Transfer Learning for Sequence Generation: from Single-source to Multi-source

Xuancheng Huang[1], **Jingfang Xu**[4], **Maosong Sun**[1,3], **and Yang Liu**[1,2,3*]
[1]Dept. of Comp. Sci. & Tech., BNRist Center, Institute for AI, Tsinghua University
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Beijing Academy of Artificial Intelligence
[4]Sogou Inc., Beijing, China

Abstract

Multi-source sequence generation (MSG) is an important kind of sequence generation tasks that takes multiple sources, including automatic post-editing, multi-source translation, multi-document summarization, etc. As MSG tasks suffer from the data scarcity problem and recent pretrained models have been proven to be effective for low-resource downstream tasks, transferring pretrained sequence-to-sequence models to MSG tasks is essential. Although directly finetuning pretrained models on MSG tasks and concatenating multiple sources into a single long sequence is regarded as a simple method to transfer pretrained models to MSG tasks, we conjecture that the direct finetuning method leads to catastrophic forgetting and solely relying on pretrained self-attention layers to capture cross-source information is not sufficient. Therefore, we propose a two-stage finetuning method to alleviate the pretrain-finetune discrepancy and introduce a novel MSG model with a fine encoder to learn better representations in MSG tasks. Experiments show that our approach achieves new state-of-the-art results on the WMT17 APE task and multi-source translation task using the WMT14 test set. When adapted to document-level translation, our framework outperforms strong baselines significantly.[1]

1 Introduction

Thanks to the continuous representations widely used across text, speech, and image, neural networks that accept multiple sources as input have gained increasing attention in the community (Ive et al., 2019; Dupont and Luettin, 2000). For example, multi-modal inputs that are complementary have proven to be helpful for many sequence generation tasks such as question answering (Antol et al., 2015), machine translation (Huang et al., 2016), and speech recognition (Dupont and Luettin, 2000). In natural language processing, multiple textual inputs have also been shown to be valuable for sequence generation tasks such as multi-source translation (Zoph and Knight, 2016), automatic post-editing (Chatterjee et al., 2017), multi-document summarization (Haghighi and Vanderwende, 2009), system combination for NMT (Huang et al., 2020), and document-level machine translation (Wang et al., 2017). We refer to this kind of tasks as *multi-source sequence generation* (MSG).

Pretraining	Single-source SG	Multi-source SG
AutoEncoding (e.g., BERT)	BERT-FUSED (Zhu et al., 2019)	DUALBERT (Correia and Martins, 2019)
Seq2Seq (e.g., BART)	MBART-TRANS (Liu et al., 2020)	*this work*

Table 1: Comparison of various approaches to transferring pretrained models to single-source and multi-source sequence generation tasks. Different from prior studies, this work aims at transferring pretrained sequence-to-sequence models to multi-source sequence generation tasks.

Unfortunately, MSG tasks face a severe challenge: there are no sufficient data to train MSG models. For example, multi-source translation requires parallel corpora involving multiple languages, which are usually restricted in quantity and coverage. Recently, as pretraining language models that take advantage of massive unlabeled data have proven to improve natural language understanding (NLU) and generation tasks substantially (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020), a number of researchers have proposed to leverage pretrained language models to enhance MSG tasks (Correia and Martins, 2019; Lee et al., 2020; Lee, 2020). For example, Correia and Martins (2019) show that pretrained autoencoding (AE) models

*Corresponding author: Yang Liu
[1]The source code is available at https://github.com/THUNLP-MT/TRICE

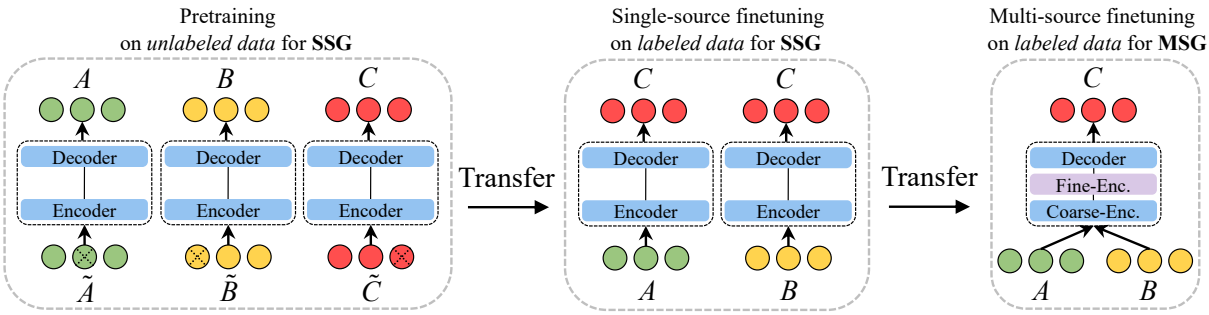

Figure 1: Overview of our framework. "A", "B", and "C" denote sentences in different languages. After being pretrained on unlabeled data, the single-source sequence generation (SSG) model is finetuned on single-source labeled data. Then, the SSG model is extended to the MSG model by adding a fine encoder upon the pretrained encoder (i.e., the coarse encoder). Finally, the MSG model is finetuned on the multi-source data. The proposed framework aims to reduce the pretrain-finetune discrepancy and learn better multi-source representations.

like BERT (Devlin et al., 2019) can improve automatic post-editing.

As most recent pretrained sequence-to-sequence (Seq2Seq) models (Song et al., 2019; Lewis et al., 2020; Liu et al., 2020) have demonstrated their effectiveness in improving single-source sequence generation (SSG) tasks, we believe that pretrained Seq2Seq models can potentially bring more benefits to MSG than pretrained AE models. Although it is easy to transfer Seq2Seq models to SSG tasks, transferring them to MSG tasks is challenging because MSG takes multiple sources as the input, leading to severe *pretrain-finetune discrepancies* in terms of both architectures and objectives.

A straightforward solution is to concatenate the representations of multiple sources as suggested by Correia and Martins (2019). However, we believe this approach suffers from two major drawbacks. First, due to the discrepancy between pretraining and MSG, directly transferring pretrained models to MSG tasks might lead to catastrophic forgetting (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017) that results in reduced performance. Second, the pretrained self-attention layers might not fully learn the representations of the concatenation of multiple sources because they do not make full use of the cross-source information.

Inspired by adding intermediate tasks for NLU (Pruksachatkun et al., 2020; Vu et al., 2020), we conjecture that inserting a proper intermediate task between them can alleviate the discrepancy. In this paper, we propose a two-stage finetuning method named *gradual finetuning*. Different from prior studies, our work aims to transfer pretrained Seq2Seq models to MSG (see Table 1). Our approach first transfers from pretrained models to SSG and then transfers from SSG to MSG (see Figure 1). Furthermore, we propose a novel MSG model with coarse and fine encoders to differentiate sources and learn better representations. On top of a coarse encoder (i.e., the pretrained encoder), a fine encoder equipped with cross-attention layers (Vaswani et al., 2017) is added. We refer to our approach as TRICE (a task-agnostic Transferring fRamework for multI-sourCe sEquence generation), which achieves new state-of-the-art results on the WMT17 APE task and the multi-source translation task using the WMT14 test set. When adapted to document-level translation, our framework outperforms strong baselines significantly.

2 Approach

Figure 1 shows an overview of our framework. First, the problem statement is described in Section 2.1. Second, we propose to use the gradual finetuning method (Section 2.2) to reduce the pretrain-finetune discrepancy. Third, we introduce our MSG model, which consists of the coarse encoder (Section 2.3), the fine encoder (Section 2.4), and the decoder (Section 2.5).

2.1 Problem Statement

As shown in Figure 1, there are three kinds of dataset: (1) the unlabeled multilingual dataset $\mathcal{D}_p$ containing monolingual corpora in various languages, (2) the single-source parallel dataset $\mathcal{D}_s$ involving multiple language pairs, and (3) the multi-source parallel dataset $\mathcal{D}_m$. The general objective is to leverage these three kinds of dataset to improve multi-source sequence generation tasks.

Formally, let $\mathbf{x}_{1:K} = \mathbf{x}_1 \dots \mathbf{x}_K$ be K source sentences, where $\mathbf{x}_k$ is the k-th sentence. We use $x_{k,i}$

to denote the i-th word in the k-th source sentence and $\mathbf{y} = y_1 \dots y_J$ to denote the target sentence with J words. The MSG model is given by

$$P_m(\mathbf{y}|\mathbf{x}_{1:K};\boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{x}_{1:K}, \mathbf{y}_{<j};\boldsymbol{\theta}), \quad (1)$$

where y_j is the j-th word in the target, $\mathbf{y}_{<j} = y_1 \dots y_{j-1}$ is a partial target sentence, $P(y_j|\mathbf{x}_{1:K}, \mathbf{y}_{<j};\boldsymbol{\theta})$ is a word-level generation probability, and $\boldsymbol{\theta}$ are the parameters of the MSG model.

2.2 Gradual Finetuning

As training neural models on large-scale unlabeled datasets is time-consuming, it is a common practice to utilize pretrained models to improve downstream tasks by using transfer learning methods (Devlin et al., 2019). As a result, we focus on leveraging single-source and multi-source parallel datasets to transfer pretrained Seq2Seq models to MSG tasks.

Curriculum learning (Bengio et al., 2009) aims to learn from examples organized in an easy-to-hard order, and intermediate tasks (Pruksachatkun et al., 2020; Vu et al., 2020) are introduced to alleviate the pretrain-finetune discrepancy for NLU. Inspired by these studies, we expect that changing the training objective from pretraining to MSG gradually can reduce the difficulty of transferring pretrained models to MSG tasks. Therefore, we propose a two-stage finetuning method named gradual finetuning. The transferring process is divided into two stages (see Figure 1). In the first stage, the SSG model is transferred from denoising auto-encoding to the single-source sequence generation task, and the model architecture is kept unchanged. In the second stage, an additional fine encoder (see Section 2.4) is introduced to transform the SSG model to the MSG model, and the MSG model is optimized on the multi-source parallel corpus.

Formally, we use $\boldsymbol{\phi}_p$ to denote the parameters of the SSG model. Without loss of generality, the pretraining process can be described as follows:

$$\mathcal{L}_p(\boldsymbol{\phi}_p) = \frac{1}{|\mathcal{D}_p|} \sum_{\mathbf{z}\in\mathcal{D}_p} \Big(-\log P_s(\mathbf{z}|\tilde{\mathbf{z}};\boldsymbol{\phi}_p)\Big), \quad (2)$$

$$\hat{\boldsymbol{\phi}}_p = \operatorname*{argmin}_{\boldsymbol{\phi}_p} \Big\{\mathcal{L}_p(\boldsymbol{\phi}_p)\Big\}, \quad (3)$$

where $\mathbf{z}$ is a sentence that could be in many languages, $\tilde{\mathbf{z}}$ is the corrupted sentence obtained from $\mathbf{z}$, P_s is the probability modeled by the SSG model, and $\hat{\boldsymbol{\phi}}_p$ are the learned parameters. In this way, a powerful multilingual model is obtained by pretraining on the unlabeled multilingual dataset $\mathcal{D}_p$.

Then, in the first finetuning stage, let $\boldsymbol{\phi}_s$ be the parameters of the SSG model, which are initialized by $\hat{\boldsymbol{\phi}}_p$. As the single-source parallel dataset $\mathcal{D}_s$ is not always available, we can build it from the K-source parallel dataset $\mathcal{D}_m$. Assume $\langle \mathbf{x}_{1:K}, \mathbf{y}\rangle$ is a training example in $\mathcal{D}_m$, a training example $\langle \mathbf{x}, \mathbf{y}\rangle$ in $\mathcal{D}_s$ can be constructed by sampling one source from each K-source training example with a probability of $1/K$. The first finetuning process is given by

$$\mathcal{L}_s(\boldsymbol{\phi}_s) = \frac{1}{|\mathcal{D}_s|} \sum_{\langle \mathbf{x},\mathbf{y}\rangle\in\mathcal{D}_s} \Big(-\log P_s(\mathbf{y}|\mathbf{x};\boldsymbol{\phi}_s)\Big), \quad (4)$$

$$\hat{\boldsymbol{\phi}}_s = \operatorname*{argmin}_{\boldsymbol{\phi}_s} \Big\{\mathcal{L}_s(\boldsymbol{\phi}_s)\Big\}, \quad (5)$$

where $\hat{\boldsymbol{\phi}}_s$ are the learned parameters. The learned SSG model is capable of taking inputs in multiple languages.

In the second finetuning stage, $\boldsymbol{\phi}_m$, the parameters of the coarse encoder, the decoder, and the embeddings, are initialized by $\hat{\boldsymbol{\phi}}_s$ while $\boldsymbol{\gamma}$ are the randomly initialized parameters of the fine encoder. Thus, $\boldsymbol{\theta} = \boldsymbol{\phi}_m \cup \boldsymbol{\gamma}$ are the parameters of the MSG model. The second finetuning process can be described as

$$\begin{aligned}&\mathcal{L}_m(\boldsymbol{\theta})\\ =&\frac{1}{|\mathcal{D}_m|} \sum_{\langle \mathbf{x}_{1:K},\mathbf{y}\rangle\in\mathcal{D}_m} \Big(-\log P_m(\mathbf{y}|\mathbf{x}_{1:K};\boldsymbol{\theta})\Big),\end{aligned} \quad (6)$$

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \Big\{\mathcal{L}_m(\boldsymbol{\theta})\Big\}, \quad (7)$$

where P_m is given by Eq. (1). As a result, the model is expected to learn from abundant unlabeled data and perform well on the MSG task. In the following subsections, we will describe the MSG model architecture (see Figure 2) applied in the second finetuning stage.

2.3 Input Representation and the Coarse Encoder

In general, pretrained encoders are considered as strong feature extractors to learn meaningful representations (Zhu et al., 2019). For this reason, Correia and Martins (2019) propose to use the pretrained multilingual encoder to encode the bilingual input pair of APE. Since MSG tasks usually

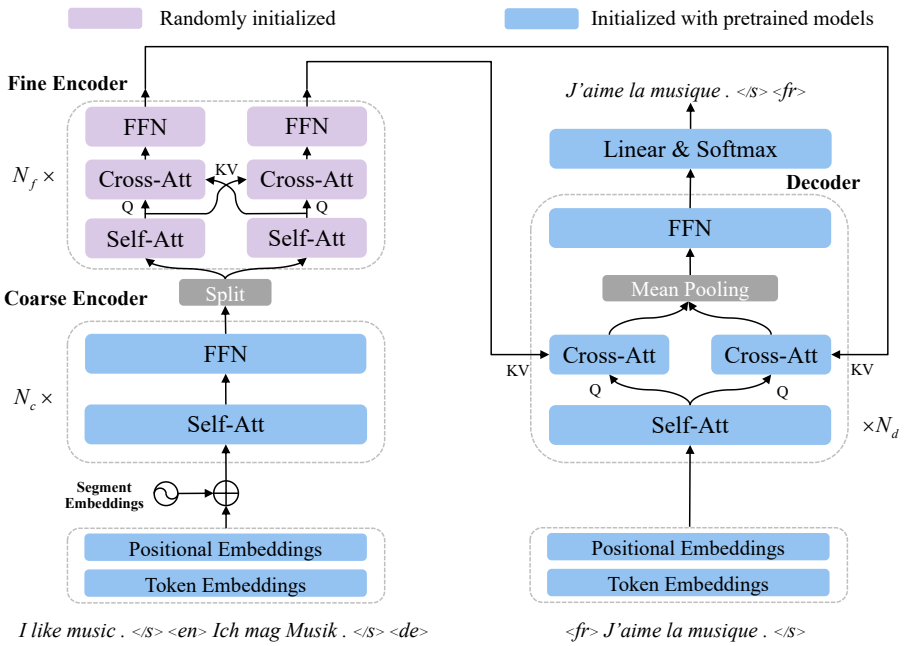

Figure 2: The architecture of our framework. Multiple sources are first concatenated and encoded by the coarse encoder and then encoded by the fine encoder to capture fine-grained cross-source information. Finally, the representations are utilized by the decoder to generate the target sentence. For simplicity, this figure only illustrates the situation that the input contains two sources ($K = 2$).

have multiple sources involving different languages and pretrained multilingual Seq2Seq models like mBART (Liu et al., 2020) usually rely on special tokens (e.g., <en>) to differentiate languages, concatenating multiple sources into a single long sentence will make the model confused about the language of the concatenated sentence (see Table 6). Therefore, we propose to add additional segment embedding to differentiate sentences in different languages and encode source sentences jointly by a single pretrained multilingual encoder.

Formally, the input representation can be denoted by

$$\mathbf{X}_{k,i} = \mathbf{E}^{\text{tok}}[x_{k,i}] + \mathbf{E}^{\text{pos}}[i] + \mathbf{E}^{\text{seg}}[k], \quad (8)$$

where $\mathbf{X}_{k,i}$ is the input representation of the i-th word in the k-th source sentence, and $\mathbf{E}^{\text{tok}}$, $\mathbf{E}^{\text{pos}}$, and $\mathbf{E}^{\text{seg}}$ are the token, position, and segment/language embedding matrices, respectively. $\mathbf{E}^{\text{tok}}$ and $\mathbf{E}^{\text{pos}}$ are initialized by pretrained embedding matrices. $\mathbf{E}^{\text{seg}}$ is implemented as constant sinusoidal embeddings (Vaswani et al., 2017), which is denoted by $\mathbf{E}^{\text{seg}}[k]_{2i} = \sin(1000{*}k/10000^{2i/d})$, where $\mathbf{E}^{\text{seg}}[k]_{2i+1}$ is similar to $\mathbf{E}^{\text{seg}}[k]_{2i}$ and i is the dimension index while d is model dimension.[2]

[2]If the pretrained model already contains the segment/language embedding matrix, then the pretrained one is used.

Then, the pretrained encoder is utilized to encode multiple sources:

$$\mathbf{R}^{(i)}_{1:K} = \text{FFN}\left(\text{SelfAtt}\left(\mathbf{R}^{(i-1)}_{1:K}\right)\right), \quad (9)$$

where $\text{SelfAtt}(\cdot)$ and $\text{FFN}(\cdot)$ are the self-attention and feed-forward networks, respectively. $\mathbf{R}^{(i)}_{1:K}$ is the representation output by the i-th encoder layer, and $\mathbf{R}^{(0)}_{1:K}$ refers to $\mathbf{X}_1 \ldots \mathbf{X}_K$, where $\mathbf{X}_k$ is equivalent to $\mathbf{X}_{k,1} \ldots \mathbf{X}_{k,I_k}$ and I_k is the number of tokens in the k-th source sentence.

However, we conjecture that indiscriminately modeling dependencies between words by the pretrained self-attention layers cannot capture cross-source information adequately. To this end, we regard the pretrained encoder as the coarse encoder and introduce a novel fine encoder to learn better multi-source representations.

2.4 The Fine Encoder

To alleviate the pretrain-finetune discrepancy, we adopt the gradual finetuning method to better transfer from single-source to multi-source. In the first finetuning step, the coarse encoder is used to encode different sources individually. As multiple sources are concatenated as a single source in which words interact by pretrained self-attentions, we conjecture that the cross-source information

cannot be fully captured. Hence, we propose to add a randomly initialized fine encoder, which consists of self-attentions, cross-attentions, and FFNs, on top of the pretrained coarse encoder to learn meaningful multi-source representations. Specifically, the cross-attention sublayer is an essential part of the fine encoder because they perform fine-grained interaction between sources (see Table 5).

Formally, the architecture of the fine encoder can be described as follows. First, the representations of multiple sources output by the coarse encoder are divided according to the boundaries of sources:

$$\mathbf{R}_1^{(N_c)}, \ldots, \mathbf{R}_K^{(N_c)} = \mathrm{Split}\left(\mathbf{R}_{1:K}^{(N_c)}\right), \quad (10)$$

where N_c is the number of the coarse encoder layers, $\mathrm{Split}(\cdot)$ is the split operation. Second, for each fine encoder layer, the representations are fed into a self-attention sublayer:

$$\mathbf{B}_k^{(i)} = \mathrm{SelfAtt}\left(\mathbf{A}_k^{(i-1)}\right), \quad (11)$$

where $\mathbf{A}_k^{(i-1)}$ is the representation corresponding to the k-th source sentence output by the $(i-1)$-th layer of the fine encoder, in other words, $\mathbf{A}_k^{(0)} = \mathbf{R}_k^{(N_c)}$. $\mathbf{B}_k^{(i)}$ is the representation output by the self-attention sublayer of the i-th layer. Third, representations of source sentences interact through a cross-attention sublayer:

$$\mathbf{O}_{\backslash k}^{(i)} = \mathrm{Concat}\left(\mathbf{B}_1^{(i)}, \ldots, \mathbf{B}_{k-1}^{(i)}, \mathbf{B}_{k+1}^{(i)}, \ldots, \mathbf{B}_K^{(i)}\right), \quad (12)$$

$$\mathbf{C}_k^{(i)} = \mathrm{CrossAtt}\left(\mathbf{B}_k^{(i)}, \mathbf{O}_{\backslash k}^{(i)}, \mathbf{O}_{\backslash k}^{(i)}\right), \quad (13)$$

where $\mathrm{Concat}(\cdot)$ is the concatenation operation, $\mathbf{O}_{\backslash k}^{(i)}$ is the concatenated representation except $\mathbf{B}_k^{(i)}$, $\mathrm{CrossAtt}(Q, K, V)$ is the cross-attention sublayer, $\mathbf{C}_k^{(i)}$ is the representation output by the cross-attention sublayer of the i-th layer. Finally, the last sublayer is a feedforward network:

$$\mathbf{A}_k^{(i)} = \mathrm{FFN}\left(\mathbf{C}_k^{(i)}\right). \quad (14)$$

After the N_f-layer fine encoder, the representations corresponding to multiple sources are given to the decoder.

2.5 The Decoder

Given that representations of multiple sources are different from that of a single source, to better leverage representations of multiple sources, we let the cross-attention sublayer take each source's representation as key/value separately and then combine the outputs by mean pooling.[3] Formally, the differences between our decoder and the traditional Transformer decoder are described below.

First, the input representations of the i-th decoder layer are fed into the self-attention sublayer to obtain $\mathbf{G}_j^{(i)}$. Second, a *separated* cross-attention sublayer is adopted by our framework to replace the traditional cross-attention sublayer:

$$\mathbf{P}_{j,k}^{(i)} = \mathrm{CrossAtt}\left(\mathbf{G}_j^{(i)}, \mathbf{A}_k^{(N_f)}, \mathbf{A}_k^{(N_f)}\right), \quad (15)$$

$$\mathbf{H}_j^{(i)} = \mathrm{MeanPooling}\left(\mathbf{P}_{j,1}^{(i)}, \ldots, \mathbf{P}_{j,K}^{(i)}\right), \quad (16)$$

where $\mathbf{A}_k^{(N_f)}$ is the output of the fine encoder derived by Eq. (14), $\mathbf{P}_{j,k}^{(i)}$ is the representation corresponding to the k-th source, $\mathbf{H}_j^{(i)}$ is the combined result of the separated cross-attention sublayer, and the parameters of separated cross-attentions to leverage each source are shared. Finally, a feedforward network is the last sublayer of a decoder layer. In this way, the decoder in our framework can better handle representations of multiple sources.

3 Experiments

3.1 Setup

Datasets

We evaluated our framework on three MSG tasks: (1) automatic post-editing (APE), (2) multi-source translation, and (3) document-level translation.

For the APE task, following Correia and Martins (2019), we used the data from the WMT17 APE task (English-German SMT) (Chatterjee et al., 2019). The dataset contains 23K dual-source examples (e.g., ⟨English source sentence, German translation, German post-edit⟩) for training in an *extremely low*-resource setting. We also followed Correia and Martins (2019) to adopt pseudo data (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018), which contains about 8M pseudo training examples, to evaluate our framework in a *high*-resource setting. We adopted the *dev16* for development and used *test16* and *test17* for testing.

For the multi-source translation task, following Zoph and Knight (2016), we used a subset of the WMT14 news dataset (Bojar et al., 2014),

[3] There is little difference between the "parallel attention combination strategy" proposed by Libovický et al. (2018) and our method.

Models	Pretraining	TEST16		TEST17	
		TER	*BLEU*	*TER*	*BLEU*
extremely low-resource					
FORCEDATT (Berard et al., 2017)	—	22.89	—	23.08	65.57
DUALBERT (Correia and Martins, 2019)	*mBERT*	18.88	71.61	19.03	70.66
DUALBART (Correia and Martins, 2019)	*mBART*	18.26	72.65	18.41	72.08
TRICE	*mBART*	**17.41**$^{\triangle\star}$	**73.43**$^{\triangle\star}$	**17.75**$^{\triangle\star}$	**72.70**$^{\triangle\star}$
high-resource					
DUALTRANS (Junczys-Dowmunt and Grundkiewicz, 2018)	—	17.81	72.79	18.10	71.72
L2COPY (Huang et al., 2019)	—	17.45	73.51	17.77	72.98
DUALBERT (Correia and Martins, 2019)	*mBERT*	16.91	74.29	17.26	73.42
DUALBART (Correia and Martins, 2019)	*mBART*	16.40	74.74	17.26	73.56
TRICE	*mBART*	**16.09**$^{\triangle\star}$	**75.39**$^{\triangle\star}$	**16.91**$^{\triangle\star}$	**74.09**$^{\triangle\star}$

Table 2: Results on the automatic post-editing task (*extremely low-* and *high*-resource). "DUALBART": a method to leverage pretrained Seq2Seq models adapted from "DUALBERT". Please refer to Appendix A.3 for detailed descriptions of baselines and the same below. "$\triangle$": significantly better than "DUALBERT" ($p < 0.01$). "$\star$": significantly better than "DUALBART" ($p < 0.01$).

Models	Source	Pretraining	TEST14
MULTIRNN (Zoph and Knight, 2016)	*(De, Fr)*	—	30.0
DUALTRANS (Junczys-Dowmunt and Grundkiewicz, 2018)	*(De, Fr)*	—	37.0
MBART-TRANS (Liu et al., 2020)	*De*	*mBART*	31.8
MBART-TRANS (Liu et al., 2020)	*Fr*	*mBART*	34.8
DUALBART (Correia and Martins, 2019)	*(De, Fr)*	*mBART*	40.2
TRICE	*(De, Fr)*	*mBART*	**41.5***

Table 3: Results on the multi-source translation task (*medium*-resource). In this task, German and French sources are translated to English target. "MBART-TRANS": a single-source model directly finetuned from mBART. "$\star$": significantly better than "DUALBART" ($p < 0.01$).

Models	#Context	Pretraining	TED		News	
			s-BLEU	*d-BLEU*	*s-BLEU*	*d-BLEU*
SAN (Maruf et al., 2019)	—	—	24.4	—	24.8	—
QCN (Yang et al., 2019)	—	—	25.2	—	22.4	—
MCN (Zheng et al., 2020)	—	—	25.1	29.1	24.9	27.0
MBART-TRANS (Liu et al., 2020)	0	*mBART*	28.1	31.7	29.4	31.2
MBART-DOCTRANS (Liu et al., 2020)	1	*mBART*	28.1	31.7	28.6	30.2
DUALBART (Correia and Martins, 2019)	1	*mBART*	27.8	31.4	29.3	31.3
TRICE	1	*mBART*	**28.5**†‡*	**32.1**†‡*	**29.8**†‡*	**31.7**†‡*

Table 4: Results on the document-level translation task (*low*-resource). "s-BLEU" and "d-BLEU" denote BLEU scores calculated at sentence- and document-level, respectively. "#Context" denotes the number of context used by context-aware models. "$\dagger$": significantly better than "MBART-TRANS" ($p < 0.01$). "$\ddagger$": significantly better than "MBART-DOCTRANS" ($p < 0.01$). "$\star$": significantly better than "DUALBART" ($p < 0.01$).

which contains 2.4M dual-source examples (e.g., ⟨German source sentence, French source sentence, English translation⟩) for training, 3,000 from *test13* for development, and 1,503 from *test14* for testing.[4] It can be seen as a *medium*-resource setting.

For the document-level translation task, we used the dataset provided by Maruf et al. (2019) from IWSLT2017 (TED) and News Commentary (News), both including about 200K English-German training examples, which can be seen as *low*-resource settings. For IWSLT2017, *test16* and *test17* were combined as the test set, and the rest served as the development set. For News Commentary, *test15* and *test16* in WMT16 were used for development and testing, respectively. We took the nearest preceding sentence as the context, and then constructed the dual-source example like ⟨German context, German current sentence, English translation⟩.

Hyper-parameters

We adopted mBART (Liu et al., 2020) as the pretrained Seq2Seq model. We set both N_c and N_d to 12, and N_f to 1. The model dimension, the filter size, and the number of heads are the same as mBART. We adopted the vocabulary of mBART, which contains 250K tokens. We used minibatch sizes of 256, 1,024, 4,096, and 16,384 tokens for *extremely low-, low-, medium-*, and *high*-resource settings, respectively. We used the development set to tune the hyper-parameters and select the best model. In inference, the beamsize was set to 4. Please refer to Appendix A.1 for more details.

Evaluation Metrics

We used case-sensitive BLEU (*multi-bleu.perl*) and TER for automatic post-editing. For multi-source translation and document-level translation, SACREBLEU[5] (Post, 2018) and METEOR[6] was adopted for evaluation. We used the paired bootstrap resampling (Koehn, 2004) for statistical significance tests.

3.2 Main Results

Table 2 shows the results on the automatic post-editing task. Our framework outperforms previous methods without pretraining (i.e., FORCEDATT, DUALTRANS, and L2COPY) by a large margin and surpasses strong baselines with pretraining (i.e., DUALBERT and DUALBART), which concatenate multiple sources into a single source, significantly in both *extremely low*- and *high*-resource settings. Notably, the performances of our framework in the *extremely low*-resource setting are comparable to results of strong baselines without pretraining in the *high*-resource setting and we achieve new state-of-the-art results on this benchmark.

Variants	#Para.	BLEU
None	0M	73.65
FFN adapter (Guo et al., 2020)	100.7M	73.71
Fine encoder ($N_f = 1$) w/o CA	12.5M	73.84
Fine encoder ($N_f = 1$)	16.8M	**74.21**
Fine encoder ($N_f = 2$)	33.6M	73.70
Fine encoder ($N_f = 3$)	50.4M	58.84

Table 5: Comparisons with the variants of the fine encoder. "#Para." denotes the number of parameters and "CA" denotes the cross-attention sublayer. "N_f" denotes the number of the fine encoder layers.

Table 3 demonstrates the results on the multi-source translation task. Our framework substantially outperforms both baselines without pretraining (i.e., MULTIRNN and DUALTRANS) and with pretraining (i.e., single-source model MBART-TRANS and dual-source model DUALBART). Surprisingly, the single-source models with pretraining are inferior to the multi-source model without pretraining, which indicates that multiple sources play an important role in the translation task.

Table 4 shows the results on the document-level translation task. Our framework achieves significant improvements over all strong baselines. Unusually, the previous method for handling multiple sources (i.e., DUALBART) fails to consistently outperform simple sentence- and document-level Transformer (i.e., MBART-TRANS and MBART-DOCTRANS) while our framework outperforms these strong baselines significantly.

In general, our framework shows a strong generalizability across three different MSG tasks and four different data scales, which indicates that it is useful to alleviate the pretrain-finetune discrepancy by gradual finetuning and learn multi-source representations by fully capturing cross-source information.

[4] A dual-source example can be obtained by matching two single-source examples.

[5] The signature is "BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14".

[6] https://www.cs.cmu.edu/~alavie/METEOR/

Model Variants	BLEU
TRICE	**74.21**
– gradual finetuning	73.83
– separated cross-attention	73.81
– concatenated encoding	73.61
– segment embedding	72.92

Table 6: Ablation study. The case-sensitive BLEU scores are calculated on the development set of the APE task for all experiments for analyses. Note that we remove only one component at a time.

Components to Finetune	BLEU
All	**74.21**
The fine encoder	70.20

Table 7: Effect of freezing pretrained parameters.

3.3 Analyses

In this subsection, we further conduct studies regarding the variants of the fine encoder, ablations of the other proposed components, and effect of freezing parameters. Experiments are conducted on the APE task in the *extremely low*-resource setting. The BLEU scores calculated on the development set are adopted as the evaluation metric.

Comparisons with the variants of the fine encoder. Table 5 demonstrates comparisons with the variants of the fine encoder. We find that the fine encoder (see Section 2.4) is effective (compared to "*None*"), the cross-attention sublayer is important (compared to the one without cross-attention), and our approach outperforms "FFN adapter", which is proposed by Zhu et al. (2019) to incorporate BERT into sequence generation tasks by inserting FFNs into each encoder layer. We find that stacking more fine encoder layers even harms the performance (see the last three rows in Table 5) which rules out the option that the improvements owe to increasing of parameters.

Ablations on the other proposed components. Table 6 shows the results of the ablation study. We find that gradual finetuning method (see Section 2.2) is significantly beneficial. Lines "- segment embedding" and "- concatenated encoding" show that concatenating multiple sources into a long sequence and adding sinusoidal segment embedding for the coarse encoder are helpful (see Section 2.3). The line "- separated cross-attention" reveals that taking each source's representation as key/value separately and then combine the outputs is better than concatenating all the representations and do the cross-attention jointly (see Section 2.5).

Effect of freezing pretrained parameters. As shown in Table 7, finetuning all parameters including parameters initialized by pretrained models and parameters initialized randomly is essential for achieving good performance on MSG tasks.

3.4 Adversarial Evaluation

We adopt adversarial evaluation similar to Libovický et al. (2018) which replaces one source with a randomly selected sentence. As shown in Table 8, both sources play important parts and the French side is more important than the German side (Randomized Fr vs. Randomized De).

3.5 Case Study

An example in multi-source translation task is shown in Table 9. The four outputs at the bottom of the table are generated by the last four models in Table 3. We find that single-source models have different errors (e.g., "each hospitals" and "travelling clinics") and multi-source models fix some errors because of taking two sources. Additionally, DualBart still output erroneous "weekly", while TRICE outputs "weekend" successfully. We believe TRICE is better than baselines because multiple sources are complementary and the fine encoder could capture finer cross-source information, which helps correct translation errors.

4 Related Work

4.1 Multi-source Sequence Generation

Multi-source sequence generation includes multi-source translation (Zoph and Knight, 2016), automatic post-editing (Chatterjee et al., 2017), multi-document summarization (Haghighi and Vanderwende, 2009), system combination for NMT (Huang et al., 2020), and document-level machine translation (Wang et al., 2017), etc. For these tasks, researchers usually leverage multi-encoder architectures to achieve better performance (Zoph and Knight, 2016; Zhang et al., 2018; Huang et al., 2019). To address the data scarcity problem in MSG, some researchers generate pseudo corpora (Negri et al., 2018; Nishimura et al., 2020) to augment the corpus size while others try to make use of pretrained autoencoding models (e.g., BERT

Models	Normal		Randomized Fr		Randomized De	
	BLEU	*METEOR*	*BLEU*	*METEOR*	*BLEU*	*METEOR*
MBART-TRANS (De)	31.8	33.9	—	—	—	—
MBART-TRANS (Fr)	34.8	37.9	—	—	—	—
DUALBART	40.2	38.9	11.3	13.1	**24.9**	**26.4**
TRICE	**41.5**	**39.8**	**13.5**	**15.0**	23.0	23.9

Table 8: Adversarial evaluation on the multi-source translation task. "Randomized Fr/De" denotes that the Fr/De source is replaced with a randomly selected sentence.

Input-Fr	Dans cet hôpital itinérant, divers soins de santé sont prodigués.
Input-De	Jede dieser Wochenendkliniken bietet medizinische Versorgung in einer Reihe von Bereichen an.
Reference-En	Each of these weekend clinics provides a variety of medical care.
MBART-TRANS (De)	Each weekend hospitals offers medical care in a number of areas.
MBART-TRANS (Fr)	This travelling clinics provides a variety of healthcare services.
DUALBART	Each of these weekly hospitals provides healthcare in a variety of areas.
TRICE	Each of these weekend clinics offers a variety of health care.

Table 9: Example of multi-source translation. Some erroneous parts are highlighted by underlines. MBART-TRANS (De/Fr) takes single source (De/Fr) as input while DUALBART and TRICE take both sources as input. We believe that multiple sources are complementary and TRICE could correct errors by capturing finer cross-source information.

(Devlin et al., 2019) and XLM-R (Conneau et al., 2020)) to enhance specific MSG tasks (Correia and Martins, 2019; Lee et al., 2020; Lee, 2020). Different from these works, we propose a task-agnostic framework to transfer pretrained Seq2Seq models to multi-source sequence generation tasks and demonstrate the generalizability of our framework.

4.2 Pretraining

In recent years, self-supervised methods have achieved remarkable success in a wide range of NLP tasks (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020; Radford et al., 2019; Song et al., 2019; Lewis et al., 2020; Liu et al., 2020). The architectures of pretrained models can be roughly divided into three categories: autoencoding (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020), autoregressive (Radford et al., 2019), Seq2Seq (Song et al., 2019; Raffel et al., 2020; Lewis et al., 2020; Liu et al., 2020). Some researchers propose to use pretrained autoencoding models to improve sequence generation tasks (Zhu et al., 2019; Guo et al., 2020) and the APE task (Correia and Martins, 2019). For pretrained Seq2Seq models, it is convenient to use them to initialize single-source sequence generation models without further modification. Different from these works, we transfer pretrained Seq2Seq models to multi-source sequence generation tasks.

5 Conclusion

We propose a novel task-agnostic framework, TRICE, to conduct transfer learning from single-source sequence generation including self-supervised pretraining and supervised generation to multi-source sequence generation. With the help of the proposed gradual finetuning method and the novel MSG model equipped with coarse and fine encoders, our framework outperforms all baselines on three different MSG tasks in four different data scales, which shows the effectiveness and generalizability of our framework.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFB0202204), National Natural Science Foundation of China (No.61925601, No. 61772302). We thank all anonymous reviewers for their valuable comments and suggestions on this work.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of ICCV*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of ICML*.

Alexandre Berard, Laurent Besacier, and Olivier Pietquin. 2017. Lig-cristal submission for the wmt 2017 automatic post-editing task. In *Proceedings of WMT*.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia. 2014. Proceedings of the ninth workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

Rajen Chatterjee, M Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task. In *Proceedings of WMT*.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of WMT*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Gonçalo M Correia and André FT Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of ACL*.

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

Stéphane Dupont and Juergen Luettin. 2000. Audio-visual speech modeling for continuous speech recognition. *IEEE transactions on multimedia*, 2(3):141–151.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. In *Proceedings of NeurIPS*.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of NAACL-HLT*.

Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer. 2016. Attention-based multimodal neural machine translation. In *Proceedings of WMT*.

Xuancheng Huang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2019. Learning to copy for automatic post-editing. In *Proceedings of EMNLP*.

Xuancheng Huang, Jiacheng Zhang, Zhixing Tan, Derek F. Wong, Huanbo Luan, Jingfang Xu, Maosong Sun, and Yang Liu. 2020. Modeling voting for system combination in machine translation. In *Proceedings of IJCAI*.

Julia Ive, Pranava Swaroop Madhyastha, and Lucia Specia. 2019. Distilling translations with visual awareness. In *Proceedings of ACL*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of WMT*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proceedings of WMT*.

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

Dongjun Lee. 2020. Cross-lingual transformers for neural automatic post-editing. In *Proceedings of WMT*.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020. Postech-etri's submission to the wmt2020 ape shared task: Automatic post-editing with cross-lingual language model. In *Proceedings of WMT*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Jindrich Libovickỳ, Jindrich Helcl, and David Marecek. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of WMT*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *TACL*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of NAACL-HLT*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of LREC*.

Y. Nishimura, K. Sudoh, G. Neubig, and S. Nakamura. 2020. Multi-source neural machine translation with missing data. *TASLP*, 28:569–580.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of WMT*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of ACL*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

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

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. In *Proceedings of EMNLP*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of EMNLP*.

Zhengxin Yang, Jinchao Zhang, Fandong Meng, Shuhao Gu, Yang Feng, and Jie Zhou. 2019. Enhancing context modeling with a query-guided capsule network for document-level translation. In *Proceedings of EMNLP*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of EMNLP*.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Towards making the most of context in neural machine translation. In *Proceedings of IJCAI*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2019. Incorporating bert into neural machine translation. In *Proceedings of ICLR*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of NAACL-HLT*.

A Experiment Setup

A.1 Model Configurations

We adopted mBART (*mBART-cc25*) (Liu et al., 2020) as the pretrained Seq2Seq model. mBART is a Seq2Seq model obtained by multilingual denoising pretraining on a subset of Common Crawl corpus. Following mBART, we set the number of layers of the Coarse-Encoder (i.e., N_c) and the number of the Decoder layers (i.e., N_d) to 12. Especially, the number of the Fine-Encoder layers (i.e., N_f) was set to 1. The model dimension, the filter size, and the number of heads are the same as mBART. We adopted the sentencepiece model provided by mBART for tokenization and adopted the vocabulary of mBART, which contains 250K tokens.

A.2 Hyper-parameters and Evaluation

We used minibatch sizes of 256, 1,024, 4,096, and 16,384 tokens for *extremely low-, low-, medium-*, and *high*-resource settings, respectively. In each stage of finetuning, we used Adam (Kingma and Ba, 2015) for optimization and used the learning rate decay policy described by Vaswani et al.

(2017). We used the development set to tune the hyper-parameters and select the best model. In inference, the beamsize was set to 4 and the length penalty was set to 1.0 , 0.6 and 0 for APE, multi-source translation, and document-level translation, respectively. We used four GeForce RTX 2080Ti GPUs for training. We used case-sensitive BLEU (*multi-bleu.perl*) and TER[7] for automatic post-editing. For multi-source translation and document-level translation, SACREBLEU[8] (Post, 2018) and METEOR[9] was used for evaluation. We used the paired bootstrap resampling (Koehn, 2004) for statistical significance tests.

A.3 Baselines

The asterisks ("*") below denote that we report results of these baseline in our implementations in the same hyper-parameter settings as our approach.

Automatic Post-Editing

In the automatic post-editing task, we compare our approach with the following baselines:

1. FORCEDATT (Berard et al., 2017): a monosource model with a task-specific attention mechanism.

2. DUALTRANS (Junczys-Dowmunt and Grundkiewicz, 2018): a dual-source Transformer based model for APE.

3. L2COPY (Huang et al., 2019): a dual-source model enabling cross-source interaction, which focuses on modeling copying mechanism in APE.

4. DUALBERT (Correia and Martins, 2019): the first method to use pretrained models to enhance APE, which concatenates multiple sources as a single source and uses two BERT models to initialize the encoder and decoder separately.

5. DUALBART* (Correia and Martins, 2019): adapting DUALBERT to leverage pretrained Seq2Seq models by concatenating multiple sources as a single source and feeding it to Seq2Seq models.

[7] http://www.cs.umd.edu/~snover/tercom/

[8] The signature is "BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14"

[9] https://www.cs.cmu.edu/~alavie/METEOR/

Multi-source Translation

In the multi-source translation task, we compare our approach with the following baselines:

1. MULTIRNN (Zoph and Knight, 2016): a multi-source encoder-decoder model based on RNN for machine translation.

2. DUALTRANS* (Junczys-Dowmunt and Grundkiewicz, 2018): a dual-source Transformer based model.

3. MBART-TRANS* (Liu et al., 2020): a transferring method that directly finetunes the pretrained single-source sequence generation model on the downstream task and takes single-source input during both training and inference.

4. DUALBART* (Correia and Martins, 2019): adapting DUALBERT to leverage pretrained Seq2Seq models by concatenating multiple sources as a single source and feeding it to the Seq2Seq model.

Document-level Translation

In the document-level translation task, we compare our approach with the following baselines:

1. SAN (Maruf et al., 2019): a context-aware NMT model with selective attentions.

2. QCN (Yang et al., 2019): a context-aware NMT model using a query-guided capsule network.

3. MCN (Zheng et al., 2020): a general-purpose NMT model that is supposed to deal with any-length text.

4. MBART-TRANS* (Liu et al., 2020): a transferring method that directly finetunes the pretrained single-source sequence generation model on the downstream task and takes single-source input during both training and inference.

5. MBART-DOCTRANS* (Liu et al., 2020): a method for document-level translation, which takes K (not more than the number of sentences in a document) source sentences as input and translates K target sentences through a SSG model all at once. For fair comparison, we set K to 2 for both MBART-DOCTRANS and our approach.

6. DUALBART* (Correia and Martins, 2019): adapting DUALBERT to leverage pretrained Seq2Seq models by concatenating multiple sources as a single source and feeding it to the Seq2Seq model.