})\cdot p(\mathbf{y})} \\
&= \frac{1}{|\mathbf{y}|}\log\frac{p(\mathbf{y}\mid\mathbf{x})\cdot \cancel{p(\mathbf{x})}}{\cancel{p(\mathbf{x})}\cdot p(\mathbf{y})} \\
&= \frac{1}{|\mathbf{y}|}\log\prod_j\frac{p(y_j\mid\mathbf{x},\mathbf{y}_{<j})}{p(y_j\mid\mathbf{y}_{<j})} \\
&= \frac{1}{|\mathbf{y}|}\sum_j\log\frac{p(y_j\mid\mathbf{x},\mathbf{y}_{<j})}{p(y_j\mid\mathbf{y}_{<j})} \\
&= \frac{1}{|\mathbf{y}|}\sum_j\mathrm{PMI}(\mathbf{x},y_j\mid\mathbf{y}_{<j}) \\
&= \frac{1}{|\mathbf{y}|}\sum_j\mathrm{CBMI}^t(\mathbf{x};y_j).
\end{aligned}
$$

Division by sentence length was adopted as an attempt to fight the variance of CBMI, since PMI is not bounded below and has a moving upper bound that increases as joint probability decreases. PMI is affected not only by the 'tendency to co-occur' but also by how likely the two observations are, which is the reason why CBMI values of different tokens and sentences may exhibit high variance, thereby hindering mapping them to weights. Although division by target length did work to some extent in mitigating this variance, as described in subsection 2.3, there are cases where this correlation is violated where putting the source into consideration together can solve the issue.

E Examples Generated from Trained Models

We present some examples from IWSLT14 En-De test set.

Source: er hatte die erfahrung gehabt.
Reference: he had had the experience.
NPMI: he had had the experience.
CBMI: he had experience.

Source: einige sind gekommen und gegangen.
Reference: some have come and gone.
NPMI: some have come and gone.
CBMI: some came and went.

Source: eine fast identische struktur.
Reference: an almost identical structure.
NPMI: an almost identical structure.
CBMI: it's an almost identical structure.

Source: ich komme nun zum ende.
Reference: so, i'm going to wrap up now.
NPMI: i'll end now.
CBMI: i'll come to the end now.

Source: er sagte: » was ist denn los mit dir? nun trink doch was. «
Reference: he said, "what's wrong with you? have some beer."
NPMI: he said, "what's going on with you? well, drink something."
CBMI: he said, "what about you? what's going on?"

Source: da frage ich, wie fachlich kompetent war diese diagnose?
Reference: then my question is, how professionally competent was this diagnosis?
NPMI: i'm asking, how professionally competent was this diagnosis?
CBMI: and i'm asking, how was this diagnostic?

Source: da sollte also besser eine neun am anfang meiner todeszahl stehen.
Reference: so there better be a nine at the beginning of my death number.
NPMI: so there should be a nine at the beginning of my death number.
CBMI: so there should be better a nine at the beginning of my death row.