CAR-Transformer: Cross-Attention Reinforcement Transformer for Cross-Lingual Summarization

Yuang Cai, Yuyu Yuan*

Beijing University of Posts and Telecommunications
Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education
{cyang,yuanyuyu}@bupt.edu.cn

Abstract

Cross-Lingual Summarization (CLS) involves generating a summary for a given document in another language. Most of the existing approaches adopt multi-task training and knowledge distillation, which increases the training cost and improves the performance of CLS tasks intuitively but unexplainably. In this work, we propose Cross-Attention Reinforcement (CAR) module and incorporate the module into the transformer backbone to formulate the CAR-Transformer. The CAR module formulates a pseudo-summary policy parameterized by the cross-attention weights reinforced by the ground-truth monolingual summary without introducing extra model parameters. Our approach demonstrates more consistent improvement across CLS tasks compared to traditional multi-task training methods and outperforms the fine-tuned vanilla mBART by 3.67 and the best-performing multi-task training approach by 1.48 in ROUGE-L F1 score on the WikiLingua Korean-to-English CLS task.

Introduction

Cross-lingual summarization (CLS) refers to the process of generating a summary in a different language for a given document. There are two main categories of methods, which are pipeline methods and end-to-end methods, for cross-lingual summarization. Pipeline methods involve breaking down CLS into two sub-tasks, namely Monolingual Summarization (MS) and Machine Translation (MT), and executing them sequentially (Leuski et al. 2003; Lim, Kang, and Lee 2004; Orăsan and Chiorean 2008; Wan, Li, and Xiao 2010). Although these approaches may seem intuitive, they are hindered by several limitations, including error accumulation, reliance on external data/models, and high inference latency, as noted in Wang et al. 2022. In an effort to address these limitations, end-to-end approaches based on neural networks have been proposed (Zhu et al. 2019; Cao, Liu, and Wan 2020; Bai, Gao, and Huang 2021; Takase and Okazaki 2022; Liang et al. 2022). In particular, multilingual pre-trained transformers have emerged as a noteworthy development, achieving state-of-the-art performance on CLS tasks. Some approaches involve training these transformers on a multilingual corpus (Liu et al. 2020; Tang et al. 2021; Xue et al. 2021), while others introduce cross-lingual training objectives to enhance performance on downstream cross-lingual seq2seq tasks (Xu et al. 2020; Chi et al. 2020; Ma et al. 2021).

Cross-lingual summarization (CLS) datasets typically consist of a document paired with multiple parallel summaries in different languages (Ladhak et al. 2020; Bhattacharjee et al. 2021; Perez-Beltrachini and Lapata 2021). Among these summaries, there is usually one that corresponds to the same language as the source document, referred to as the monolingual summary. The monolingual summary plays a crucial role in existing CLS approaches as it is widely utilized to enhance performance. To leverage the monolingual summary, various techniques, such as multi-task training and knowledge distillation, have been adopted. Multi-task training involves jointly performing MS and CLS, while knowledge distillation entails extracting knowledge from a pre-trained MS model (Wang et al. 2022). Although these approaches can improve CLS performance by utilizing the monolingual summary, they do introduce additional modules and/or training samples, which come at a cost. Furthermore, while multi-task training and knowledge distillation have shown intuitive improvements in CLS performance, they often lack interpretability. Recognizing this limitation, Duan et al. 2019 propose a more explainable approach by distilling cross-attention weights from a teacher model. However, this approach still incurs costs due to the utilization of knowledge distillation. Another approach, put forth by Liang et al. 2022, suggests the adoption of hierarchical structures for translation and summarization. Although this approach shows promise, it introduces multiple additional encoders for Machine Translation (MT) and Monolingual Summarization (MS), leading to a significant increase in training and inference costs.

Consider a human editor proficient in multiple languages. We require the editor to generate parallel summaries in different languages, ensuring that the meanings of the summaries remain the same for a given document. Intuitively, whether to summarize the document in which language, the editor will pay attention to similar parts of the document. Fortunately, most existing fundamental seq2seq models employ a cross-attention mechanism, which describes the attention from the target sequence (i.e., the summary) to the source sequence (i.e., the document) (Bahdanau, Cho,

*Corresponding author.

and Bengio 2014; Vaswani et al. 2017; Lewis et al. 2019). Therefore, a straightforward solution is to establish cross-attention alignment, specifically aiming to maintain similarity between cross-attentions used in cross-lingual summarization and those used in monolingual summarization. This alignment enables both Cross-Lingual Summarization (CLS) and Monolingual Summarization (MS) to attend to corresponding words in the source document.

The direct implementation of cross-attention alignment within the context of machine translation entails the need to undertake the costly process of MS training, which significantly amplifies both the temporal and spatial training requirements. In order to tackle these challenges, we propose the CAR-Transformer (**C**ross-**A**ttention **R**einforcement Transformer) framework, which commences by formulating a pseudo-summary policy based on the cross-attention weights without introducing extra learnable parameters. This policy facilitates the generation of a pseudo-summary in the monolingual language by selecting words from the source document, with higher probabilities assigned to words that have accumulated greater cross-attention weights. The policy is subsequently trained by incentivizing it to generate a summary that closely approximates the ground-truth monolingual summary. During training, policy gradient methods (Williams 1992) are employed due to the non-differentiability of the summary similarity computation. As a result, words that appear more frequently in the ground-truth monolingual summary will exhibit higher accumulated cross-attention scores. We refer to the training process as cross-attention reinforcement since the policy is parameterized by the cross-attention weights and optimized through policy gradient.

Our approach surpasses the vanilla mBART fine-tuning approach and three common multi-task training approaches, i.e., CLS-MS (Zhu et al. 2019), CLS-MT (Zhu et al. 2019), and 2-step (Ladhak et al. 2020), on WikiLingua (Ladhak et al. 2020), GlobalVoice (Nguyen and Daumé III 2019), and CrossSum (Bhattacharjee et al. 2021) datasets. Specifically, our approach outperforms the best multi-task training approach to our knowledge by 1.55, 1.04, 0.50, and 1.04 in ROUGE-L on Korean, Hindi, Czech, and Turkish to English summarization, respectively.

This work makes the following main contributions:

- We are the first, to the best of our knowledge, to address the challenge of cross-attention alignment in cross-lingual summarization tasks where parallel ground-truth summaries are available.
- We present cross-attention reinforcement, a lightweight (no extra parameters) and explainable auxiliary training objective, which effectively enhances the performance of the Cross-Lingual Summarization (CLS) task.
- We demonstrate significant and consistent improvements compared to three commonly adopted multi-task training approaches on CLS tasks involving eight different languages.

Related Work

Attention Relay

Our work is to some extent relevant to the attention relay mechanism proposed in (Duan et al. 2019) for teaching the student model the attention weights of the teacher model in knowledge distillation. Both our work and attention relay share the idea of optimizing the cross-attention weights to enhance the performance of the CLS task in a more explainable manner. However, attention relay relies on teacher models trained on the MS corpus and machine translation (MT) to facilitate knowledge distillation, which leads to increased training costs. In contrast, our approach offers a more lightweight alternative by directly manipulating the cross-attention weights without the need for external models.

Multilingual Pre-Trained Transformers

Multilingual pre-trained transformers have demonstrated remarkable performance on CLS tasks (Wang et al. 2022). One straightforward approach to pre-training these multilingual transformers is to adapt the pre-training objective used in monolingual transformers by replacing the monolingual training corpus with a multilingual corpus. For instance, mBART (Liu et al. 2020) and mBART50 (Tang et al. 2021) are two variations of the BART model (Lewis et al. 2019). Both mBART and mBART50 employ the denoising pre-training objective, which is the same as that used in the BART model. They utilize a training corpus consisting of 25 languages and 50 languages, respectively. Another example is mT5 (Xue et al. 2021), which is a variant of the T5 model (Raffel et al. 2020). mT5 uses a training corpus encompassing 101 languages and adopts the same corruption pre-objective as the T5 model.

Multi-Task Training in CLS

Multi-task training is an approach that involves jointly optimizing the training objectives of multiple tasks, leveraging the potential benefits of inter-task interaction during training. One commonly cited explanation for the effectiveness of multi-task training is that the simultaneous training of multiple tasks helps prevent each individual task from overfitting. In the context of the CLS task, Monolingual Summarization (MS) and Machine Translation (MT) are often employed as auxiliary training objectives. For instance, Zhu et al. 2019 propose a bi-decoder approach where two decoders are utilized to perform CLS and MS/MT jointly, aiming to enhance the performance of the CLS task.

Similarly, Ladhak et al. 2020 propose a 2-step training procedure that involves performing MT first and then performing CLS. The motivation behind incorporating MS and MT as auxiliary objectives is intuitive since the CLS task requires both translation and (monolingual) summarization abilities. However, it is important to note that the bi-decoder approach introduces additional training parameters, while the 2-step approach increases the number of training steps, thereby raising the overall training cost. Furthermore, although the improvements observed with the inclusion of auxiliary tasks are intuitive, the underlying reasons behind

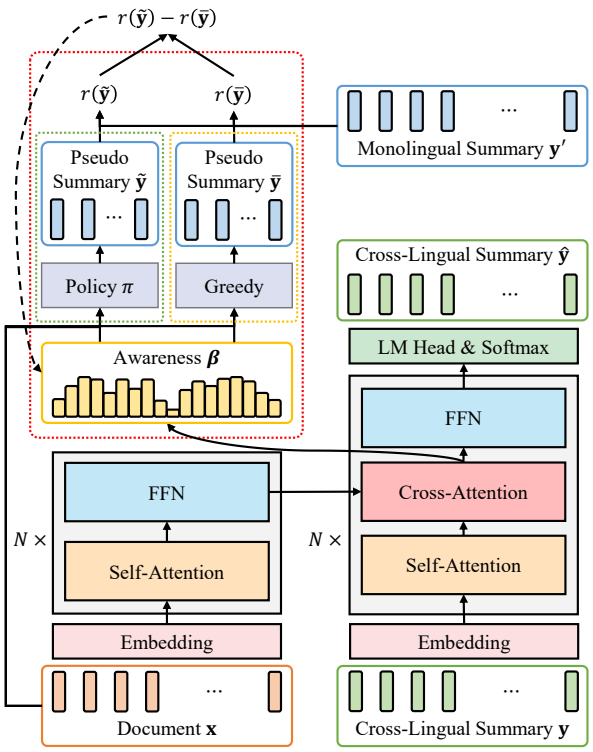

Figure 1: Overview of CAR-transformer.

these improvements may not be fully explainable. In other words, while the benefits of incorporating auxiliary tasks are evident, the precise mechanisms by which they contribute to enhanced performance may not be entirely comprehensible.

CAR-Transformer

Problem Formulation

Given the source document $\mathbf{x} = (x_1, x_2, \cdots, x_S)$, the objective of CLS is to generate its summary in another language $\mathbf{y} = (y_1, y_2, \cdots, y_T)$. Taking the vanilla transformer architecture (Vaswani et al. 2017) as an example, we formulate the training process of generating the cross-lingual summary $\mathbf{y}$ from the source document $\mathbf{x}$ in detail.

The encoder E takes the source document $\mathbf{x}$ as input and outputs the encoder hidden states $\mathbf{h}$, as shown in Equation 1. In order to distinguish from the encoder hidden states, we denote the output hidden states of the l-th decoder layer as $\mathbf{v}^l$ and define $\mathbf{v}^0$ as the output of the embedding layer. The self-attention layer of the l-th decoder layer D^l_{self} takes the output of the previous decoder layer as input and outputs an intermediate state $\mathbf{c}^l$, as shown in Equation 2. Then, the cross-attention weight matrix of the l-th layer A^l is computed as in Equation 3, and output hidden states are further computed as in Equation 4. Here, $W^Q\mathbf{c}^l$, $W^K\mathbf{h}$, and $W^V\mathbf{h}$ respectively denote the query, key, and value, d_k is the dimension of queries and keys, and FFN is the feed-forward network (Vaswani et al. 2017). Finally, the distribution of the output token is computed as in Equation 5, where L is the max number of decoder layers, W^O is the parameter of the final linear layer, and v^L_i denotes the i-th output hidden state of the last decoder layer.

$$\mathbf{h} = E(\mathbf{x}) \tag{1}$$

$$\mathbf{c}^l = D^l_{\text{self}}(\mathbf{v}^{l-1}),\ l = 1, 2, \cdots, L \tag{2}$$

$$A^l = \text{softmax}\left(\frac{(W^Q\mathbf{c}^l)(W^K\mathbf{h})^T}{\sqrt{d_k}}\right) \tag{3}$$

$$\mathbf{v}^l = \text{FFN}(A^l(W^V\mathbf{h})) \tag{4}$$

$$p_i = \text{softmax}(W^O v^L_i) \tag{5}$$

The training objective is to minimize the cross-entropy loss between the predicted token distribution p_i and the ground-truth target token, as shown in Equation 6, where XE denotes the cross-entropy loss function.

$$\mathcal{L}_{\text{xe}} = \frac{1}{T}\sum_{i=1}^{T}\text{XE}(p_i, y_i) \tag{6}$$

Model Architecture

In the CAR-transformer architecture, the design is based on the vanilla transformer (Vaswani et al. 2017). Before diving into the detailed design, let's provide an overview of the model architecture. As shown in Figure 1, the encoder-decoder architecture constitutes the modules outside the red dotted square. The encoder takes the source document as input, while the decoder takes the ground-truth cross-lingual summary as input and generates the predicted summary. The red dotted square represents the cross-attention reinforcement (CAR) module, which plays a crucial role in the CAR-transformer. It consists of three key components: the awareness computation, represented by the green dotted square; the pseudo-summary policy , denoted by the yellow dotted square; and the self-critical baseline, also depicted by the yellow dotted square. In this module, $r(\tilde{\mathbf{y}})$ represents the reward obtained from the pseudo-summary policy, while $r(\bar{\mathbf{y}})$ corresponds to the reward obtained from the self-critical baseline. The reward normalized by the self-critical baseline is denoted as $r(\tilde{\mathbf{y}}) - r(\bar{\mathbf{y}})$. To train the CAR-transformer, the gradient of the normalized reward is estimated and back-propagated, as illustrated by the black dashed line in Figure 1. In the upcoming sections, we will provide detailed explanations of the training process of the CAR module.

Cross-Attention Alignment

Before delving into cross-attention reinforcement, let's discuss Cross-Attention Alignment (CAA) and its purpose. CAA aims to make the cross-attention weight matrix of one encoder-decoder model somewhat similar to that of another model. To perform CAA, we start by training a Monolingual Summarization (MS) model using the source documents and the corresponding ground-truth monolingual summaries. Once the MS model is trained, we move on to training a Cross-Lingual Summarization (CLS) model while aligning its cross-attention with the MS model. In the CLS model, the cross-attention weight matrix of the last decoder

layer can be represented as $A \in \mathbb{R}^{T \times S}$, where T denotes the length of the target sequence (summary) and S denotes the length of the source document. To compute the awareness vector, denoted as $\boldsymbol{\beta}$, we sum the cross-attention matrix A along the query axis. In other words, for each source token x_i, we compute β_i as in Equation 7, where a_{ki} represents the attention weight between the k-th token in the target sequence and the i-th token in the source document. The resulting vector $\boldsymbol{\beta}$ captures the level of attention that the entire target sequence pays to each source token. Hence, we refer to $\boldsymbol{\beta}$ as the awareness vector, with β_i indicating the awareness of source token x_i. Figure 1 illustrates the process of computing the awareness vector based on the cross-attention weights. This step is an important component of CAA and sets the stage for cross-attention reinforcement, which we will discuss in detail later.

$$\beta_i = \frac{1}{T} \sum_{k=1}^{T} a_{ki} \tag{7}$$

To align the cross-attention between the CLS model and the trained MS model, the goal is to minimize the difference between the awareness vector $\boldsymbol{\beta}$ of the CLS model and the awareness vector $\boldsymbol{\beta'}$ of the MS model. This alignment encourages the CLS model to attend to the same or similar source tokens as the MS model during the summarization process. Mathematically, this objective can be expressed as:

$$\begin{aligned} \mathcal{L}_{\text{caa}} &= \|\boldsymbol{\beta} - \boldsymbol{\beta'}\|^2 \\ &= \frac{1}{S} \sum_{i=1}^{S} \Big(\frac{1}{T} \sum_{k=1}^{T} a_{ki} - \frac{1}{K} \sum_{k=1}^{K} a'_{ki} \Big)^2 \end{aligned} \tag{8}$$

In this context, the trained MS model serves as a teacher for the CLS model, providing guidance on attending to the correct source tokens. By minimizing the difference between $\boldsymbol{\beta}$ and $\boldsymbol{\beta'}$, the CLS model learns to align its cross-attention with that of the MS model, enhancing its ability to capture important information from the source document during the summarization process. This alignment contributes to the overall performance improvement of the CLS model.

Cross-Attention Reinforcement

The computation of the CAA loss entails training an MS model, which results in increased temporal and spatial training costs. To address this issue and provide a more lightweight solution for cross-attention alignment, we propose an alternative approach called cross-attention reinforcement. Since the MS task serves as an auxiliary task within the CLS framework, it is not necessary to obtain an accurate monolingual summary during training. Therefore, we can formulate a **pseudo-summary policy** (monolingual) based on the awareness vector $\boldsymbol{\beta}$ derived from the CLS cross-attentions, eliminating the need for an additional MS decoder. Specifically, we regard the awareness β_i as the probability of x_i being selected as the pseudo-summary. We denote the pseudo-summary distribution in Equation 9, where $\boldsymbol{\sigma}$ represents a binary selection vector indicating whether each token is selected, with $\sigma_i = 1$ denoting that x_i is chosen as part of the pseudo-summary. Consequently, the pseudo-summary $\tilde{\mathbf{y}} = (\tilde{y}_1, \cdots, \tilde{y}_M)$ is generated based on the selection vector, as shown in Equation 10, where $\iota(\boldsymbol{\sigma}, i)$ denotes the index of the i-th one-element in the binary vector $\boldsymbol{\sigma}$ and $M = \sum_{i=1}^{S} \sigma_i$ represents the count of one-elements in $\boldsymbol{\sigma}$. The process of sampling the pseudo-summary $\tilde{\mathbf{y}}$ from the policy parameterized by $\boldsymbol{\beta}$ is depicted within the green dotted square in Figure 1.

$$\pi(\boldsymbol{\sigma}; \boldsymbol{\beta}) = \prod_{i=1}^{S} [\sigma_i \beta_i + (1 - \sigma_i)(1 - \beta_i)] \tag{9}$$

$$\tilde{\mathbf{y}} = (x_{\iota(\boldsymbol{\sigma},1)}, x_{\iota(\boldsymbol{\sigma},2)}, \cdots, x_{\iota(\boldsymbol{\sigma},M)}) \tag{10}$$

We introduce a reward function to evaluate the quality of the generated pseudo-summary $\tilde{\mathbf{y}}$, as depicted in Equation 11, where f_{rg} represents a linear combination of the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores (Lin 2004) that are commonly used metrics for assessing summarization quality, and $\mathbf{y}'$ is the ground-truth monolingual summary. The reward function measures the similarity between the pseudo-summary and the ground-truth monolingual summary. Our objective is to maximize the expected reward, which is equivalent to minimizing the negative expected reward shown in Equation 12, where $\mathcal{D}$ denotes the entire dataset. The expression $\boldsymbol{\beta}(\mathbf{x}, \mathbf{y}')$ indicates that $\boldsymbol{\beta}$ is a function of $\mathbf{x}$ and $\mathbf{y}'$. We refer to the loss denoted in Equation 12 as the Cross-Attention Reinforcement (CAR) loss.

$$r(\tilde{\mathbf{y}}, \mathbf{y}') = f_{\text{rg}}(\tilde{\mathbf{y}}, \mathbf{y}') \tag{11}$$

$$\mathcal{L}_{\text{car}} = - \sum_{(\mathbf{x}, \mathbf{y}') \in \mathcal{D}} \mathbb{E}_{\tilde{\mathbf{y}} \sim \pi(\cdot; \boldsymbol{\beta}(\mathbf{x}, \mathbf{y}'))} [r(\tilde{\mathbf{y}})] \tag{12}$$

The minimization of the negative expected reward $\mathcal{L}_{\text{car}}$ encourages the policy to generate pseudo summaries that are similar to the ground-truth monolingual summaries. Given that the policy is parameterized by the CLS awareness vector $\boldsymbol{\beta}$, this minimization process also increases the awareness of tokens that appear in the ground-truth monolingual summary. We refer to this process as cross-attention reinforcement. Due to the non-differentiable nature of reward computation, we estimate the gradient of $\mathcal{L}_{\text{car}}$ with respect to $\boldsymbol{\beta}$ using the policy gradient theorem (Williams 1992), as depicted in Equation 13. Here, $\tilde{\mathbf{y}}$ is sampled from the current pseudo-summary policy. The gradient with respect to other parts of the model can be further computed following the chain rule.

$$\nabla_{\boldsymbol{\beta}} \mathcal{L}_{\text{car}} \approx - \sum_{(\mathbf{x}, \mathbf{y}') \in \mathcal{D}} r(\tilde{\mathbf{y}}) \nabla_{\boldsymbol{\beta}} \log \pi(\tilde{\mathbf{y}}; \boldsymbol{\beta}) \tag{13}$$

Self-Critical Baseline

Self-critical training, initially introduced in (Rennie et al. 2017) for image caption generation, addresses the issue of policy gradient fluctuations by incorporating a baseline into the reward computation. In contrast to conventional "REINFORCE with baseline" algorithms, which involve separate sampling and training processes to estimate the baseline,

self-critical training simplifies the procedure by using the reward of the greedy action as the baseline. This approach allows for more stable and effective policy gradient estimation.

To define a self-critical baseline, we first establish the notion of the greedy action. In the case where the policy outputs a multinomial distribution, also known as a categorical distribution, the greedy action corresponds to selecting the action with the highest probability. In our pseudo-summary policy, we define the greedy action as choosing the top K tokens with the highest awareness scores from the source document, where K represents the length of the ground-truth monolingual summary. The resulting greedy summary can be represented by Equation 14, where $\rho(\boldsymbol{\beta}, i)$ denotes the index of the i-th largest element in $\boldsymbol{\beta}$. The yellow dotted square in Figure 1 denotes the process of taking the greedy action $\bar{\mathbf{y}}$ under the policy parameterized by $\boldsymbol{\beta}$.

$$\bar{\mathbf{y}} = (x_{\rho(\boldsymbol{\beta},1)}, x_{\rho(\boldsymbol{\beta},2)}, \cdots, x_{\rho(\boldsymbol{\beta},K)}) \tag{14}$$

The self-critical baseline is defined as the reward achieved after taking the greedy action, i.e., $r(\bar{\mathbf{y}})$. We can now subtract the self-critical baseline from the original training objective and obtain a new training objective, as depicted in Equation 15. The black dashed arrow in Figure 1 illustrates the process of backpropagating the gradient $\nabla_{\boldsymbol{\beta}} \mathcal{L}_{\text{sc}}$.

$$\mathcal{L}_{\text{sc}} = - \sum_{(\mathbf{x},\mathbf{y}') \in \mathcal{D}} \mathbb{E}_{\tilde{\mathbf{y}} \sim \pi(\cdot;\boldsymbol{\beta})} \left[r(\tilde{\mathbf{y}}) - r(\bar{\mathbf{y}}) \right] \tag{15}$$

Training and Inference

The final training objective is to minimize the cross-entropy loss and the self-critical policy loss jointly, as shown in Equation 16, where λ is a hyperparameter balancing the two loss items. Since the CAR module does not introduce any extra model parameters, the increase in spatial training cost can be ignored. The increase in temporal training cost mainly depends on the computation of the reward, which cannot be done in parallel on GPU.

$$\mathcal{L} = \mathcal{L}_{\text{xe}} + \lambda \mathcal{L}_{\text{sc}} \tag{16}$$

During inference, the CAR module is disabled, i.e., only the backbone transformer model is used to generate the summary. Therefore, the CAR module does not increase the inference cost.

Experiments

Dataset

We use WikiLingua (Ladhak et al. 2020), GlobalVoice (Nguyen and Daumé III 2019), and CrossSum (Bhattacharjee et al. 2021) for training and evaluation. For WikiLingua, we directly use the parallel monolingual summaries for cross-attention reinforcement. However, for GlobalVoice and CrossSum, there is only the cross-lingual summary without the corresponding parallel monolingual summary for each article. Therefore, we use an existing open-source mT5 model [1] that is fine-tuned on the XLSum dataset (Hasan et al. 2021) to generate a parallel monolingual summary for each article. We would like to see the performance of CAR-Transformer on morphologically different languages. For WikiLingua, we choose Korean (ko-en), Hindi (hi-en), Czech (cs-en), Chinese (zh-en), Thai (th-en), and Turkish (tr-en) as the source article language and English as the target summary language. For GlobalVoice and CrossSum, we choose French (fr-en) and Arabic (ar-en).

[1] https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum

Baselines

Our proposed approach referred to as **CAR** (Cross-Attention Reinforcement), serves as the focal point of our comparative analysis against several baselines. To establish a baseline, we initially employ the vanilla mBART fine-tuning approach (referred to as **mBART**), where we directly fine-tune the mBART model on each CLS task without incorporating auxiliary training objectives. By contrasting the performance of CAR with the mBART baseline, we can effectively evaluate the extent to which the CAR module and its associated training objective contribute to enhancing the mBART model's performance on CLS tasks.

Furthermore, we extend our evaluation to include baselines that adopt multi-task training strategies. Inspired by the work of Zhu et al. 2019, we incorporate **CLS-MS** and **CLS-MT** as additional baselines. These two baselines utilize an additional decoder to handle the Monolingual Summarization (MS) task and the Machine Translation (MT) task. The CLS-MS means jointly performing the CLS task and the MS task, while CLS-MT means jointly performing the CLS task and the MT task. For GlobalVoice and CrossSum, we do not train or evaluate CLS-MT since there are no parallel source articles to be used for the MT task. Additionally, we adopt the **2-step** baseline proposed by Ladhak et al. 2020, which involves a 2-step fine-tuning process of mBART, starting with the MS task followed by the CLS task.

Experiment Setting

We leverage the mBART50-large model for the initialization of our approach and all baseline approaches. Then, we conduct fine-tuning procedures using the aforementioned CLS datasets. We truncate the source document to 512 tokens as input for the encoder, while the ground-truth summary in the target language, serving as input for the decoder, is truncated to 128 tokens. Similarly, the supervision signal for the CAR module, which comprises the ground-truth summary in the source language, is also truncated to 128 tokens.

We fine-tune our approach and all baseline approaches on each CLS task for a total of 30 epochs utilizing the training set. After each epoch of training, we evaluate the models using the validation set and record the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. We save the model with the highest sum of ROUGE scores and subsequently evaluate this saved model on the test set. The training and evaluation procedures for each task are performed on a single NVIDIA A40 GPU. With a training batch size of 8, we employ a gradient accumulation step of 2. The training and evaluation codes are implemented based on HuggingFace Transform-

	ko-en			*hi-en*			*cs-en*		
mBART	27.66	8.19	22.36	30.50	9.71	24.07	25.82	6.84	20.57
CLS-MS	30.56†	9.57†	24.35†	30.09	10.25†	25.17†	25.37	6.87	20.33
CLS-MT	25.21	7.39	20.60	30.70	9.79	24.05	25.32	7.05	20.51
2-step	31.04†	9.74†	24.48†	**31.27**†	9.84	24.28	26.21	7.40†	21.96†
CAR (Ours)	**31.17**†	**10.53**†	**26.03**†	30.66	**10.46**†	**25.32**†	**27.47**†	**8.00**†	**22.46**†

	zh-en			*th-en*			*tr-en*		
mBART	30.64	9.32	24.70	31.47	10.27	25.24	35.41	13.11	27.94
CLS-MS	30.56	9.37	24.83	22.56	6.07	19.06	35.45	13.30	28.21
CLS-MT	30.16	9.07	24.47	32.05†	11.15†	26.85†	34.84	12.91	27.62
2-step	30.58	9.32	24.75	**32.38**†	10.92†	26.21†	35.61	13.21	28.07
CAR (Ours)	**31.28**†	**9.56**	**25.31**†	32.33†	**11.20**†	**27.00**†	**36.15**†	**14.01**†	**29.11**†

Table 1: ROUGE-1/2/L F1 scores of our approach and baseline approaches on the WikiLingua CLS tasks. The † superscript means a significant performance improvement (more than 0.4) compared with the vanilla mBART fine-tuning approach.

	GlobalVoice						CrossSum					
	ar-en			*fr-en*			*ar-en*			*fr-en*		
mBART	24.59	6.01	18.46	29.44	9.30	22.19	28.05	6.53	21.26	29.57	9.06	22.25
CLS-MS	24.26	5.61	18.48	29.52	9.38	22.15	28.17	6.60	21.35	29.69	9.15	22.26
2-step	25.19†	6.16	19.00†	29.20	9.59	22.31	28.15	6.58	21.27	29.67	9.20	22.45
CAR (Ours)	**25.42**†	**6.41**†	**19.10**†	**30.01**†	**9.77**†	**22.68**†	**28.20**	**6.79**	**21.37**	**29.80**	**9.34**	**22.54**

Table 2: ROUGE-1/2/L F1 scores on the GlobalVoice and CrossSum tasks.

	ar-en			*fr-en*		
CAR	25.42	6.41	19.10	30.01	9.77	22.68
CAR w/ CLS-MS	23.34	5.00	18.12	28.56	8.93	21.78
CAR w/ 2-step	24.96	**6.60**	**19.27**	29.96	**9.95**	**22.99**

Table 3: ROUGE-1/2/L scores of hybrid approaches on GlobalVoice dataset.

	mBART	CLS-MS	CLS-MT	2-step	CAR
Steps	1.0	1.0	1.0	2.0	1.0
GPU Memory	1.0	1.3	1.3	1.0	1.0
GPU Time	1.0	1.2	1.2	2.0	1.1
Total Time	1.0	1.2	1.2	2.0	1.3

Table 4: The training costs relative to mBART. The reported training costs of each approach is averaged on 10 times of training.

ers [2]. For the detailed experiment setting and implementation, please refer to the source code in supplementary files.

Result Analysis

The recorded ROUGE-1/2/L F1 scores for the baseline approaches and our proposed approach are presented in Table 1 and 2. We highlight the best ROUGE scores with the bold font and the significant improvement with the † superscript. Except for the CrossSum dataset, our approach achieves significant improvements on all other considered CLS tasks. Although the baseline multi-task approaches can demonstrate significant improvements on some tasks, they fall short of delivering consistent improvement across the entire set of tasks. For example, on the WikiLingua dataset, the CLS-MS approach does not achieve significant improvements on the cs-en and tr-en sub-tasks, and the 2-step approach does not achieve significant improvements on the tr-en sub-task. On the GlobalVoice dataset, the 2-step approach does not achieve significant improvements on the fr-en sub-task. Note that our approach does not achieve significant improvements on the CrossSum dataset, but it outperforms all other approaches on this dataset. On most CLS tasks, our approach even outperforms the 2-step approach, which requires 2x training steps as other approaches. The relative training costs of all approaches are illustrated in Table 4. Note that the total time of our approach is slightly higher than mBART and the multi-task approaches because training CAR-Transformer requires reward computation, which involves computing a ROUGE score and can only be done on the CPU.

The convergence of our approach, the vanilla mBART, and the best-performing multi-task training approach for the subtasks in WikiLingua is depicted in Figure 2. All curves represent the average score of all subtasks. Notably, our approach consistently exhibits the highest ROUGE-L score convergence compared to both the vanilla mBART and the best-performing multi-task training approach (MT), which reinforces the effectiveness and robustness of our approach.

[2] https://github.com/huggingface/transformers

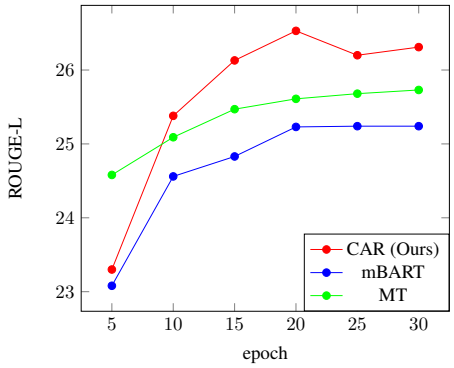

Figure 2: Validation ROUGE-L scores along training epochs on WikiLingua validation set.

Approach A	A >B	A = B	A <B	Approach B
CLS-MS	17923	13995	15265	mBART
CLS-MT	18213	13858	15112	mBART
2-step	18786	13278	15119	mBART
CAR (Ours)	18912	13228	15043	mBART

Table 5: GSB evaluation by ChatGPT. The value in the table denotes the number of samples.

Table 3 shows the results achieved by CAR and three hybrid approaches, where **CAR w/ CLS-MS** and **CAR w/ 2-step** denote the mix of CAR with the CLS-MS and 2-step multi-task approaches, respectively. The result shows that incorporating the CLS-MS approach harms the CLS performance, while incorporating the 2-step approach further improves the CLS performance in ROUGE-2 and ROUGE-L scores.

Moreover, we realize that ROUGE may not reflect the human-level quality. Therefore, we employ ChatGPT to make a GSB (Good-Same-Bad) evaluation on the 47183 test set samples of the six subtasks of WikiLingua. Specifically, let ChatGPT rank the summaries generated by 5 approaches (out of order) from best to worst with an inequality (e.g., 3>4=2>1>5). Then, we collect all inequalities generated by ChatGPT and count the good, same, and bad samples between mBART and other four approaches. The GSB evaluation results are shown in Table 5.

Interpretability

We check the interpretability of our approach by visualizing the cross-attention weight matrix of the fine-tuned CLS-MS and CAR-Transformer. As shown in Figure 3, the instance summarizes the source document "Tato metoda se vám bude hodit zejména pokud jste venku a šlápnete na ještě měkkou žvýkačku. Budete potřebovat pouze trochu písku a klacík." in Czech into "Find a wooden stick and some dry sand." in English. The upper part is the cross-attention weight matrix of the fine-tuned CLS-MS without cross-attention reinforcement, while the lower part is the cross-attention weight matrix of the fine-tuned CAR-Transformer. The red squares include the source sentence *Budete potřebovat pouze trochu písku a klacík*, which means *All you need is some sand and a stick* in English. We emphasize this sentence since it is highly relevant to the English summary.

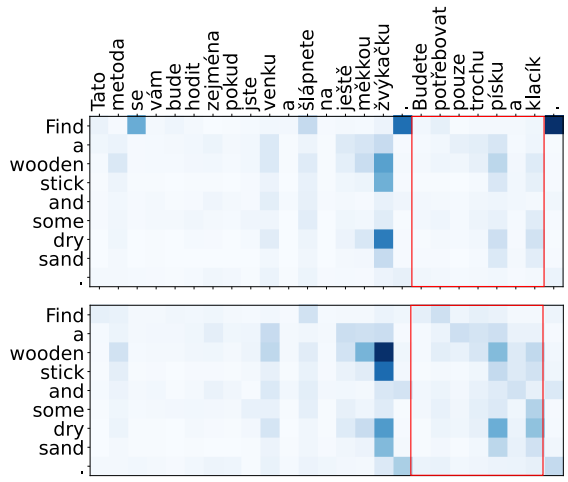

Figure 3: Visualization of CLS-MS and CAR-Transformer cross-attention weight matrix fine-tuned on the Czech-to-English CLS task of WikiLingua.

We can conclude from Figure 3 that the blocks in the lower red square have darker colors than blocks in the upper red square, which means that the target sequence in CAR-Transformer has higher attention to the part of the source sequence that is highly relevant to the summary.

Figure 3 only presents the interpretability from a qualitative perspective. We also provide a quantitative perspective of the interpretability. We define the Key Awareness Ratio (KAR) in Eq. 17, where $\mathcal{I}_K$ is the set of positions of the key content in the source article and S is the length of the source article. We compute the KAR for 5 approaches in summarizing the article in Figure 3, and the KAR of mBART, CLS-MS, CLS-MT, 2-step, and CAR (Ours) are: 8.51%, 8.43%, 8.64%, 8.32%, and 9.02%, where our approach achieves the best KAR.

$$\text{KAR} = \frac{\sum_{i \in \mathcal{I}_K} \beta_i}{\sum_{i=1}^{S} \beta_i} \tag{17}$$

Conclusion

We present Cross-Attention Reinforcement (CAR) and the CAR-Transformer, which applies the policy gradient method commonly used in reinforcement learning to the cross-attention weights of Transformer-based encoder-decoder models during fine-tuning. The formulation of the pseudo-summary policy provides a novel idea of applying policy gradient to cross-lingual summarization tasks. Our experiments show that CAR better improves the performance on CLS tasks than conventional multi-task training approaches.

Acknowledgments

This work has been greatly supported by the National Natural Science Foundation of China (No. 91118002). The quality of this paper has been greatly improved thanks to the insightful comments of anonymous reviewers.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bai, Y.; Gao, Y.; and Huang, H.-Y. 2021. Cross-Lingual Abstractive Summarization with Limited Parallel Resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6910–6924.

Bhattacharjee, A.; Hasan, T.; Uddin Ahmad, W.; Li, Y.-F.; Kang, Y.-B.; and Shahriyar, R. 2021. CrossSum: Beyond English-Centric Cross-Lingual Abstractive Text Summarization for 1500+ Language Pairs. *arXiv e-prints*, arXiv–2112.

Cao, Y.; Liu, H.; and Wan, X. 2020. Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 6220–6231.

Chi, Z.; Dong, L.; Wei, F.; Wang, W.; Mao, X.-L.; and Huang, H. 2020. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7570–7577.

Duan, X.; Yin, M.; Zhang, M.; Chen, B.; and Luo, W. 2019. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3162–3172.

Hasan, T.; Bhattacharjee, A.; Islam, M. S.; Mubasshir, K.; Li, Y.-F.; Kang, Y.-B.; Rahman, M. S.; and Shahriyar, R. 2021. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4693–4703.

Ladhak, F.; Durmus, E.; Cardie, C.; and Mckeown, K. 2020. WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4034–4048.

Leuski, A.; Lin, C.-Y.; Zhou, L.; Germann, U.; Och, F. J.; and Hovy, E. 2003. Cross-lingual c* st* rd: English access to hindi information. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3): 245–269.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Liang, Y.; Meng, F.; Zhou, C.; Xu, J.; Chen, Y.; Su, J.; and Zhou, J. 2022. A Variational Hierarchical Model for Neural Cross-Lingual Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2088–2099.

Lim, J.-M.; Kang, I.-S.; and Lee, J.-H. 2004. Multi-Document Summarization Using Cross-Language Texts. In *NTCIR*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8: 726–742.

Ma, S.; Dong, L.; Huang, S.; Zhang, D.; Muzio, A.; Singhal, S.; Hassan Awadalla, H.; Song, X.; and Wei, F. 2021. DeltaLM: Encoder-Decoder Pre-training for Language Generation and Translation by Augmenting Pretrained Multilingual Encoders. *arXiv e-prints*, arXiv–2106.

Nguyen, K.; and Daumé III, H. 2019. Global Voices: Crossing Borders in Automatic News Summarization. *EMNLP-IJCNLP 2019*, 90.

Orăsan, C.; and Chiorean, O. A. 2008. Evaluation of a cross-lingual romanian-english multi-document summariser.

Perez-Beltrachini, L.; and Lapata, M. 2021. Models and Datasets for Cross-Lingual Summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9408–9423.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7008–7024.

Takase, S.; and Okazaki, N. 2022. Multi-Task Learning for Cross-Lingual Abstractive Summarization. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 3008–3016.

Tang, Y.; Tran, C.; Li, X.; Chen, P.-J.; Goyal, N.; Chaudhary, V.; Gu, J.; and Fan, A. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3450–3466.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wan, X.; Li, H.; and Xiao, J. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 917–926.

Wang, J.; Meng, F.; Zheng, D.; Liang, Y.; Li, Z.; Qu, J.; and Zhou, J. 2022. A survey on cross-lingual summarization. *Transactions of the Association for Computational Linguistics*, 10: 1304–1323.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 5–32.

Xu, R.; Zhu, C.; Shi, Y.; Zeng, M.; and Huang, X. 2020. Mixed-Lingual Pre-training for Cross-lingual Summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 536–541.

Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; and Raffel, C. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498.

Zhu, J.; Wang, Q.; Wang, Y.; Zhou, Y.; Zhang, J.; Wang, S.; and Zong, C. 2019. NCLS: Neural Cross-Lingual Summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3054–3064.