Enhancing the Nonlinear Mutual Dependencies in Transformers with Mutual Information

Anonymous ACL submission

Abstract

The Predictive Uncertainty problem does exist in Transformers. We present that pre-trained Transformers can be further regularized by mutual information to alleviate such issue in Neural Machine Translation (NMT). In this paper, we explicitly capture the nonlinear mutual dependencies existing in decoder self-attentions to reduce the model uncertainty concerning token-token interactions. Specifically, we adopt an unsupervised objective of mutual information maximization on self-attentions with the contrastive learning methodology and construct the estimation of mutual information by using InfoNCE. Experimental results on WMT'14 En→De, WMT'14 En→Fr demonstrate the consistent effectiveness and evident improvements of our model over the strong baselines. Quantifying the model uncertainty again verifies our hypothesis. The proposed plug-and-play approach can be easily incorporated and deployed into pre-trained Transformer models. Code will be released soon[1].

1 Introduction

Predictive uncertainty ubiquitously exists in Deep Learning or Machine Learning based models (Ott et al., 2018a; Xiao and Wang, 2019; Wang et al., 2019; Abdar et al., 2020; Xiao and Wang, 2021). It consists of data uncertainty (aleatoric uncertainty) and model uncertainty (epistemic uncertainty). Researchers capture and quantify uncertainties to better interpret models and enhance performance. Generally, model uncertainty depicts whether the model can best describe the data distribution (Wang et al., 2019). Different from the data uncertainty, model uncertainty can be reduced by feeding more data or knowledge to the model.

Recently, almost all research fields of Artificial Intelligence have been deeply influenced by the Transformer (Vaswani et al., 2017). State-of-the-art Neural Machine Translation (NMT) models are mostly built upon Transformers (Ott et al., 2018b; Dehghani et al., 2018; So et al., 2019; Zhou et al., 2020a; Liu et al., 2020).

	Token-token interactions	Uncertainty	
		Token	Token-token
Transformer	linear	↑	↓ (implicitly)
Our model	linear + nonlinear	↑	↓ (explicitly)

Table 1: Comparison between the vanilla Transformer and our model on the interaction style between tokens and how to deal with the uncertainty. Both models employ the label smoothed cross entropy to properly raise the uncertainty (↑) of determining a single token across the vocabulary. In addition, we **explicitly** reduce the uncertainty (↓) in the dimension of token-token interactions within a certain context to address the predictive uncertainty problem (Xiao and Wang, 2021).

However, Transformer models with the training paradigm of teacher-forcing suffer from the exposure bias problem (Tan et al., 2018; Zhang et al., 2019) and the uncertainty problem (Ott et al., 2018a; Wei et al., 2020; Xiao and Wang, 2021; Shelmanov et al., 2021). Xiao and Wang (2021) and Wei et al. (2020) handle with such problem outside of the model[2]. Namely, manually feeding more unseen samples due to the data uncertainty to the model to reduce the model uncertainty. By contrast, we address the issue inside the model. Given existing training data, we enhance the model representation to better fit the data distribution.

In this paper, we aim to explicitly capture the nonlinear mutual dependencies among tokens during the self-attention calculation and reduce the uncertainty residing in the token-token interactions

[1] Anonymous: https://github.com/self-attention-MI/UE

[2] Note that, the word 'uncertainty' is somewhat heavily reused in the literature. For instance, Xiao and Wang (2021) incorporated uncertainty into the decoding process to reduce the hallucination. In practice, the introduced uncertainty enables the model to see otherwise unseen cases to reduce the model uncertainty in a certain context. Wei et al. (2020) employed the similar presentation. It should be appropriately distinguished from the data uncertainty and the model uncertainty in the literature (Kochkina and Liakata, 2020).

as shown in Table 1. Specifically, we employ the mutual information to measure the nonlinear mutual dependencies between pairs of tokens. Mutual information is a good measure of nonlinear relationships between random variables. To avoid the intractable feature of certain problems by using mutual information, we resort to InfoNCE for mutual information estimation (Logeswaran and Lee, 2018; van den Oord et al., 2019; Gutmann and Hyvärinen, 2012). InfoNCE is a mature framework for unsupervised contrastive learning and it has the theoretical and practical guarantee that a reliable lower bound can be obtained by maximizing it. Experiments on WMT'14 En→De, WMT'14 En→Fr present that the performance of our model has achieved competitive results over the strong baselines and other counterparts. By contrast, to reach the same performance, contrast models either consumes extra training corpus or more trainable parameters.

Contributions and highlights are as follows:

- The proposed idea is simple and makes little change to the model. It can potentially generalize to other pre-trained models leveraging self-attention.
- We explicitly capture nonlinear mutual dependencies between pairs of tokens in decoder self-attentions to reduce the model uncertainty.
- We adopt an unsupervised contrastive learning framework to estimate the mutual information served in the NMT problem.
- We present a detailed analysis of the variants of the model uncertainty before and after enhancing the mutual dependencies.

2 Preliminary

2.1 Mutual Information

Mutual information in discrete distributions is generally described as Equation 1:

$$\begin{aligned} I(X;Y) &= D_{\mathrm{KL}}(p(X,Y)\|p(X)p(Y)) \\ &= \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\ &= \mathbb{E}_{p(x,y)}\left[\log\frac{p(x,y)}{p(x)p(y)}\right], \end{aligned} \tag{1}$$

where, X, Y denote two random variables. x, y indicate concrete samples in X and Y. $p(\cdot)$ and $p(\cdot,\cdot)$ represent marginal probability and joint probability respectively. D_{KL} is the Kullback–Leibler divergence (also known as the *relative entropy*) (Kullback and Leibler, 1951).

2.2 Contrastive Learning

Following Kong et al. (2019), we employ InfoNCE to estimate the mutual information under the contrastive learning framework. InfoNCE maximizes the mutual information to obtain a lower bound, which in practice is a good estimation of mutual information:

$$\begin{aligned} &I(X,Y) \geq \\ &\mathbb{E}_{p(X,Y)}\left[f_{\boldsymbol{\theta}}(x,y) - \mathbb{E}_{q(\tilde{\mathcal{Y}})}\left[\log\sum_{\tilde{y}\in\tilde{\mathcal{Y}}}\exp f_{\boldsymbol{\theta}}(x,\tilde{y})\right]\right] \\ &+\log|\tilde{\mathcal{Y}}|, \end{aligned} \tag{2}$$

Where, x is the positive sample token of the source sentence and y is the positive sample token of the target sentence. f_θ is a measure of relevance between x and y. Usually, a similarity score function is adopted. $\tilde{\mathcal{Y}}$ is the negative sample set of y, note that it contains the positive sample. $q(\cdot)$ is a distribution proposal function offering the specific rule to build the negative sample set. $\tilde{y}$ is a random sample from the negative sample set.

The following part of Equation 2 is the crucial component when we incorporate the contrastive learning framework into NMT problem:

$$\mathbb{E}_{p(X,Y)}\left[f_{\boldsymbol{\theta}}(x,y) - \log\textstyle\sum_{\tilde{y}\in\mathcal{Y}}\exp f_{\boldsymbol{\theta}}(x,\tilde{y})\right]. \tag{3}$$

3 Enhancing the Mutual Dependencies in Transformers

3.1 Motivation to Reduce the Model Uncertainty

As mentioned in Ott et al. (2018a), a well-trained model still spreads too much probability mass across sequences. In other words, model distribution is too spread in hypothesis spaces in that it has to cater the uncertainty brought by the data distribution. Also, as stated in Xiao and Wang (2021), unsuitable tokens attaining considerable probability mass attribute to the uncertainty of the token prediction. Moreover, Wang et al. (2019); Zhou et al. (2020b) present that lower model uncertainty indicates a better fitting of the data distribution. Therefore, in a certain context, the model uncertainty should be reasonably and appropriately reduced.

The widely adopted training paradigm is token-level teacher-forcing in NMT, which notoriously leads to the discrepancy between training and inference, namely, the exposure bias problem (Xie et al., 2016; Ranzato et al., 2016; Norouzi et al., 2016). During inference, model distribution dominates the decoding process. However, high model uncertainty directly indicates unsatisfactory fitting of the data distribution (Zhou et al., 2020b; Xiao and Wang, 2019). Canonical auto-regressive generation can be formulated as Equation 4:

$$p(Y \mid X;\theta) = \prod_{t=1}^{N+1} p\left(y_t \mid y_{<t}, x_{1:M};\theta\right), \quad (4)$$

where, θ denotes the parameters modeling the language model. M is the length of the source sentence and N is the length of the target sentence.

At each time step, clues on the next token are all from previously generated tokens. In other words, it depends on *how much uncertainty on the next token can be reduced by knowing partially generated prefix tokens*. Vanilla Transformer implicitly reduces the uncertainty of token-token interactions during decoding. By contrast, we aim to explicitly reduce the uncertainty of the token-token interactions during the next token generation.

3.2 Contrastive Learning Framework Construction in NMT

Methods to Build the Training Samples: Contrastive learning needs an effective and efficient relevance measure of two tokens. Specifically, a clear distinction should be presented between the similarity score of a positive sample a and a positive sample b and the similarity score of a positive sample a and a negative sample $\tilde{b}$. However, the naive cosine based measure cannot accurately reflect the subtle difference. Therefore, we elaborately design a simple but effective method as Equation 5 and Equation 6:

$$f_\theta(x, y) = f\_sim(x, y) + f\_logit(y), \quad (5)$$

where, $f\_sim(x, y)$ is the cosine similarity score between x and y as usual. $f\_logit(y)$ is the logit (score before $softmax$) by the most confident prediction of y (during inference) or the logit corresponding to the ground-truth token of y (during training).

$$f_\theta(x, \tilde{y}) = f\_sim(x, y) + f\_logit(\tilde{y}), \quad (6)$$

where, the first part of the right-hand side is exactly the same with Equation 5. Difference between Equation 5 and Equation 6 relies on $f\_logit(\cdot)$. Figure 1 depicts how to calculate the concrete value of $f_\theta(a, b)$.

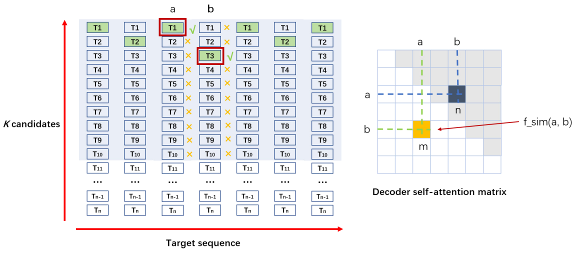

Figure 1: Graphical illustration of how to calculate $f_\theta(a, b)$. a and b denote two positions (tokens) in target sentence. Suppose T_1 and T_3 are ground-truth targets of position a and b respectively. There are two critical components composing $f_\theta(a, b)$, namely $f\_sim(a, b)$ and $logit(b)$ for the pair of a and positive b while $f\_sim(a, b)$ and $logit(\tilde{b})$ for the pair of a and negative sample $\tilde{b}$ from top k candidates. The value of $f\_sim(a, b)$ can be directly fetched from the self-attention matrix. In the left subfigure, negative samples are from the top k candidates in position b marked by '×' or marked by '✓', which offer $logit(\cdot)$. Causal self-attention matrix is demonstrated in the right subfigure. Due to the property of symmetry, there are two $f\_sim(a, b)$ scores of the same value. However, position m is taken into account rather than position n in view of the causal relationship.

Due to the steady state of the pre-trained NMT model, the component f_logit can take up most of the constituent that well distinguishes a legal pair of tokens with contrastive pairs. Moreover, this divergence can further amplify due to the monotonicity of $softmax$ operation. This is a key point our idea leverages to distinguish positive sample pairs from contrastive sample pairs.

Leveraging the Pre-trained Self-attention Logits: To fetch $f\_sim(x, y)$ from multi-head attentions, we need a rational strategy. According to Michel et al. (2019); Voita et al. (2019); Rogers et al. (2020), it is non-trivial to partition these heads into groups. Therefore, we take as similarity scores the average of all heads as follows[3]:

$$f\_sim(X, Y) = \mathrm{Average}\left(\mathrm{head}_1, \ldots, \mathrm{head}_\mathrm{h}\right), \quad (7)$$

where, $\mathrm{Average}$ is the average operation on similarity scores over all self-attention heads. $f\_sim(X, Y)$ contains all pairs of similarity scores between tokens and tokens to be attended.

[3] We employed other methods to do such work, say MAX operation. However, average operation meets our expectation.

Model	BLEU	
	En→De	En→Fr
GNMT+RL Wu et al. (2016)	25.20	40.50
ConvS2S Gehring et al. (2017)	25.16	40.46
Transformer (base) Vaswani et al. (2017)	27.30	38.10
Transformer (big) Vaswani et al. (2017)	28.40	41.80
Evolved Transformer (big) So et al. (2019)	29.80 / 29.20	41.30
Transformer (ADMIN init) Liu et al. (2020)[†]	30.10 / 29.50	43.80 / 41.80
Baseline (WMT only) Ott et al. (2018b)	29.30 / 28.60	43.20 / 41.40
Baseline (WMT+Paracrawl) Ott et al. (2018b)	29.80 / 29.30	42.10 / 40.90
Baseline (Reproduced)	29.75 / 29.30	43.16 / 41.06
Ours (*tokenized* BLEU / *detok.* sacreBLEU)[‡]	30.45 / 29.80	43.67 / 41.51

[†] The model has approx. 40M more parameters than ours.

[‡] Note that, the widely accepted SOTA for WMT14 En-De without other training corpus is 29.90. https://paperswithcode.com/sota/machine-translation-on-wmt2014-english-german. The proposed idea does little change to the model, but achieve the evident improvement that helps the baseline almost hit this score while saves a large amount of parameters.

Table 2: Performance comparison between different models on WMT'14 dataset. 'DS' indicates the proposed regularization method applied on the decoder self-attention. 'ED' means the proposed regularization method applied on the cross encoder-decoder attention in the decoder. Our results are based on the reproduced results. Default values are case-sensitive *tokenized* BLEU scores and otherwise a pair of (case-sensitive *tokenized* BLEU) / (*detok.* sacreBLEU). BLEU scores are based on newstest2014 for WMT'14 English-German (En→De) and WMT'14 English-French (En→Fr). Checkpoint averaging is not used in our results. For WMT'14 En→De, we use the configuration of $L_{3,4,5}$+DS+ED and $k = 40$. For WMT'14 En→Fr, we use the configuration of $L_{3,4,5}$+DS+ED and $k = 50$.

Combination objective: The overall objective consists of the label smoothed cross entropy and another two custom objectives based on mutual information maximization constraints as follows:

$$\begin{aligned} loss &= (1 - \alpha - \beta) \times lce\_loss \\ &\quad + \alpha \times mi\_loss\_cross \\ &\quad + \beta \times mi\_loss\_self, \end{aligned} \tag{8}$$

where, lce_loss indicates the label smoothed cross entropy loss, mi_loss_cross represents the mutual information constraints on encoder-decoder attention and mi_loss_self denotes the mutual information constraints on decoder self-attention. Both of them are defined and estimated as Equation 2. α and β are hyperparameters to balance the label smoothed cross entropy loss and two custom losses.

4 Experiments

In this section, we describe the details of our experiments. We evaluate our model on WMT'14 English-German (WMT'14 En→De) and WMT'14 English-French (WMT'14 En→Fr) datasets. Moreover, we conduct ablation studies to assess the effectiveness of different objectives and hyperparameters setup.

4.1 Experimental Setup

We implement our model based on the official Fairseq toolkit implemented by PyTorch[4] (Ott et al., 2019).

4.1.1 Dataset and Metric

We train our model on WMT'16 English-German (En→De, 4.5M)[5] and WMT'14 English-French (En-Fr, 36M). For WMT'14 En→De, we validate our model on newstest13 and test on newstest2014. Following Ott et al. (2018b), we use byte pair encoding (BPE) (Sennrich et al., 2016) to prepare the joint vocabulary of 32K symbols. For WMT'14 En→Fr, we validate our model on newstest12+13 and test on newstest14. The joint vocabulary is 40K. We mainly use two BLEU metrics to evaluate our performance, namely, case-sensitive *tokenized* BLEU and *detokenized* sacreBLEU. When necessary, compound split BLEU is also reported. We report BLEU scores with a beam size of 4 and a length penalty of 0.6.

[4] https://github.com/pytorch/fairseq

[5] To be consistent with the baseline and other counterparts, we use WMT'16 En→De to train our model.

	Models[†]					
w_1	0.4	0.5	0.6	0.7	0.8	0.9
w_2	0.6/2	0.5/2	0.4/2	0.3/2	0.2/2	0.1/2
L_5+DS+ED	30.19/29.50	30.26/29.60	30.29/29.60	30.26/29.60	30.13/29.50	30.04/29.40
L_0+DS+ED	30.22/29.50	30.30/29.60	30.34/29.60	30.27/29.60	30.37/29.80	30.09/29.50
w_2	0.6	0.5	0.4	0.3	0.2	0.1
L_5+DS	30.09/29.40	30.24/29.50	30.41/29.70	30.21/29.60	30.30/29.70	30.08/29.50
L_5+ED	30.12/29.40	30.31/29.60	30.25/29.50	30.21/29.60	30.25/29.70	30.08/29.50
L_0+DS	30.10/29.40	30.22/29.50	30.39/29.70	30.23/29.60	30.22/29.60	30.09/29.50
L_0+ED	30.06/29.40	30.38/29.70	30.23/29.50	30.24/29.60	30.28/29.70	30.15/29.50
w_2	0.6/2	0.5/2	0.4/2	0.3/2	0.2/2	0.1/2
$L_{0,5}$+DS	30.28/29.60	30.29/29.60	30.42/29.70	30.34/29.70	30.26/29.60	30.17/29.60
$L_{0,5}$+ED	30.22/29.50	30.29/29.60	30.29/29.60	30.17/29.50	30.32/29.70	30.20/29.60
$L_{4,5}$+DS	30.27/29.60	30.31/29.60	30.43/29.70	30.41/29.70	30.30/29.70	30.19/29.60
$L_{4,5}$+ED	30.14/29.40	30.27/29.60	30.25/29.60	30.24/29.60	30.25/29.70	30.22/29.70
$L_{0,1}$+DS	30.27/29.60	30.38/29.70	30.46/29.70	30.35/29.70	30.30/29.70	30.18/29.60
$L_{0,1}$+ED	30.06/29.30	30.24/29.60	30.27/29.60	30.19/29.60	30.28/29.70	30.18/29.60
w_2	0.6/3	0.5/3	0.4/3	0.3/3	0.2/3	0.1/3
$L_{0,1,2}$+DS	30.26/29.60	30.29/29.60	30.42/29.70	30.38/29.70	30.29/29.70	30.16/29.60
$L_{0,1,2}$+ED	30.07/29.40	30.27/29.60	30.28/29.60	30.23/29.60	30.26/29.70	30.14/29.60
$L_{3,4,5}$+DS	30.21/29.50	30.24/29.50	30.46/29.70	30.42/29.70	30.30/29.70	30.13/29.60
$L_{3,4,5}$+ED	30.14/29.50	30.18/29.50	30.28/29.60	30.23/29.60	30.25/29.70	30.19/29.60
w_2	0.6/4	0.5/4	0.4/4	0.3/4	0.2/4	0.1/4
$L_{1,2,3,4}$+DS	30.27/29.60	30.30/29.60	30.44/29.70	30.32/29.70	30.27/29.70	30.16/29.60
$L_{1,2,3,4}$+ED	30.18/29.50	30.19/29.60	30.20/29.50	30.33/29.70	30.21/29.60	30.22/29.70
$L_{0,1,2,3}$+DS	30.22/29.50	30.31/29.60	30.39/29.70	30.37/29.70	30.31/29.70	30.19/29.60
$L_{0,1,2,3}$+ED	30.15/29.40	30.22/29.50	30.18/29.50	30.27/29.60	30.29/29.70	30.29/29.60
$L_{2,3,4,5}$+DS	30.25/29.50	30.30/29.60	30.40/29.70	30.35/29.70	30.34/29.70	30.20/29.60
$L_{2,3,4,5}$+ED	30.12/29.40	30.23/29.60	30.24/29.60	30.28/29.70	30.23/29.70	30.22/29.60
w_2	0.6/5	0.5/5	0.4/5	0.3/5	0.2/5	0.1/5
L_{all-0}+DS	30.27/29.60	30.29/29.60	30.36/29.60	30.33/29.70	30.26/29.60	30.15/29.60
L_{all-0}+ED	30.12/29.40	30.21/29.60	30.24/29.60	30.31/29.70	30.27/29.70	30.18/29.60
L_{all-5}+DS	30.24/29.50	30.29/29.60	30.47/29.70	30.33/29.70	30.27/29.70	30.12/29.60
L_{all-5}+ED	30.17/29.50	30.15/29.50	30.18/29.50	30.27/29.60	30.27/29.70	30.19/29.60
w_2	0.6/6	0.5/6	0.4/6	0.3/6	0.2/6	0.1/6
L_{all}+DS	30.25/29.50	30.20/29.60	30.44/29.70	30.33/29.70	30.27/29.60	30.16/29.60
L_{all}+ED	30.12/29.40	30.26/29.60	30.22/29.50	30.31/29.70	30.24/29.70	30.15/29.60

[†] We tune the parameters on the validation set, and report these results on the test set. Values in this table may be susceptible to different setups that we did not thoroughly explore. However, we do not aim to provide the best situations of all cases, instead, we offer analysis of possible trends. We ignore the influence of k and set $k = 10$ in these experiments.

Table 3: Ablation studies on the layer-level performance. 'DS' indicates the proposed regularization approach applied on the decoder self-attention. 'ED' means the proposed regularization approach applied on the cross encoder-decoder attention in the decoder. To simplify the experiments, we adopt the same parameter w_2 to balance 'DS' and 'ED'. For instance, $w_2 = (1 - w_1)/2$, when 'DS' and 'ED' are applied on a single layer of the decoder. Besides, $w_2 = (1 - w_1)/6$, when 'DS' or 'ED' are applied on all layers of the decoder, and so on. Different contributions of 'DS' or 'ED' in the combination fashion of 'DS+ED', we leave them in the future work. L_0 means the first layer in the decoder. L_5 means the last layer. $L_{0,5}$ means the first layer and the last layer. $L_{4,5}$ means the last two layers. $L_{0,1}$ means the first two layers. $L_{0,1,2}$ means the first three layers. $L_{3,4,5}$ means the last three layers. L_{all-0} means all layers except the first layer. L_{all-5} means all layers except the last layer. We average the last 5 checkpoints to report these results. Experiments are conducted on WMT'14 En→De. From these results, we can infer that 'DS' has slight better performance compared with 'ED'. Employing either 'DS' or 'ED' on all layers of the decoder is somewhat over-constraint. In a certain range, appropriately adding regularization can be effective in improving performance.

k	1	2	3	4	5	10	20	30	40	50	100	200
BLEU	27.52	27.63	27.77	27.79	27.86	27.79	27.89	27.85	27.92	27.89	27.91	-

Table 4: The impact of different choices of k (regarding to the capacity of a negative sample set) on performance. The experiment is conducted on WMT'14 En→De valid set. Combination of two regularizations (ED+DS) is adopted. Here, the metric 'BLEU' indicates case-sensitive *tokenized* BLEU. In the case of $k = 200$, the model hits the OOM under the same setup of other configurations. We use $k = 40$ to report the final result of WMT'14 En→De. Similarly, we use $k = 50$ to report the final result of WMT'14 En→Fr.

Dropout Type	Model Acquisition	En→De		En→Fr	
		UE (before)	UE (after)	UE (before)	UE (after)
MC-all	Sampled max. probability	354.5077	337.3681	166.6318	146.3338
MC-all	Mean entropy	2515.1008	2457.2503	1215.0922	1137.0944
MC-all	BALD-VR	339.2128	334.9575	114.1011	108.4149

Table 5: Variation of the model uncertainty before fine-tuning and after fine-tuning. 'MC-all' means 'Monte Carlo Dropout' employed on all layers. We employ three Uncertainty Estimation (UE) methods, namely, Sampled max. probability, Mean Entropy and BALD-VR to investigate the variations. The number of forward passes T is 10. The results are not normalized over the number of tokens.

4.1.2 Model and Hyperparameters

Our model leverages the pre-trained baseline model, which is an extension of the Transformer big model ($d_{model} = d_{hidden} = 1024$, $n_{layer} = 6$, $n_{head} = 16$) (Vaswani et al., 2017). We adopt Adam (Kingma and Ba, 2015) to optimize our model by setting $\beta_1 = 0.90$, $\beta_2 = 0.98$ and $\epsilon = 1e\text{-}08$. We finetune our model from a pre-trained checkpoint with the learning rate 3e-04 for En→De and 5e-04 for En→Fr. Our criterion to configure 'ntokens' and 'update-freq' is that, neither hitting the OOM nor the threshold of the loss scale. 'ntokens' is 10240 for En→De and 9216 for En→Fr. 'update-freq' is 1 for En→De and 4 for En→Fr. The maximum epoch for En→De is 20 and 10 for En→Fr. Embeddings are shared in all positions. We tune hyperparameters on the validation set.

All experiments are conducted on a machine with 8 NVIDIA TITAN RTX GPU and memory-efficient version of FP16 half-precision training.

4.1.3 Performance Analysis

Table 2 demonstrates the performance comparison of our model and the baseline models along with other SOTA models on WMT'14 dataset. For a fair comparison, we depict both the case-sensitive *tokenized* BLEU and *detokenized* SacreBLEU (Post, 2018)[6].

From these results, it is apparent to infer that our model achieves a competitive improvement over the strong baselines and other SOTA models. Since our model does minute change to the baseline model, the improvements are reasonable and justified.

4.2 Ablation Study

Hyperparameter k in Contrastive Learning Framework Construction: According to Kong et al. (2019), the larger the capacity of the negative sample set, the more accurate the framework is to estimate the lower bound of mutual information. Also, as we demonstrated in Equation 2 and Equation 3, the lower bound becomes even tighter when the number of tokens in the negative sample set is large enough. We conduct experiments with different hyperparameter k as shown in Table 4, in which we can infer that capacity of a negative sample set has a positive impact on performance in a certain range. In the case of $k = 1$, model performance is not far from satisfactory, which is due to the pre-trained nature of NMT model. In other words, a pre-trained NMT model itself is a competent distribution proposal function.

Contribution of Different Objectives: We employ two hyperparameters α and β to balance different losses as shown in Equation 8. In practice, to simplify the experiments and cover as more combinations as possible, we adopt another two hyperparameters w_1 and w_2 to balance the loss of the label smoothed cross entropy and the proposed regularization methods in ablation studies. We validate the effectiveness of the proposed mutual informa-

[6] SacreBLEU hash: BLEU+case.mixed+lang.en-de+ numrefs.1+smooth.exp+test.wmt14/full+tok.13a+version.1.4.14

Num. of T	1	2	3	4	5	6
SMP	338.0088 / 319.5488	347.5487 / 329.9464	350.2366 / 333.0439	351.9552 / 334.9495	353.7504 / 335.7504	353.4781 / 336.2595
ME	2403.5835 / 2341.8491	2460.3462 / 2400.5967	2479.6318 / 2421.1494	2492.6663 / 2435.1404	2500.8201 / 2441.8916	2504.8918 / 2445.9519
BALD-VR	0 / 0 [†]	154.9255 / 150.7553	214.0106 / 210.4574	251.8404 / 246.7234	275.8936 / 270.2872	294.9787 / 288.6808
Num. of T	7	8	9	10	20[‡]	30[‡]
SMP	353.6949 / 336.5727	353.9379 / 336.8132	354.3253 / 337.1445	354.5077 / 337.3681	176.3070 / 168.1396	87.0544 / 83.3469
ME	2507.3079 / 2449.6633	2509.6550 / 2451.1414	2512.8601 / 2454.7310	2515.1008 / 2457.2503	1249.8004 / 1224.1233	615.8340 / 605.8625
BALD-VR	307.9149 / 303.4787	321.2128 / 315.9893	331.2021 / 326.0425	339.2128 / 334.9575	193.9734 / 192.5053	101.8218 / 101.2766

[†] Zero values are due to the calculation of variance towards a single value.
[‡] In the case of $T = 20$ and $T = 30$, results seem to be disproportionate to other cases. This is due to the setup of batch size during inference in order to avoid OOM.

Table 6: The impact of the number of forward passes T on MC dropout inference. We show the variations of the three metrics. 'SMP' for 'sampled maximum probability'; 'ME' for 'mean entropy'; 'BALD-VR' for a combination of 'Bayesian Active Learning by Disagreement' and 'variation ratio'. The values presented here are UE (before) / UE (after). Experiments are conducted on WMT'14 En→De. Dropout ratio p is the default value 0.3. We can infer that as the value T increases, the gap between two UEs tends to decrease. However, UE (after) is consistently smaller than UE (before). Considering the practical situation and following the common literature, we choose $T = 10$ throughout the experiments.

tion constraints by setting hyperparameter w_1 from 0.4 to 0.9 and w_2 from 0.6 to 0.1 (when it comes to multiple layers regularization, w_2 is equally divided.). Results of comparison are depicted in Table 3. From Table 3, it is intuitive to infer that both custom objectives have a positive impact on the model performance. 'DS' performs slightly better than 'ED'. Case $w_1 = 0.4$ and case $w_1 = 0.9$ are contrast groups.

Impact of the Proposed Regularization Methods on Different Layers of the Decoder: We conduct ablation experiments of regularization on layer level performance in this section. Results are presented in Table 3. From Table 3, it can be inferred that there is no absolutely positive relationship between the increase in performance and the increase in regularization on more layers. To a certain extent, appropriately adding regularization can be effective in improving performance. However, too many regularization can lead to performance degradation. We speculate that it is caused by over-regularization. Therefore, considering the performance and the overhead, we recommend that the number of regularization layers should be less than 3.

4.3 Analysis

Variation of Model Uncertainty: Bayesian Neural Networks are widely used to build a posterior distribution over model parameters given the bilingual training data $\mathcal{D}_{\text{train}}$ (Wang et al., 2019; Xiao and Wang, 2019):

$$\begin{aligned} & P\left(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D}_{\text{train}}\right) \\ = & \int P\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\theta}\right) P\left(\boldsymbol{\theta} \mid \mathcal{D}_{\text{train}}\right) d\boldsymbol{\theta}. \end{aligned} \tag{9}$$

Following Shelmanov et al. (2021); Zhou et al. (2020b); Xiao and Wang (2019); Wang et al. (2019), we employ Monte Carlo Dropout (Gal and Ghahramani, 2016) to approximate Bayesian inference to conduct the Uncertainty Estimation (UE). Specifically, we demonstrate the quantification of model uncertainty before and after the fine-tuning to investigate the variation:

$$\begin{aligned} & UE(\boldsymbol{\theta}) \\ = & \frac{1}{N} \sum_{n=1}^{N} \operatorname{Var}\left[P\left(y^n \mid x^n, \hat{\boldsymbol{\theta}}^k\right)\right]_{k=1}^{K}, \end{aligned} \tag{10}$$

where, θ is the set of model parameters. N indicates the number of samples. K is the number of stochastic passes. $\left\{\hat{\theta}^1, ..., \hat{\theta}^K\right\}$ are sampled parameters during stochastic passes. To be consistent with Wang et al. (2019), we calculate the uncertainty after the prediction process is done in that we do not employ the model uncertainty to improve the model prediction, instead, we quantify the model uncertainty.

However, in practice, due to the intrinsic character of multidimensional data in NMT problem, it is non-trivial to calculate the model uncertainty solely based on Equation 10[7]. Therefore, following Hazra et al. (2021), we employed a combination of BALD (Bayesian Active Learning Disagreement) (Houlsby et al., 2011) and Variation Ratio (Kochkina and Liakata, 2020) to conceptually form a new metric BALD-VR. Along with BALD-VR, we also use Mean Entropy (Kochkina and Liakata, 2020) and Sampled Maximum Probability (Shelmanov et al., 2021) to evaluate the model uncertainty, results are shown in Table 5.

[7] Results are not consistent under different circumstances

dropout ratio p	0.1	0.2	0.3[†]	0.4	0.5
SMP	302.3890 / 286.0438	323.7969 / 306.6345	354.5077 / 337.3681	403.9660 / 388.3170	~~495.5341 / 485.3623~~
ME	2057.5542 / 1990.6696	2240.8325 / 2173.9890	2515.1008 / 2457.2503	2962.1492 / 2926.7832	~~3779.8779 / 3796.4238~~
BALD-VR	234.0745 / 231.3511	285.9575 / 282.3511	339.2128 / 334.9575	406.0213 / 403.2021	~~529.4787 / 537.0319~~
dropout ratio p	0.6	0.7	0.8	0.9	1.0
SMP	~~698.8461 / 703.8344~~	~~890.4090 / 887.0627~~	~~940.6628 / 943.8118~~	~~955.7371 / 955.7843~~	~~868.1199 / 868.6059~~
ME	~~5537.7705 / 5691.3364~~	~~7761.6455 / 7963.3516~~	~~9321.2520 / 9468.3799~~	~~9783.8789 / 9785.2402~~	~~5698.2153 / 5684.1841~~
BALD-VR	~~803.1170 / 823.4362~~	~~954.4681 / 955.8192~~	~~957.7553 / 957.7553~~	~~957.7553 / 957.7553~~	~~0 / 0~~

[†] There are three main types of dropout operation in the implementation of Transformer model, namely, dropout for layer output, dropout for attention weights and dropout for activation in FFN. Here, we refer 'dropout' to the first case. Note that, 0.3 is the default value for WMT'14 En→De model.

Table 7: The impact of the dropout ratio p on MC dropout inference. We show the variations of the three metrics. 'SMP' for 'sampled maximum probability'; 'ME' for 'mean entropy'; 'BALD-VR' for a combination of 'Bayesian Active Learning by Disagreement' and 'variation ratio'. The values presented here are UE (before) / UE (after). Experiments are conducted on WMT'14 En→De. The number of forward passes T is 10. From the results above, we can infer that the appropriate value of the dropout ratio p is no more than 0.4, which is in line with our expectations.

Ablation Studies on MC Dropout Inference There are two main factors that affect the MC dropout inference. Namely, the number of forward passes T and the dropout ratio p. We investigate such factors in this section. For the former, we change the number of forward passes T and observe the results. From Table 6, it can be seen that three UE metrics are robust to depict the variants of the model uncertainty. Within a certain range, the larger the number of forward passes, the more objective the results reflect. For the latter, we keep the number of forward passes T constant and change the dropout ratio p. We have the prior knowledge that p cannot be too large whether during training or during inference. From Table 7, it is apparent that the results are as we expected and again verify our hypothesis.

Correlation with the Label Smoothed Cross Entropy: There is no conflict between the widely adopted label smoothed cross entropy (raising uncertainty) and the proposed idea (reducing uncertainty) in that they perform in the different dimensions. For clarity, label smoothing loosens a one-hot label to a soft alternative, which occurs from the viewpoint of a single token across the vocabulary. It aims to penalize the over-confidence of the model, namely raising the model uncertainty towards a single token decision. While our approach reduces the uncertainty existing in the interactions between token and token in a certain context. It occurs from the perspective of the token-token interactions, especially when a certain context is held during decoding. By contrast, our model pays attention to the inevitably introduced uncertainty that takes up non-negligible probability mass (Ott et al., 2018a). Therefore, the proposed idea is a companion to the label smoothed cross entropy rather than a replacement or alternative.

5 Conclusion

In this paper, we propose a novel regularization method based on maximization of mutual information. We implement our ideas under the unsupervised contrastive learning framework to capture and enhance nonlinear mutual dependencies among tokens, which reduces the model uncertainty. Experiments and ablation studies demonstrate the consistent effectiveness of our approach. Besides, analysis of model uncertainty quantification again verifies our hypothesis.

Limitation and Future Work: To simplify the ablation studies, we employ the same weights on 'DS' and 'ED'. Whether there will be further performance gains when taking into account regularization on different encoder layers, we will leave in the future work. Besides, our idea is based on the self-attention mechanism, which serves plenty of pre-trained language models. Nonlinear mutual dependencies may potentially have a positive influence on these models for downstream tasks. This is the first step we take to investigate how to incorporate the model uncertainty analysis into NMT problem.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2020. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *arXiv preprint arXiv:2011.06225*.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2018. Universal transformers. In *International Conference on Learning Representations*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The journal of machine learning research*, 13(1):307–361.

Rishi Hazra, Parag Dutta, Shubham Gupta, Mohammed Abdul Qaathir, and Ambedkar Dukkipati. 2021. Active^2 learning: Actively reducing redundancies in active learning methods for sequence tagging and machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1982–1995, Online. Association for Computational Linguistics.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Elena Kochkina and Maria Liakata. 2020. Estimating predictive uncertainty for rumour verification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6964–6981, Online. Association for Computational Linguistics.

Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2019. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.

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, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018a. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

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 *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9.

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 *International Conference on Learning Representations*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

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.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online. Association for Computational Linguistics.

David So, Quoc Le, and Chen Liang. 2019. The evolved transformer. In *International Conference on Machine Learning*, pages 5877–5886.

Bowen Tan, Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. 2018. Connecting the dots between mle and rl for sequence generation.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802, Hong Kong, China. Association for Computational Linguistics.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Luxi Xing, and Weihua Luo. 2020. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2724–2735, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7322–7329.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2016. Data noising as smoothing in neural network language models.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2020a. Improving autoregressive NMT with non-autoregressive model. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, pages 24–29, Seattle, Washington. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020b. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.