Distilling Knowledge Learned in BERT for Text Generation

Yen-Chun Chen[1], **Zhe Gan**[1], **Yu Cheng**[1], **Jingzhou Liu**[2], **Jingjing Liu**[1]
[1]Microsoft Dynamics 365 AI Research [2]Carnegie Mellon University
{yen-chun.chen,zhe.gan,yu.cheng,jinjl}@microsoft.com; liujingzhou@cs.cmu.edu

arXiv:1911.03829v3 [cs.CL] 17 Jul 2020

Abstract

Large-scale pre-trained language model such as BERT has achieved great success in language understanding tasks. However, it remains an open question how to utilize BERT for language generation. In this paper, we present a novel approach, Conditional Masked Language Modeling (C-MLM), to enable the finetuning of BERT on target generation tasks. The finetuned BERT (*teacher*) is exploited as extra supervision to improve conventional Seq2Seq models (*student*) for better text generation performance. By leveraging BERT's idiosyncratic bidirectional nature, distilling knowledge learned in BERT can encourage auto-regressive Seq2Seq models to plan ahead, imposing global sequence-level supervision for coherent text generation. Experiments show that the proposed approach significantly outperforms strong Transformer baselines on multiple language generation tasks such as machine translation and text summarization. Our proposed model also achieves new state of the art on IWSLT German-English and English-Vietnamese MT datasets.[1]

1 Introduction

Large-scale pre-trained language model, such as ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2019), has become the *de facto* first encoding step for many natural language processing (NLP) tasks. For example, BERT, pre-trained with deep bidirectional Transformer (Vaswani et al., 2017) via masked language modeling and next sentence prediction, has revolutionized the state of the art in many language understanding tasks, such as natural language inference (Bowman et al., 2015) and question answering (Rajpurkar et al., 2016).

However, beyond common practice of finetuning BERT for language understanding (Wang et al., 2019), applying BERT to language generation still remains an open question. Text generation aims to generate natural language sentences conditioned on certain input, with applications ranging from machine translation (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015), text summarization (Nallapati et al., 2016; Gehring et al., 2017; Chen and Bansal, 2018), to image captioning (Vinyals et al., 2015; Xu et al., 2015; Gan et al., 2017). In this work, we study how to use BERT for better text generation, which is still a relatively unexplored territory.

Intuitively, as BERT is learned with a generative objective via Masked Language Modeling (MLM) during the pre-training stage, a natural assumption is that this training objective should have learned essential, bidirectional, contextual knowledge that can help enhance text generation. Unfortunately, this MLM objective is not auto-regressive, which encumbers its direct application to auto-regressive text generation in practice.

We tackle this challenge by proposing a novel and generalizable approach to distilling knowledge learned in BERT for text generation tasks. We first propose a new Conditional Masked Language Modeling (C-MLM) task, inspired by MLM but requiring additional conditional input, which enables finetuning pre-trained BERT on a target dataset. In order to extract knowledge from the finetuned BERT and apply it to a text generation model, we leverage the finetuned BERT as a teacher model that generates sequences of word probability logits for the training samples, and treat the text generation model as a student network, which can effectively learn from the teacher's outputs for imitation. The proposed approach improves text generation by providing a good estimation on word probability distribution for each token in a sentence, consum-

[1]Code is available at https://github.com/ChenRocks/Distill-BERT-Textgen.

ing both the left and the right context, the exploitation of which encourages conventional text generation models to *plan ahead*. At inference time, the teacher model (BERT) is not required thus the decoding speed is as fast as the underlying student model.

Text generation models are usually trained via Maximum Likelihood Estimation (MLE), or *teacher forcing* (Bengio et al., 2015): at each time step, it maximizes the likelihood of the next word conditioned on its previous ground-truth words. This corresponds to optimizing one-step-ahead prediction. As there is no explicit signal towards global planning in the training objective, the generation model may incline to focusing on local structure rather than global coherence. With our proposed approach, BERT's *looking into the future* ability can act as an effective regularization method, capturing subtle long-term dependencies that ensure global coherence and in consequence boost model performance on text generation.

An alternative way to leverage BERT for text generation is to initialize the parameters of the encoder or decoder of Seq2Seq with pre-trained BERT, and then finetuning on the target dataset. However, this approach requires the encoder/decoder to be identical to BERT, inevitably making the final text generation model too large. Our approach, on the other hand, is modular and compatible to any text-generation model, and has no restriction on model size or model architecture (e.g., LSTM or Transformer).

The main contributions of this work are threefold: (i) We present a novel approach to utilizing BERT for text generation. The proposed method induces sequence-level knowledge into the conventional one-step-ahead and teacher-forcing training paradigm, by introducing an effective regularization term to MLE training loss. (ii) We conduct comprehensive evaluation on multiple text generation tasks, including machine translation and text summarization. Experiments show that our proposed approach significantly outperforms strong Transformer baselines and is generalizable to different tasks. (iii) The proposed model achieves new state of the art on both IWSLT14 German-English and IWSLT15 English-Vietnamese datasets.

2 Related Work

Pre-trained Language Models Prior to large-scale pre-trained language model, word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) were widely used for NLP tasks. Recently, CoVe (McCann et al., 2017) introduced (conditional) language models pre-trained on paired machine translation corpus. ELMo (Peters et al., 2018) learned a contextual language model on a large corpus with bidirectional RNN. GPT (Radford et al., 2018) used unidirectional Transformer to achieve better contextualized word representation. By fine-tuning pre-trained language models, ULMFit (Howard and Ruder, 2018) also achieved promising results on text classification.

In our study, we focus on BERT due to its superior performance on multiple language understanding tasks. However, different from previous work exploiting BERT for language understanding tasks, here we aim to apply BERT to text generation. To the best of our knowledge, this is still a relatively unexplored space. The proposed approach is also model-agnostic and can be applied to other pre-trained language models as well.

BERT for Text Generation There has been some recent attempt on applying BERT to text generation. Specifically, Lample and Conneau (2019) trained cross-lingual MLM and demonstrated promising results for cross-lingual natural language inference (Conneau et al., 2018) and unsupervised neural machine translation (NMT) (Lample et al., 2018). Wang and Cho (2019) formulated BERT as a Markov Random Field LM and showed preliminary results on unsupervised text generation with improved diversity. Zhang et al. (2019a) utilized an encoder with BERT and a two-stage decoder for text summarization. Song et al. (2019) proposed Masked Seq2Seq (MASS) pre-training, demonstrating promising results on unsupervised NMT, text summarization and conversational response generation. Concurrent with our work, Ghazvininejad et al. (2019) proposed a similar conditional MLM for constant-time translation, and Yang et al. (2019) studied how to fine-tune BERT for NMT.

Our approach is novel in the sense that we do not directly use the parameters of BERT in the Seq2Seq model. Instead, BERT acts as an effective regularization to the MLE training loss, by proactively injecting future information for predicting the present.

Right-to-Left Generation Our work also shares a high-level intuition with those approaches that try to regularize left-to-right generative models with

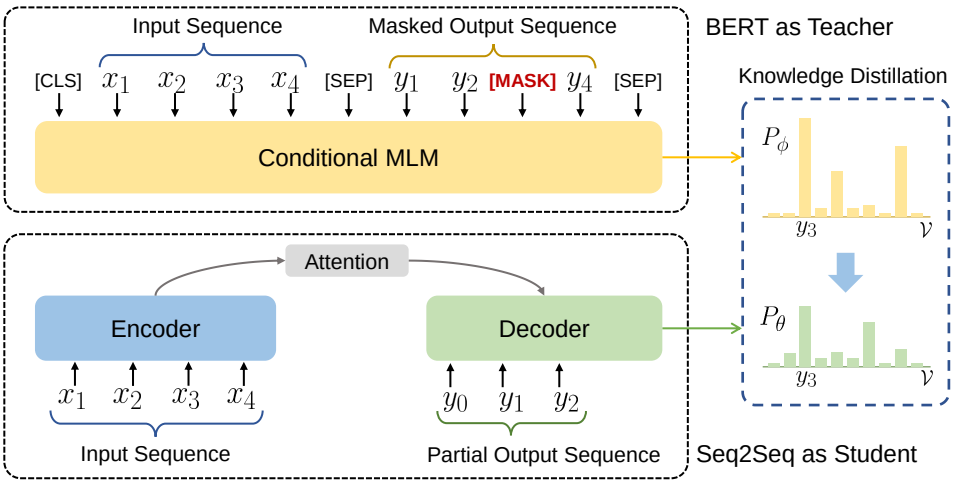

Figure 1: Illustration of distilling knowledge from BERT for text generation. See Section 3.2 and 3.3 for details.

a right-to-left counterpart. Specifically, Liu et al. (2016) trained a separate reverse NMT and performed joint decoding at inference time to enforce agreement between forward and reverse models. Twin Networks (Serdyuk et al., 2018) used a backward RNN jointly trained with a forward RNN decoder by matching their hidden states. Zhang et al. (2019b) further extended the idea to Transformer with joint training, so that the forward and the backward models iteratively improve each other. Our proposed approach stems from a similar intuition. However, we focus on using pre-trained language model such as BERT to regularize an auto-regressive generation model.

Knowledge Distillation Our method shares the same loss formulation as Knowledge Distillation (KD) proposed in Bucilu et al. (2006); Hinton et al. (2015); Kim and Rush (2016), where a smaller student model is trained on soft labels provided by a larger teacher model. More recently, Tan et al. (2019) applied KD to multilingual NMT, and Sun et al. (2019) proposed patient KD for BERT model compression. Compared with these previous studies, where both the teacher and the student are trained on the same task, our approach is different in the sense that the BERT teacher is not designed to perform the student's generation task. We focus on using KD to leverage the learned knowledge in BERT for text generation, while previous work mostly focused on model compression.

3 Approach

In this section, we present our proposed approach to distilling the knowledge in BERT for text generation in generic sequence-to-sequence (Seq2Seq) setting. We first review Seq2Seq learning in Section 3.1, and then describe the proposed approach in Section 3.2 and 3.3.

3.1 Sequence-to-Sequence Learning

Seq2Seq learning (Sutskever et al., 2014) aims to generate a sequence of discrete output $Y = (y_1, \ldots, y_N)$ of length N, conditioned on a sequence of discrete input $X = (x_1, \ldots, x_M)$ of length M. A Seq2Seq model learns parameters θ to estimate the conditional likelihood $P_\theta(Y|X)$, typically trained via Maximum Likelihood Estimation (MLE), or equivalently, minimizing the cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{xe}(\theta) &= -\log P_\theta(Y|X) \\ &= -\sum_{t=1}^{N} \log P_\theta(y_t|y_{1:t-1}, X)\,,\end{aligned} \tag{1}$$

where each conditional probability can be calculated via an attention-based recurrent neural network (RNN) (Bahdanau et al., 2015; Luong et al., 2015), Transformer (Vaswani et al., 2017), or any other neural sequence-generation models.

3.2 Finetune BERT with Conditional MLM

This generic Seq2Seq learning framework is the state of the art on a wide range of text generation tasks. Using modern deep neural networks, the conditional probabilities can be readily modeled as a sequence of classifications over the word vocabulary. However, during training, in order to generate the t-th token y_t, the model only sees a partial sentence $y_{1:t-1}$ from the ground-truth training data. Intuitively, it is reasonable to assume that a bidirectional model can be more informative than a left-

to-right generation model, since additional context from the right (or future) is also incorporated to predict the current word. Unfortunately, this additional information is not utilized in a standard Seq2Seq model, since it can only be trained in a left-to-right manner, where the future context is masked out to prevent each word from indirectly "*seeing itself*". To compensate this single-directional limitation of Seq2Seq setting, we propose a new conditional language model (C-MLM) to enable the finetuning of BERT on target generation task, in hope that the finetuned bidirectional BERT can be utilized for better text generation.

BERT (Devlin et al., 2019) is a deep bidirectional Transformer trained via Masked Language Modeling (MLM).[2] In a similar setting, where the input is a sequence pair (X, Y),[3] 15% of the tokens are randomly masked. Formally, we denote the masked token sets as X^m and Y^m, and the disjoint counterpart (*i.e.*, the unmasked tokens) as X^u and Y^u, respectively. The trained BERT model aims to estimate the joint probability:

$$P(x_1^m, \ldots, x_i^m, y_1^m, \ldots, y_j^m | X^u, Y^u)\,, \qquad (2)$$

where i and j denote the number of masked tokens in X and Y, respectively. Each $x_\star^m \in X^m$, and each $y_\star^m \in Y^m$. Eqn. (2) can be trained with the standard word-level cross-entropy loss.

We aim to marry MLM pre-training with Seq2Seq learning, to leverage bidirectional language model for text generation. To this end, we propose a conditional-MLM, a variant of MLM that allows further finetuning of pre-trained BERT on target dataset. For example, for machine translation, X and Y represent the source and the target sentence, respectively. We first concatenate them together and randomly mask 15% of the tokens only in Y, then train the network to model the joint probability:

$$P(y_1^m, \ldots, y_j^m | X, Y^u)\,. \qquad (3)$$

The above C-MLM objective is similar to the conditional language modeling (LM) objective in Eqn. (1), but conditional LM only permits predicting a word based on its left context. C-MLM is also related to Masked Seq2Seq (MASS) pre-training (Song et al., 2019). However, in MASS, the encoder takes a sentence with randomly masked fragment (several consecutive tokens) as input, and the decoder tries to predict this masked fragment, which is different from our model design. The final goal is also different: MASS focuses on Seq2Seq pre-training, while we focus on leveraging BERT for text generation. In our experiments, we observe that the C-MLM task can obtain high accuracy and good generalization on word prediction. However, it is not feasible to generate sequential output directly from C-MLM. Instead, we use knowledge distillation to distill the knowledge learned from the finetuned BERT into a Seq2Seq model for direct text generation, which will be explained in the next sub-section.

3.3 Knowledge Distillation for Generation

Our inspiration springs from the observation that the probability distribution of the masked word y_t^m is estimated using both $y_{1:t-1}^u$ and $y_{t+1:N}^u$ from Y^u. In other words, the distribution for a given word $P(y_t^m | X, Y^u)$ contains information from both backward and forward contexts, which is a desirable benefit for providing sequence-level global guidance. This probability distribution can be considered as soft targets for a text generation model to mimic from, which potentially contains more useful and fine-grained information than the usual hard-assigned, one-hot label, therefore enhancing conventional left-to-right generation models to *look into the future*.

In a knowledge distillation setting, the BERT model can be considered as a *teacher*, while the Seq2Seq model acts as a *student*. Specifically, the Seq2Seq model can be trained with the following objective function:

$$\mathcal{L}_{bidi}(\theta) = -\sum_{w \in \mathcal{V}} \Big[P_\phi(y_t = w | Y^u, X) \cdot \log P_\theta(y_t = w | y_{1:t-1}, X) \Big]\,, \qquad (4)$$

where $P_\phi(y_t)$ is the soft target estimated by the finetuned BERT with learned parameters ϕ, and $\mathcal{V}$ denotes the output vocabulary. Note that ϕ is fixed during the distillation process. An illustration of this learning process is provided in Figure 1, which aims to match the word probability distribution $P_\theta(y_t)$ provided by the student with $P_\phi(y_t)$ provided by the teacher (*i.e.*, distillation).

To further improve the Seq2Seq student model, hard-assigned labels are also utilized. The final

[2] Besides MLM, Devlin et al. (2019) also introduced the next sentence prediction task for training BERT. We omit this task since it is unrelated to our work.

[3] The two sequences are consecutive paragraphs sampled from a very large corpus such as Wikipedia.

model is trained with the following compound objective:

$$\mathcal{L}(\theta) = \alpha \mathcal{L}_{bidi}(\theta) + (1-\alpha)\mathcal{L}_{xe}(\theta)\,, \quad (5)$$

where α is a hyper-parameter for tuning the relative importance of the two training targets: soft estimation from finetuned BERT, and ground-truth hard label. Note that our proposed approach only has a minimal requirement on the architecture of the incorporated Seq2Seq model. As long as the model is trained to estimate word-level probability as in Eqn. (1), it can be trained jointly with the proposed objective function Eqn. (5).

At a higher level, the additional loss term $\mathcal{L}_{bidi}$ can be interpreted as a sequence-level objective function. Our auto-regressive (or causal) model θ tries to predict the probability distribution that matches the estimation the bidirectional teacher model predicts, hence encouraging the planning of future (right context) for generation.

4 Experiments

In this section, we describe our experiments on two well-studied text generation tasks: machine translation, and abstractive text summarization.

4.1 Datasets

Machine Translation We consider two relatively small-scale datasets, IWSLT15 English-Vietnamese (En-Vi, 113k training samples) and IWSLT14 German-English (De-En, 160k training samples), and one medium-scale dataset, WMT14 English-German (En-De, 4.5M training samples). For IWSLT15 En-Vi, we use the pre-processed dataset provided by Luong and Manning (2015). We use tst2012 as dev set and test on tst2013. For IWSLT14 De-En, we follow the pre-processing steps and the same train/dev/test split as in Wu et al. (2019). For WMT14 En-De, we follow the pre-processing steps in Vaswani et al. (2017) for fair comparison. We use newstest2013 as the dev set and newstest2014 as the test set. We report BLEU scores (Papineni et al., 2002) for evaluation of MT performance following the Moses script.[4]

Abstractive Summarization For summarization, we conduct experiments on the Gigaword summarization dataset (Rush et al., 2015). Note that the original train/valid/test split of Gigaword is 3.8M/190k/2k. In our experiments, we observed severe distribution mismatch between the validation and test data. See Table 4, 5, and Sec. 4.4 for detailed discussion. Therefore, we further sampled 5k/5k dev/test-dev splits from the validation set and tuned hyper-parameters on the dev set only. We report ROUGE scores (Lin, 2004) on test-dev for the evaluation of our proposed approach, and include results on the standard test split for the comparison with prior work.

4.2 Implementation Details

Our implementation is based on the PyTorch (Paszke et al., 2017) version of OpenNMT (Klein et al., 2018) seq2seq toolkit. We use the 'base' model of 6-layer Transformer with 512-hidden 8-head attention blocks and 2048-hidden feed-forward layer for all experiments, with label smoothing regularization (LSR) (Szegedy et al., 2016) of 0.1.[5] We batch examples with similar sequence length, and count batch size by the number of tokens. For MT we use the pre-trained *BERT-base-multilingual-cased* model, and for summarization we use *BERT-base-uncased* as the starting point of BERT finetuning.[6] We use the corresponding pre-trained byte-pair-encoding (Sennrich et al., 2016) shipped together with the BERT model for tokenization.

For all training methods of all Transformer models, the learning rate schedule is set to $lr = \eta \cdot d_{model}^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup_steps^{-1.5})$, where $d_{model} = 512$ is the attention representation size (Vaswani et al., 2017). For all BERT finetuning, we follow Devlin et al. (2019) and use a triangular learning rate schedule with maximum learning rate η. The parameters are updated with the Adam optimizer (Kingma and Ba, 2015). In the distillation stage, we pre-compute BERT's prediction logits of the training data[7] and use top-K distillation (Tan et al., 2019) to reduce computation overhead and memory footprint, where K is set to 8 across all the experiments.[8]

[4] For fair comparison to previous work, we report tokenized BLEU scores using https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl, and for WMT14 En-De, we further split the compound words after tokenization.

[5] Our method can also be viewed as a 'learned LSR'. The results reported of our proposed method are trained together with regular LSR, showing the effectiveness of our teacher.

[6] BERT pre-trained models are available at https://github.com/google-research/bert. Our finetuning implementation is modified from code available at https://github.com/huggingface/pytorch-pretrained-BERT.

[7] The masking strategy is described in the supplementary.

[8] We also tune the temperature T for the $softmax$ applied at the teacher's logits. Different from the original KD, we

De-En Models	dev	test
Our Implementations		
Transformer (base)	35.27	34.09
+ BERT teacher	**36.93**	**35.63**
Other Reported Results		
ConvS2S + MRT‡	33.91	32.85
Transformer (big)$^{\diamond}$	-	34.4†
Lightweight Conv$^{\diamond}$	-	34.8†
Dyn. Convolution$^{\diamond}$	-	35.2†

Table 1: BLEU scores for IWSLT14 German-English translation. (†) tuned with checkpoint averaging. (‡) from Edunov et al. (2018). (⋄) from Wu et al. (2019).

En-Vi Models	tst2012	tst2013
Our Implementations		
RNN	23.37	26.80
+ BERT teacher	25.14	27.59
Transformer (base)	27.03	30.76
+ BERT teacher	**27.85**	**31.51**
Other Reported Results		
RNN†	-	26.1
Seq2Seq-OT*	24.5	26.9
ELMo$^{\diamond}$	-	29.3
CVT$^{\diamond}$	-	29.6

Table 2: BLEU scores for IWSLT15 English-Vietnamese translation. (†) from Luong et al. (2017). (⋆) from Chen et al. (2019). (⋄) from Clark et al. (2018).

For the detailed values of the hyper-parameters for each experiment, please refer to the supplementary material. We found it necessary to train longer with $\mathcal{L}_{bidi}$, since it is still improving after the step at which the baseline Transformer starts to plateau. At inference time, we use beam search with beam size 4 and length penalty (Wu et al., 2016) of 0.6 across all the models. All the hyper-parameters are tuned on the development set. Note that our Transformer baselines achieve higher scores than the reference implementation on each dataset (in most cases comparable to the state-of-the-art).

4.3 Results on Machine Translation

We first validate our proposed text generation approach on machine translation task. Experimental results are summarized in Table 1, 2 and 3, which show that our model significantly improves over the strong Transformer baseline across all three datasets. Note that our baseline is the 'base' model of Transformer, which has 44M trainable parameters, and the reference implementation by Wu et al. (2019) of the 'big' model with 176M parameters.[9]

For IWSLT German-English translation, our method improves over the Transformer baseline by 1.54 BLEU points, and achieves new state of the art. Our approach outperforms previously-reported results such as ConvS2S+MRT, a convolutional-based model (Gehring et al., 2017) with minimum risk training (Edunov et al., 2018), and Lightweight and Dynamic Convolution (Wu et al., 2019). Note that Wu et al. (2019) also tuned checkpoint averaging, which creates a soft ensemble effect. And their model has roughly the same amount of parameters as Transformer (big).

For IWSLT English-Vietnamese translation, since most prior work experimented with RNN models, we also report RNN-based results here. This also suggests that our method is model-agnostic. Our best model outperforms Seq2Seq-OT (Chen et al., 2019) that utilizes optimal transport for sequence-level training, as well as the ELMo and CVT results reported in Clark et al. (2018).[10] For WMT14 English-German translation, our method still improves over the well-tuned Transformer baseline. We also report the scores of Transformer (big) and state-of-the-art Dynamic Convolution model (Wu et al., 2019) for reference.

En-De Models	NT2013	NT2014
Our Implementations		
Transformer (base)	25.95	26.94
+ BERT teacher	**26.22**	**27.53**
Other Reported Results		
Transformer (base)$^{\diamond}$	25.8	27.3†
Transformer (big)*‡	26.5	29.3†
Dyn. Convolution$^{\bullet\ddagger}$	**26.9**$\pm$0.2	**29.7**†

Table 3: BLEU scores for WMT14 English-German translation. (†) tuned with checkpoint averaging. (‡) trained on WMT16, a slightly different version of training data. (⋄) from Vaswani et al. (2017). (⋆) from Ott et al. (2018). (•) from Wu et al. (2019).

4.4 Results on Abstractive Summarization

Table 4 and Table 5 show the results of our approach on abstractive summarization task, where

do not apply the same T on the student. In preliminary experiment we found high T of Seq2Seq results in much worse performance. We hypothesize the low-entropy nature of conditioned text generation is not suitable for temperature scaling.

[9] Parameter counts exclude word embedding and final linear projection, which mostly depends on the vocabulary size. BERT-base has 86M trainable parameters.

[10] The CVT results used a much larger RNN and CNN-based character embedding, as well as a customized structure. Therefore, we did not try to use RNN to match their results.

GW Models	R-1	R-2	R-L
Dev			
Transformer (base)	46.64	24.37	43.17
+ BERT teacher	**47.35**	**25.11**	**44.04**
Test-Dev			
Transformer (base)	46.84	24.80	43.58
+ BERT teacher	**47.90**	**25.75**	**44.53**

Table 4: ROUGE F_1 scores for Gigaword abstractive summarization on our internal test-dev split.

GW Models	R-1	R-2	R-L
Seq2Seq†	36.40	17.77	33.71
CGU‡	36.3	18.0	33.8
FTSum$_g{}^\star$	37.27	17.65	34.24
E2T$_{cnn}{}^\diamond$	37.04	16.66	**34.93**
Re^{3}Sum$^\bullet$	37.04	**19.03**	34.46
Trm + BERT teacher	**37.57**	18.59	34.82

Table 5: ROUGE F_1 scores for Gigaword abstractive summarization on the official test set (Trm: Transformer). (†) from Nallapati et al. (2016). (‡) from Lin et al. (2018). (⋆) from Cao et al. (2018b). (⋄) from Amplayo et al. (2018). (•) from Cao et al. (2018a).

R-1, R-2, and R-L denote F_1 scores of ROUGE-1, ROUGE-2, and ROUGE-L, respectively. Our method shows improvement on all the metrics, as shown in Table 4. We observe a large gap between dev and test scores, which suggests that the data in the test set is very different from that in the validation set, as mentioned in Section 4.1. Given the fact that the official test split contains only 1,951 noisy examples,[11] we believe that our results on the dev/test-dev sets further strengthens our claim.

On the test split, our best model is comparable to state-of-the-art models that use much more complex architectures specifically designed for summarization. CGU (Lin et al., 2018) augmented convolutional gating units. FTSum$_g$ (Cao et al., 2018b) leveraged extra information extraction and dependency parsing features. E2T$_{cnn}$ (Amplayo et al., 2018) utilized entities provided by an external entity linking system. Re^{3}Sum (Cao et al., 2018a) carefully designed a retrieve-and-rerank pipeline with human-written soft templates. Despite that our model has no summarization-specific model design, we still achieve comparable performance to these models on all the metrics.

[11] When we manually inspected the test set data, we found many corrupted examples such as extremely short input articles, meaningless summary, and dominating unknown words.

Methods	De-En (dev)	En-Vi (tst2012)
Transformer (base)	35.27	27.03
Trm + BERT$_{l2r}$	35.20	26.99
Trm + BERT$_{sm}$	36.32	27.68
Trm + BERT	**36.93**	**27.85**

Table 6: Ablation study. (Trm: Transformer)

4.5 Ablation Study

There are several possible factors that could contribute to the performance gain: additional parameters of BERT, extra data (pretraining corpus) of BERT, and the bidirectional nature. To better understand the key contributions of our method, we conduct an ablation study described in the following. We finetune 2 extra teachers: BERT$_{sm}$ and BERT$_{l2r}$. For BERT$_{sm}$, we use a smaller BERT (6 layers) for C-MLM finetuning, which has approximately the same number of parameters as Transformer-base.[12] For BERT$_{l2r}$, we use the full BERT model but finetune it using left-to-right LM as in the conventional Seq2Seq model. Next, we apply the proposed KD method to train the Transformer on En-Vi and De-En MT tasks. Results are shown in Table 6. BERT$_{sm}$ still works well though the full BERT provides further improvement. On the other hand, BERT$_{l2r}$ slightly hurts the performance. We hypothesize that it generates noisy learning targets for the student, hence the performance drop. Empirically, we show that the bidirectional knowledge could be more important than the extra parameters, while the pre-trained weights remain useful for more stable C-MLM training.

4.6 Generation for Different Lengths

We next analyze the effect of our proposed approach on different output lengths. We plot the BLEU scores on MT w.r.t. different output generation lengths N on the development set.[13] Results are provided in Figure 2 and Figure 3. For IWSLT German-English dataset (Figure 2: Left), we can see a shared trend that the proposed $\mathcal{L}_{bidi}$ objective gains higher BLEU points on longer translation pairs. For WMT English-German (Figure 3), we can see that although the proposed method performs much worse when the output sentences

[12] We still use the pretrained weights of BERT, otherwise the C-MLM does not converge very well.

[13] For Gigaword summarization, almost all summaries are short sentences (less than 0.5% of the summaries contain more than 16 words), so we omit the analysis.

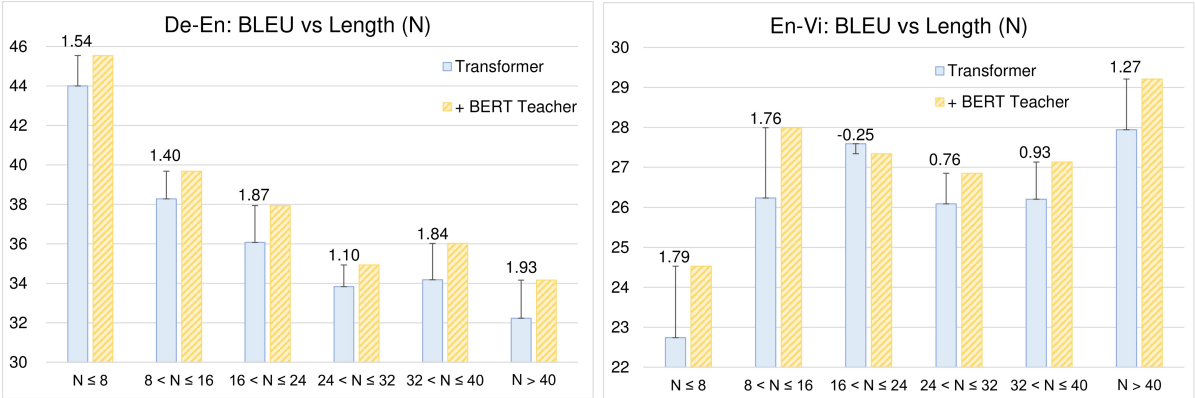

Figure 2: BLEU scores on IWSLT German-English and English-Vietnamese for different output lengths.

Reference	my mother says that i started reading at the age of two , although i think four is probably close to the truth .
Transformer	my mother says that i started reading with two years , but i think that four of them probably correspond to the truth . (39.6)
Ours	my mother says that i started reading at the age of two , but i think four is more likely to be the truth . (65.2)
Reference	we already have the data showing that it reduces the duration of your flu by a few hours .
Transformer	we 've already got the data showing that it 's going to crash the duration of your flu by a few hours . (56.6)
Ours	we already have the data showing that it reduces the duration of your flu by a few hours . (100.0)
Reference	we now know that at gombe alone , there are nine different ways in which chimpanzees use different objects for different purposes .
Transformer	we know today that alone in gombe , there are nine different ways that chimpanzees use different objects in different ways . (35.8)
Ours	we now know that in gombe alone , there are nine different ways that chimpanzees use different objects for different purposes . (71.5)

Table 7: Qualitative examples from IWSLT German-English translation. Numbers inside the parenthesis are sentence-level BLEU scores. Red word is where the baseline Transformer makes a mistake without considering the possible future phrase and fails to recover. On the other hand, our model makes the right decision at the blue word, hence generates more coherent sentence. Please refer to Section 4.7 for detailed explanation.

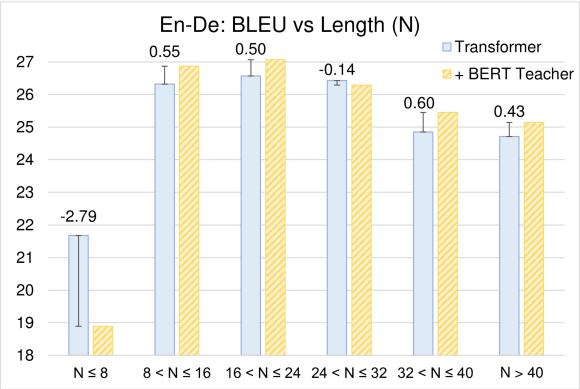

Figure 3: BLEU scores on WMT English-German for different output lengths.

are very short, it achieves relatively consistent improvement on longer cases, hence resulting in overall BLEU improvement. For IWSLT English-Vietnamese (Figure 2: Right), we see a similar trend when the length $N > 24$.

4.7 Qualitative Examples

In Table 7, we show some translation examples on IWSLT German-English dataset. In the first example, the baseline Transformer cannot recover from '*with*' and '*of*', which renders the full sentence not making much sense. "I started reading *with*..." would make sense from the left context; however, if the model also considers the right context "the age of two", the word '*with*' would be assigned with lower probability by the soft labels provided by the BERT teacher. Even though at test-time the model cannot 'look ahead', the soft-targets at training-time prevents the over-confidence of the model on one-hot label; hence the better generalization at the test-time. Similarly, other examples show that our model can generate text more coherently w.r.t. the context on the right (underlined in Table 7), thus making more accurate and natural translation.

5 Conclusion

In this work, we propose a novel and generic approach to utilizing pre-trained language models to

improve text generation *without explicit parameter sharing, feature extraction, or augmenting with auxiliary tasks.* Our proposed Conditional MLM mechanism leverages unsupervised language models pre-trained on large corpus, and then adapts to supervised sequence-to-sequence tasks. Our distillation approach *indirectly* influences the text generation model by providing soft-label distributions only, hence is *model-agnostic*. Experiments show that our model improves over strong Transformer baselines on multiple text generation tasks such as machine translation and abstractive summarization, and achieves new state-of-the-art on some of the translation tasks. For future work, we will explore the extension of Conditional MLM to multimodal input such as image captioning.

References

Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. 2018. Entity commonsense representation for neural abstractive summarization. In *NAACL*.

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

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018a. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018b. Faithful to the original: Fact aware neural abstractive summarization. In *AAAI*.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *ICLR*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *ACL*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *EMNLP*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *NAACL*.

Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

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

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Constant-time machine translation with conditional masked language models. *arXiv preprint arXiv:1904.09324*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

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

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *AMTA*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL Text Summarization Branches Out Workshop*.

Junyang Lin, Xu Sun, Shuming Ma, and Qi Su. 2018. Global encoding for abstractive summarization. In *ACL*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *NAACL*.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *IWSLT*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *NIPS*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Chris Pal, and Yoshua Bengio. 2018. Twin networks: Matching the future for sequence generation. In *ICLR*.

Kaitao Song Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR*.

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 *NIPS*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *ICLR*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. 2019. Towards making the most of bert in neural machine translation. *arXiv preprint arXiv:1908.05672*.

Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019a. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2019b. Regularizing neural machine translation by target-bidirectional agreement. In *AAAI*.

A Implementaion Details and Hyper-parameter Values

We run all experiments on single GPU of NVIDIA Titan RTX or V100 except for WMT En-De we use 4 V100s for training. Note that for large batch sizes that do not fit in GPU memory, we use the gradient accumulation tricks as in Ott et al. (2018). Batch sizes are counted in number of tokens. Note that all the hyper-parameters are tuned on the development set only.

To compute the logits (soft labels) from teacher, we repeat a training pair for 7 times and create a circular mask as illustrated in Figure 4. This mask approximates the 15% masking rate of the BERT training. From the masked positions we can obtain soft probabilities predicted by the BERT teacher for each output tokens y. These logits are pre-computed once for the training set so that we do not have to repeatedly sample random masks and run forward pass of BERT while training.

IWSLT De-En For C-MLM fine-tuning, we train for 100k steps with 5k $warmup_steps$, $\eta = 5 \cdot 10^{-5}$, and batch size of 16k tokens. For baseline model, we train for 50k steps with 4k $warmup_steps$ and batch size of 6k tokens. The learning rate η is set to 1. For the proposed model, we train for 100k steps with 8k $warmup_steps$ and batch size of 6k tokens. The learning rate η is set to 2, $\alpha = 0.5$, and $T = 10$. Seq2Seq model uses dropout (Srivastava et al., 2014) of 0.3 in both cases.

IWSLT En-Vi For C-MLM fine-tuning and baseline Transformer, the hyper-parameters are identical to that of IWSLT De-En. For the proposed model, we train for 100k steps with 8k $warmup_steps$ and batch size of 6k tokens. The learning rate η is set to 2, $\alpha = 0.1$, and $T = 5$. Dropout is still 0.1.

WMT En-De For C-MLM fine-tuning, we train for 100k steps with 5k $warmup_steps$, $\eta = 5 \cdot 10^{-5}$, and batch size of 512k tokens. For baseline model, we train for 30k steps with 4k $warmup_steps$ and batch size of 384k tokens. The learning rate η is set to 4. Since this is our largest dataset and training is slow, for the proposed model we use the baseline Transformer to initialize the Seq2Seq student. For the proposed model, we continue training for 50k steps with 4k $warmup_steps$ and batch size of 64k tokens. The learning rate η is set to 2, $\alpha = 0.1$, and $T = 5$. Seq2Seq model uses dropout of 0.1 in both cases.

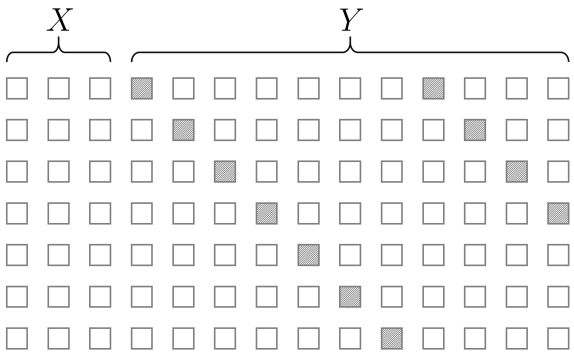

Figure 4: Illustration of the masking strategy for computing the teacher soft labels. Gray slashed boxes denote the `[MASK]` positions.

Gigaword For C-MLM fine-tuning, we train for 100k steps with 5k $warmup_steps$, $\eta = 5 \cdot 10^{-5}$, and batch size of 64k tokens. For baseline model, we train for 50k steps with 4k $warmup_steps$ and batch size of 40k tokens. The learning rate η is set to 1. For the proposed model, we train for 70k steps with 4k $warmup_steps$ and batch size of 36k tokens. The learning rate η is set to 2, $\alpha = 0.1$, and $T = 10$. Seq2Seq model uses dropout of 0.1 in both cases.

B Additional Generation Examples

We show Gigaword summarization examples in Table 9 and extra En-DE generation examples in Table 8. Qualitatively, our Transformer + BERT Teacher outperforms baseline Transformer and generate more coherent sentences.

Reference	the political climate in the u.s. at the time was tense , and there were debates going on about immigration .
Transformer	the political climate in the u.s. was back then , and there was constant disasters . (29.5)
Ours	the political climate in the united states at the time was tense , and there were ongoing shifting debates . (57.3)
Reference	it would be immoral to leave these young people with a climate system spiraling out of control .
Transformer	it would be immoral to let these young people leave a climate system that was out of control . (44.6)
Ours	it would be immoral to leave these young people with a climate system out of control . (84.3)
Reference	the tahltan have called for the creation of a tribal heritage reserve which will set aside the largest protected area in british columbia .
Transformer	tahltan demands the institution of a tribe in british columbia that should make the largest protection area in british columbia . (19.9)
Ours	the tahltan demands to build a tribe reserve that should be the largest protected area in british columbia . (32.2)

Table 8: Qualitative examples from IWSLT German-English translation. Numbers inside the parenthesis are sentence-level BLEU scores. Red word is where the baseline Transformer makes a mistake without considering the possible future phrase and fails to recover. On the other hand, our model makes the right decision at the blue word, hence generates more coherent sentence. Please refer to Section 4.6 in the main paper for detailed explanation.

Reference	china offers tax exemptions for laid-off workers
Transformer	china encourages laid-off workers to seek employment
Ours	china offers tax exemptions to laid-off workers
Reference	swiss police arrest britons who allegedly ran rental car racket
Transformer	three britons arrested in swiss luxury hotel
Ours	swiss police arrest three britons in rental car racket case
Reference	south korea stocks extend declines as kia concerns intensify
Transformer	south korean stocks fall for #th time in # days ; kia leads
Ours	south korean stocks fall as kia troubles intensify

Table 9: Qualitative examples from the Gigaword summarization dataset. Baseline model suffers from early mistakes. Our model generates more coherent summaries.