Global Learning with Triplet Relations in Abstractive Summarization

Fengyu Lu[1], Jiaxin Duan[1]*, Junfei Liu[2]
[1]School of Software and Microelectronics, Peking University, Beijing, China
[2]National Engineering Research Center for Software Engineering, Peking University, Beijing, China
{fengyul,duanjx}@stu.pku.edu.cn, liujunfei@pku.edu.cn

Abstract

Abstractive summarization models learned with token-level maximum likelihood estimation suffer from exposure bias, that the condition for predicting the next token is discrepant during training and inference. Existing solutions bridge this gap by learning to estimate semantic or lexical qualities of a candidate summary from the global view, namely global learning (GL), yet ignore maintaining rational triplet-relations among document, reference summary, and candidate summaries, e.g., the candidate and reference summaries should have a similar faithfulness degree judging by a source document. In this paper, we propose an iterative autoregressive summarization paradigm - IARSum, which fuses the learning of triplet relations into a GL framework and further enhances summarization performance. Specifically, IARSum develops a dual-encoder network to enable the simultaneous input of a document and its candidate (or reference) summary. On this basis, it learns to 1) model the relative semantics defined over tuples (candidate, document) and (reference, document) respectively and balance them; 2) reduce lexical differences between candidate and reference summaries. Furthermore, IARSum iteratively reprocesses a generated candidate at inference time to ground higher quality. We conduct extensive experiments on two widely used datasets to test our method, and IARSum shows the new or matched state-of-the-art on diverse metrics.

1 Introduction

Abstractive summarization is a classical natural language generation (NLG) task, which aims to rewrite a long document into a shorter version, retaining only the salient information (Kumar and Chakkaravarthy, 2023; Xie et al., 2023). In recent years, the advancement of pre-trained language models (PLMs) (Lewis et al., 2020; Zhang et al., 2020a) founded on large-scale corpora boosted abstractive summarization significantly, and sequence-to-sequence (Seq2Seq) learning has shown promising results in almost all scenarios. It commonly learns an autoregressive PLM with maximum likelihood estimation (MLE), and the teacher-forcing algorithm (Goyal et al., 2016) is together used to ensure training efficiency and stability. However, such a model predicts each token in a summary based on the gold pre-context during training but on its preceding outputs at inference, causing a training-inference discrepancy - *exposure bias* (Bengio et al., 2015; Goodman et al., 2020), which heavily limits summarization performance.

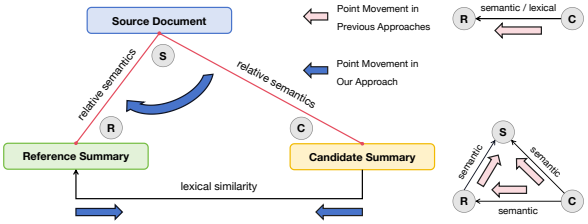

Figure 1: Graphicalization of triplet relations among a document, the reference summary, and a candidate summary, where each point of the triangle $\triangle SRC$ presents source document (S), reference summary (R), and candidate summary (C), respectively. **Upper right**: Traditional GL methods only consider the edge RC and deem it the semantic or lexical gaps between the points R and C, which they learn to minimize. **Lower right**: SeqCo further considers S and regards each edge as a semantic gap to be minimized. **Left**: We extend traditional GL methods by treating the side edge SR (SC) as a relative semantics metric, and we highlight the balance of edges SR and SC.

Since exposure bias happens on the token level, existing solutions train models to maximize the global similarity between candidate and reference summaries, namely *global learning* (GL). Reinforcement and contrastive learning in summarization are the most used GL technologies. For example, in reinforcement learning based GOLD (Pang

*Corresponding author.

Proceedings of the 28th Conference on Computational Natural Language Learning, pages 198–208

and He, 2021) and RLEF (Roit et al., 2023), a summarization model is rewarded depending on the quality of candidate summaries it produces, with the reference as the standard. Similarly, contrastive learning methods (Liu et al., 2022; Xie et al., 2023; Zhang et al., 2022) compare candidate summaries with the reference and assign the one closer to the reference a higher probability, and vice versa. On the one hand, all these methods measure candidate-reference similarities without considering the source document conditions. Furthermore, their measurement stands on only the semantic or lexical aspect rather than comprehensive perspectives, resulting in biased learning objectives. SeqCo (Xu et al., 2022) aims to minimize the semantic discrepancies among a source document, its reference, and candidate summaries. Still, it is powerless to learn lexical perception and shows undesirable summarization results.

To address these problems, we highlight rational relations within the triplet (document, candidate summary, reference summary) in abstractive summarization. Take the geometrical triplet in Figure 1 for intuitive perception. Traditional GL methods view the edge RC as lexical or semantic gaps between candidate and reference summaries, aiming to draw points R and close as possible. Similarly, SeqCo considers from the semantics perspective and further aims to condense the triangle $\triangle SRC$ into a single point (draw all three points S, R, and C as close as possible). In this work, we base on the traditional GL assumption and further treat side edges SR and SC as relative semantics metrics between the document and summaries. We propose to balance the edges SR and SC while minimizing the edge RC. Our inspiration is that an ideal model should generate candidate summaries similar to the reference on both semantic and lexical aspects (Sul and Choi, 2023). In particular, the limitations of GL-based methods, which tend to yield summaries with unsatisfactory relative semantics, such as faithfulness and abstractiveness (Dixit et al., 2023) measured by the source document, can be effectively fixed by balancing the two side edges of $\triangle SRC$.

According to the above insights, we propose an iterative autoregressive summarization paradigm (IARSum), which facilitates learning the mentioned triplet relations with a standard GL framework to enhance summarization performance. Specifically, IARSum generates a summary through a series of iterations, during which the model re-inputs and reprocesses the previously generated summary in each iteration to get improved versions. This encourages assessing summaries' quality from a global view and effectively prevents exposure bias. We build IARSum on a double encoder-decoder network following Transformer architecture to fulfill the desired properties. It uses two serial encoders to encode the document and reinput summaries, respectively, and uses the second encoder's outputs to model summary-document semantics. To learn the IARSum model aware of triplet relations, we reward the model to get similar outputs from the second encoder when provided with candidate and reference summaries as input, respectively. We also reward the model once a candidate achieves higher lexical overlap with the reference after reprocessing. Furthermore, we adopt an offline mini-risk training strategy that enforces the model to maximize the mentioned rewards. In inference, a trained IARSum model can adaptively refine the generated summaries in sequential iterations for increased quality.

In summary, we make three-fold contributions. First, we explore rational relations within the triplet (source document, reference summary, candidate summary) in summarization and propose to balance the relative semantics over tuples (candidate, document) and (reference, document) while reducing the lexical differences within (candidate, reference). Second, we propose IARSum, a novel summarization paradigm that facilitates learning our suggested triplet relations with a GL framework to boost summaries' quality. Finally, we conduct extensive experiments on two public datasets to test our methods. Results show that IARSum matches or outperforms previous state-of-the-art (SOTA) approaches in generating high-quality summaries measured by multiple metrics. Furthermore, we transfer IARSum to few-shot settings and show its superior robustness.

2 Related Work and Background

2.1 Abstractive Summarization

Summarization is always modeled as a Seq2Seq generation task, creating function f that is conditioned on a source document X to output a target summary Y:

$$Y \leftarrow f(X) \tag{1}$$

For the abstractive paradigm, existing approaches commonly learn an autoregressive language model with parameter θ to fit f and approximate the

conditional probability $\mathrm{P}(Y|X)$ token by token. Maximum likelihood estimation (MLE) is the most used learning schema. It aims to maximize the probability that the model predicts gold reference, following independent and identically distributed conditions, i.e., $\max_\theta \mathrm{P}_\theta(Y|X) = \max_\theta \prod_{t=1}^{l} \mathrm{P}_\theta(y_t|Y_{<t}, X)$, where l denotes the length of reference and $Y_{<t}$ refers to sub-sequence $\{y_1, y_2, \cdots, y_{t-1}\}$.

During training, the teacher-forcing mechanism (Goyal et al., 2016) is adopted, which conditions on exact pre-context to predict a target token and minimizes the following negative log-likelihood (NLL) loss:

$$\mathcal{L}_{nll}(\theta) = -\sum_{t=1}^{l} \log \mathrm{P}_\theta(y_t|Y_{<t}, X) \qquad (2)$$

Though this encourages stable MLE learning, such a trained model depends heavily on accurate prediction. Intuitively, it learns to sample the next token at timestep t from the distribution $\mathrm{P}(\cdot|Y_{<t}, X)$, while the case at inference is to sample from $\mathrm{P}(\cdot|Y'_{<t}, X)$, where $Y'_{<t}$ denotes the previous generation. This gap between training and inference is the so-called *exposure bias*, causing errors accumulation during inference, especially once any improper token is generated in early steps.

2.2 Global Learning

Reinforcement learning (RL) rewards a model with sequence-level feedback, depending on varying evaluation metrics. Most works (Tan, 2023; Roit et al., 2023) are based on on-policy learning (Paulus et al., 2018), where a model generates a sampled candidate and a greedily searched candidate during training. It requires high computational costs and tends to get stuck in a zero-reward region. As a result, MLE loss is used as an assistant. Richard et al. (Pang and He, 2021) proposed an off-policy learning method that uses reference summary as a demonstrator. Although it averts zero rewards, the exploring ability is reduced.

Traditional contrastive learning (CTL) uses positive and negative sample pairs to train a model to distinguish real data labels. For example, CLIFF (Cao and Wang, 2021) builds sample pairs by the back-translation and improves the faithfulness and factuality of the generated summaries. In recent years, ranking-based learning originated from the standard CTL and has shown advanced performance in abstractive summarization. Liu et al. (Liu and Liu, 2021) first propose a two-stage framework that trains a RoBERTa (Liu et al., 2019) to rank the candidates generated by BART at first. BRIO (Liu et al., 2022) makes a further optimization, trains BART itself as an evaluation tool, and ranks the conditional probability of candidates. Later, a lot of improved BRIO variants (Xie et al., 2023; Zhao et al., 2023; Zhang et al., 2022) were proposed in succession. Despite performing surprisingly, such methods only focus on maximizing the candidate-reference similarity without considering the source document effects. Noting this point, SeqCo (Xu et al., 2022) contrasts semantics among source documents, candidates, and references. However, SeqCo assumed irrational triplet relations and suffered unstable optimization caused by online learning, the main reason for undesirable performance.

3 Method

In this section, we describe the details of our proposed methods. We introduce the iterative autoregressive text generation paradigm in Section 3.1, describe the IARSum model architecture in Section 3.2, and illustrate the offline global learning strategy in Section 3.3.

3.1 Iterative Autoregressive Generation

The standard autoregressive (AR) text generation illustrated in Figure 2 (a) is widely known as a unidirectional process, where a text is generated sequentially token by token. The major limitation is that each token is predicted depending on its pre-context. As a result, a wrongly generated token may mislead the later content and make the generated text entirely deviate from the target due to error accumulation. Calibrating the predicted token distributions on a global view (global learning) is effective in addressing this problem. However, it is hard to involve the source document conditions, i.e., the semantic relations between the document and summaries, in calibration. We propose an iterative autoregressive generation paradigm (IAR) to break these limitations.

As demonstrated in Figure 2, IAR models the generation of target text in the Seq2Seq task as a text-level Markov Chain, where the state transitions from a draft to more refined results. Taking abstractive summarization as an example, at the i-th iteration, IARSum samples a candidate summary Y^{i-1} from the previous iteration's outputs,

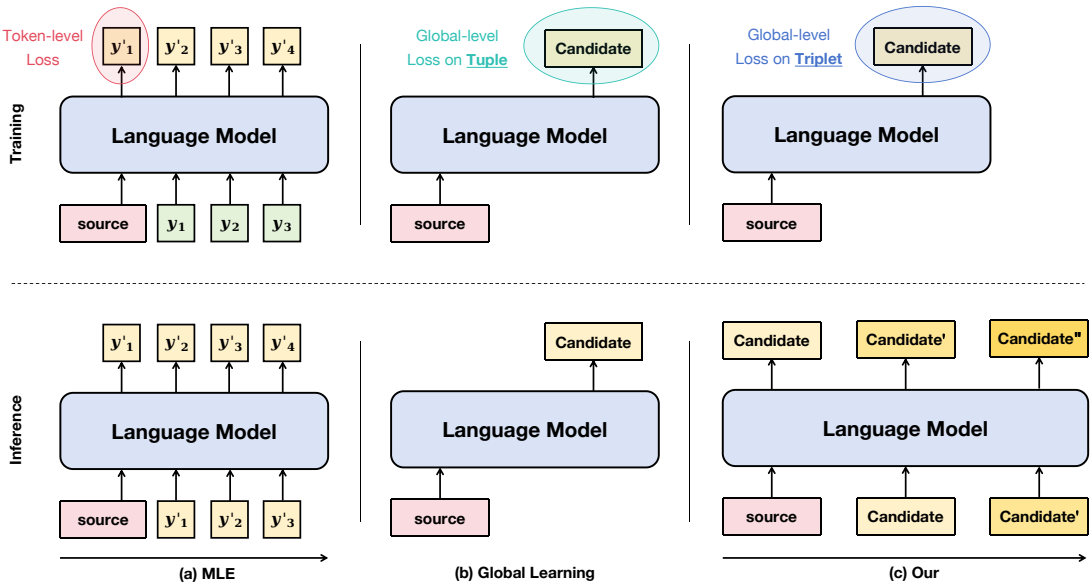

Figure 2: Comparison of different learning paradigms. (a) MLE trains a model to minimize token-level loss. (b) Traditional global learning trains a model to minimize global-level loss defined over the candidate and reference summaries. (c) Our method trains a model to minimize global-level loss defined over the triplet built upon a document, candidate summary, and reference summary.

estimates its quality, and produces a new one Y^i. This process repeats until the maximum number of iterations N is reached, which formally presents as:

$$\begin{aligned} \mathrm{P}(Y|X) &= \mathrm{P}\left(Y^0|X\right)\mathrm{P}\left(Y^N|Y^0,X\right) \\ &= \mathrm{P}\left(Y^0|X\right)\prod_{i=1}^{N}\mathrm{P}\left(Y^i|Y^{i-1},X\right) \end{aligned} \tag{3}$$

where $Y^0 \sim \mathrm{P}(\cdot|X)$, $Y^i \sim \mathrm{P}(\cdot|X,Y^{i-1})$, and $Y^N = Y$.

3.2 Model Architecture

We implement IARSum with a Transformer-based encoder-decoder model shown in Figure 3. It has two encoders with bidirectional attention and one decoder with unidirectional attention. To speed up convergence, we start our model with a single-encoder Transformer with pre-trained parameters θ and share initial parameters between the two encoders.

The architecture of IARSum is very similar to GSum (Dou et al., 2021). However, encoders of GSum are independent and connect to the decoder orderly by cross-attention layers (i.e., parallel encoders). IARSum instead adapts serial encoders, where the second encoder relies on the output of the first to feature the input content, similar to a Transformer decoder without a sequence mask. Besides, GSum uses the second encoder to encode guidance words, while IARSum's second encoder is used to encode the candidate summary. Mathematically, IARSum models the following token distributions during the first and later iterations, respectively:

$$\begin{aligned} \mathrm{P}_\theta(y_t^0|X) &= \sigma(D_\theta(E_\theta^1(X), y_{<t}^0)) \\ \mathrm{P}_\theta(y_t^i|Y^{i-1},X) &= \sigma(D_\theta(E_\theta^2(E_\theta^1(X), Y^{i-1}), y_{<t}^i)) \end{aligned} \tag{4}$$

where E_θ^1, E_θ^2 are the first and second encoders, and D_θ is the decoder. $\sigma(\cdot)$ is softmax function.

3.3 Learning Objective

According to our intention proposing IAR in Section 3.1, we learn an IARSum model for mainly two objectives. One is to maximize the lexical similarity between candidate and reference summaries, and the other one involves matching the relative semantics of candidates with that of the reference, taking the document as the standard. Both objectives can be attended within a multi-rewards learning framework.

Semantics Rewards. The recent study (Dreyer et al., 2023) pointed out that a summary should be logically entailed in the source document to ensure faithfulness. On the other hand, researchers also observed that humans write summaries with hallucinatory words to keep abstractiveness (Maynez et al., 2020) despite contradicting summary-document entailment. Note that faithfulness and abstractiveness are perceived on the source document basis. To

bypass their contradictions, we uniformly refer to such perceptions as relative semantics and model them with a neural function $\mathcal{S}(\cdot,\cdot)$. Intuitively, an ideal candidate summary should be at a similar level of relative semantics compared with the reference, and their differences in this attribute are quantitatively observed:

$$M_s(X,Y,Y^i) = \langle \mathcal{S}(X,Y), \mathcal{S}(X,Y^i) \rangle \quad (5)$$

where $\langle \cdot,\cdot \rangle$ is a distance function, such as Euclidean distance. Furthermore, we will reward the model according to the gain of semantics rewards between two adjacent iterations:

$$R_s(Y^i, Y^{i-1}) = M_s(X,Y,Y^i) - M_s(X,Y,Y^{i-1}) \quad (6)$$

Lexical Rewards. From the lexical view, we encourage a candidate, after reprocessing, to contain more tokens overlapped with the reference. To this end, we reward the model using the lexical rewards:

$$R_l(Y^i, Y^{i-1}) = \frac{|(\{Y^i\} - \{Y^{i-1}\}) \cap \{Y\}|}{|\{Y\}|} \quad (7)$$

where $\{\cdot\}$ denotes a token set and $|\cdot|$ means the set size.

Learning Objective. Finally, we mix the two types of rewards with a balance coefficient $\xi \in (0,1)$:

$$R(Y^i) = \xi R_s(Y^i, Y^{i-1}) + (1-\xi) R_l(Y^i, Y^{i-1}) \quad (8)$$

and the overall learning objective for IARSum is unified to maximize the following expected rewards:

$$\sum_{i=1}^{N} \max_{\theta} \mathrm{E}_{Y^i \sim \mathrm{P}_\theta(\cdot|Y^{i-1},X)}[R(Y^i, Y^{i-1})]. \quad (9)$$

3.4 Training

As we learn IARSum to maximize the expected rewards, the infinite sampling space makes the expectation in Eq.9 untraceable. Predominant studies commonly use the Monte Carlo approach to address this problem, which approximates the real distribution with empirical samples. We follow this idea and adopt a minimum-risk training (Shen et al., 2016) strategy. At each iteration i, we sample k candidates $Y^i_{(1)}, \cdots, Y^i_{(k)}$ from $\mathrm{P}_\theta(\cdot|Y^{i-1}, X)$ using beam-search (Vijayakumar et al., 2016), and the model is trained to minimize an expected risk loss:

$$\mathcal{L}_{er}(\theta) = -\sum_{t=1}^{k} R(Y^i_{(t)}) \frac{\mathrm{P}_\theta(Y^i_{(t)}|Y^{i-1}, X)}{\sum_{t=1}^{k} \mathrm{P}_\theta(Y^i_{(t)}|Y^{i-1}, X)} \quad (10)$$

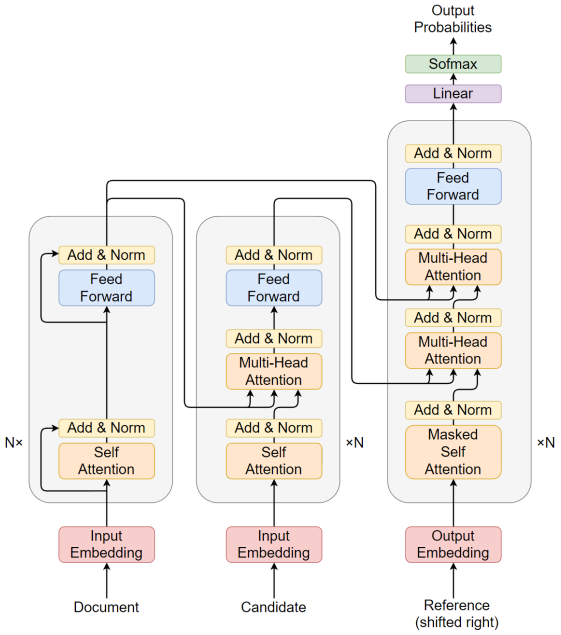

Figure 3: IARSum model's dual-encoder architecture.

Semantic Relation Modeling. Another challenge we encounter is the implementation of function $\mathcal{S}$. Given a document-summary pair (X,Y), we average the output of the IARSum second encoder to feature their semantic relations:

$$\mathcal{S}(X,Y;\theta) = MeanPool(E^2_\theta(E^1_\theta(X), Y)) \quad (11)$$

This approach is parameter-efficient. However, dynamically learned parameters θ cause the observation of $\mathcal{S}$ to vary sharply as training progresses. Drawing from (Zhang et al., 2022) lessons, we introduce momentum-based parameterization to remove this risk. Concretely, we build ζ-parameterized $\mathcal{S}(\cdot,\cdot;\zeta)$, which is initialized by θ and updated with the moving average:

$$\zeta \leftarrow \mu\zeta + (1-\mu)\theta \quad (12)$$

where μ is a momentum coefficient to coordinate the synchronization rate of two types of parameters. Based on this, Eq.5 is reformulated as:

$$M_s(X,Y,Y^i) = -\|\mathcal{S}(X,Y;\zeta) - \mathcal{S}(X,Y^i;\zeta)\| \quad (13)$$

Offline Learning. We use offline samples during training to save the computational costs of generating candidates. A pre-trained model is first fine-tuned with MLE and proceeds to generate k candidate summaries for every document in the training set. Each candidate, coupled with the source document, forms a $\{X, Y^{i-1}\}$ pair used for the model

training. Note that the generated candidates share varying semantic and lexical qualities, and the ones closer to the reference standard simulate the drafts that have been reprocessed more times. This nature facilitates the training to focus on only one iteration without considering the multi-turn rewards. Moreover, training the model to maximize expected rewards alone is unguaranteed to generate fluent language. Following (Liu et al., 2022; Zhao et al., 2023), we add a regularization term in Eq.9, and the overall loss function is then:

$$\mathcal{L}(\theta) = \mathcal{L}_{nll}(\theta) + \lambda \mathcal{L}_{er}(\theta) \quad (14)$$

4 Experiments

4.1 Datasets

Two public open-domain datasets are used to evaluate our method. **CNN/DM** (Hermann et al., 2015; Nallapati et al., 2016) is a well-known news summarization dataset with the associated highlights as summaries. **XSum** (Narayan et al., 2018) is an extremely abstractive dataset also in the news domain that contains a one-sentence summary for each article from BBC.

4.2 Comparison Methods

BART (Lewis et al., 2020) is a pre-trained Transformer model with a denoising objective widely used for abstractive summarization. **PEGASUS** (Zhang et al., 2020a) is another widely used pre-trained model with gap sentence generation and masked language modeling pre-training objectives. **GSum** (Dou et al., 2021) is an abstractive summarization model guided by extraction results with an identical double-encoder architecture as ours. **GOLD** (Pang and He, 2021) is an off-policy reinforcement learning method using the reference summary as a demonstrator. **SeqCo** (Xu et al., 2022) is a contrastive learning method that enforces the semantic similarity between reference and candidate. **BRIO** (Liu et al., 2022) is a contrastive learning method that assigns probability mass to candidate summaries according to their quality. **SimMCS** (Xie et al., 2023) is a multi-level contrastive learning method improved from BRIO and achieved state-of-the-art on both CNN/DM and XSum. **SLiC** (Zhao et al., 2023) is essentially a variant of BRIO, calibrating PEGASUS with types of contrastive losses. **MoCa** (Zhang et al., 2022) is improved from BRIO, introducing online candidate sampling.

Table 1: Automatic evaluation results on CNN/DM test set. †: results from our reproduction. The best results are in **bold**. The previous best results are highlighted with underline. R-1/2/L: ROUGE-1/2/L F1 scores. BS: BERTScore. BaS: BARTScore-$\mathcal{F}$.

Model	R-1	R-2	R-L	BS	BaS
BART	44.16	21.28	40.90	87.95	-3.91
PEGASUS	44.17	21.47	41.11	85.07†	-3.80†
GSum	45.94	22.32	42.48	-	-
GOLD	45.40	22.01	42.25	-	-
SeqCo	45.02	21.80	41.75	-	-
BRIO	47.78	23.55	44.57	89.14†	-3.62†
SimMCS	48.16	24.08	44.65	89.20	-3.58
SLiC	47.97	24.18	44.88	-	-
MoCa	48.88	24.94	45.76	-	-
IARSum	**48.96**	**25.14**	**45.93**	**89.32**	**-3.25**

Table 2: Automatic evaluation results on XSum test set.

Model	R-1	R-2	R-L	BS	BaS
BART	45.14	22.27	37.25	89.63†	-3.64†
PEGASUS	47.21	24.56	39.25	89.68	-3.89
GSum	45.40	21.89	36.67	-	-
GOLD	45.85	22.58	37.65	-	-
SeqCo	45.65	22.41	37.04	-	-
BRIO	49.07	25.59	40.40	89.10†	-3.79†
SimMCS	49.39	25.73	40.49	90.23	-3.77
SLiC	**49.77**	27.09	42.08	-	-
MoCa	49.32	25.91	41.47	-	-
IARSum	49.42	**27.20**	**42.50**	**92.13**	**-3.61**

4.3 Implementation Details

In the following experiments, we use BART as the backbone and start our model from the public fine-tuned versions bart-large-cnn[1] (on CNN/DM) or bart-large-xsum[2] (on XSum). As for hyperparameters, we set $\xi = 0.5$, $\mu = 0.5$, and $\lambda = 100$. We train our model on 4 NVIDIA RTX 4090 GPUs for 100K steps with a batch size of 16. The AdamW optimizer (Loshchilov and Hutter, 2019) with a noam learning rate schedule is used. The initial learning rate lr is 2e-3, and its value is updated following $lr^* = lr \cdot \min(\mathcal{S}^{-0.5}, \mathcal{S} \times \mathcal{W}^{-1.5})$, where $\mathcal{W}$ denotes the warmup steps, is set to 3,000, and $\mathcal{S}$ accumulates the current number of learning rate updates. The beam width k held for beam search decoding (Vijayakumar et al., 2016) is set to 16. The default number of iterations N is set to 3. Following conventions, we use ROUGE-F_1 scores (Lin, 2004) to evaluate the lexical overlap between the model-generated summary and the reference. Also, we use BERTScore (Zhang et al., 2020b) and BARTScore-$\mathcal{F}$ (Yuan et al., 2021) to evaluate their semantic similarity.

[1]https://huggingface.co/facebook/bart-large-cnn
[2]https://huggingface.co/facebook/bart-large-xsum

Table 3: Ablation study results on CNN/DM. i: the number of revision iterations. Dist.: Levenshtein distance. *w/o*: without.

Model	Iteration	R-1	R-2	R-L	Dist.
IARSum	i=1	46.19	22.27	43.69	0.60
	i=2	47.69	23.92	44.68	0.03
	i=3	48.96	25.14	45.93	0.01
	i=4	48.71	24.12	44.89	0.01
	i=5	48.74	24.12	44.91	0.01
-w/o $\mathcal{L}_{er}$	i=1	44.16	21.28	40.90	0.62
	i=2	45.28	22.63	40.96	0.59
	i=3	44.68	21.32	40.57	0.58
	i=4	44.30	22.38	41.34	0.59
	i=5	44.78	21.06	40.00	0.59

4.4 Main Results

We have the following observations from the automatic evaluation results in Table 1 and Table 2. 1) IARSum outperforms the backbone models by a large margin on both datasets, revealing the superiority of our learning scheme over the traditional supervised fine-tuning after pre-training. 2) IARSum also shows superiorities over the similar double-encoder Transformer - GSum. On the one hand, GSum needs an additional system to predict guidance signals. Besides, it suffers a severe training-inference discrepancy beyond exposure bias as the quality of guidance in training differs from in inference. In contrast, our IARSum requires no additional systems, and the model behaves identically during both training and inference. 3) Taking ROUGE as the measurement, IARSum achieves new SOTA on CNN/DM and matches the currently best performance on XSum. Moreover, IARSum demonstrates the best BERTScore and BARTScore on both datasets. We note that BRIO, SLiC, and our IARSum employ a similar training schema, which can be consolidated as the formulation in Eq. 14. IARSum stands out from the other two by emphasizing effective reprocessings after one-time summarization, which is the main reason for its superior performance.

4.5 Ablation Study

Our IARSum optimizes the backbone models mainly with global learning and iterative autoregressive generation. We conduct ablation studies to validate the effectiveness of these two strategies and list the experimental results in Table 3.

The Effectiveness of Global Learning. Note that IARSum$_{i=1}$ *-w/o* $\mathcal{L}_{er}$ represents a variant of our method that lacks the global learning procedure and involves no reprocesses after generating a draft summary (i.e., the backbone model trained with MLE). In contrast, IARSum$_{i=1}$ means our method drops further iterations once it has generated a summary. IARSum$_{i=1}$ performs better when trained with $\mathcal{L}_{er}$. We attribute the reason to the effectiveness of global learning in reducing exposure bias. Also, once giving up further iterations, our method only differs from RL-based GOLD and CTL-based BRIO regarding global learning objectives. IARSum$_{i=1}$ show better ROUGE scores than the two counterparts, indicating that learning with our defined triplet relations effectively enhances the current learning schema in abstractive summarization.

The Effectiveness of Iterations. To explore the effectiveness of the IAR generation paradigm, we adopt the normalized Levenshtein distance (Levenshtein et al., 1966) as an additional metric apart from ROUGE scores:

$$Dist(Y, Y^i) = \frac{1}{N} \sum \frac{Distance(Y, Y^i)}{\max(|Y|, |Y^i|)}, \quad (15)$$

where $Distance(\cdot, \cdot)$ denotes Levenshtein distance. Twofold insights can be drawn from Table 3. Firstly, we see from the lower part of the Table that the IAR with more iterations is useless without global learning. Secondly, the iterations performed in IARSum are only effective within a limited number of times. According to the Levenshtein distance, the impacts of iterations are hard to distinguish more than three times, and the generated summaries' quality even worsens. However, from another aspect, this finding also indicates that our method has a determined direction of good summaries within finite steps.

4.6 Case Study

To intuitively assess the quality of the IARSum generated summaries, we sample two cases from the CNN/DM test set and compare different systems' output in Table 4. The typical pattern of exposure bias can be found in the summaries generated by BART, i.e., if the model generates inappropriate content at the initial phase, this tends to cause a deviation of the subsequent information to ensure the coherence of a whole text. During each iteration, IARSum encodes the previously generated summary with bidirectional attention. This allows the model to modify potentially inappropriate tokens after understanding an entire summary and comparing it with the source document. We find in Table 4 that this mechanism also contributes to reducing hallucinatory facts. Moreover, benefiting

Table 4: Case study on CNN/DM. Content in blue is unfaithful or irrelevant to the reference. The draft is produced by IARSum$_{i=0}$, and we use red to mark the keywords (vs. the source document) it omits. After being reprocessed, the factuality and abstractiveness of the draft are improved.

System	Summary
	Case 1
Reference	RBS has racked up £50billion in losses since it was bailed out by taxpayers. State-back giant warned that it faces 'another tough year' to come. Lurched to a £446million loss for the first three months of the year.
BART	state-backed giant lurched to a £446million loss for the first three months of the year. Comes after RBS was forced to put aside £856million to cover lawsuits and fines still being decided. It includes an extra £334million for its part in rigging the foreign exchange market with other banks.
IARSum$_{i=1}$	RBS has racked up £50billion in losses since it was bailed out by taxpayers. The state-backed giant lurched to a £446million loss for the first three months of the year. It warned that it faces another 'tough year' in a statement last night. The bank was forced to put aside £856million to cover lawsuits and fines.
IARSum$_{i=2}$	Since receiving a bailout from taxpayers, RBS has racked up £50 billion in losses. The state-backed giant has issued a warning, stating that it anticipates facing another tough year. It reported a loss of £446 million for the first quarter of this year.
	Case 2
Reference	Hungarian national Evelin Mezei, 12, has been found safe and well. She had gone missing from the Stratford area in London last night. Evelin had been seen on CCTV footage with an unknown man.
BART	Evelin Mezei, a 12-year-old Hungarian national, was spotted with the man at around 10.30pm yesterday. She was last seen by her mother in East London, Scotland Yard said. But the youngster, who came to the UK six months ago, was traced this morning.
IARSum$_{i=1}$	Evelin Mezei, 12, went missing in Stratford, London, last night. She was seen on CCTV footage with an unknown man on a city street. The Hungarian national was found safe and well this morning. Her mother was last seen with the man's mother.
IARSum$_{i=2}$	Evelin Mezei, a 12-year-old Hungarian girl who went missing from the Stratford area in London last night, has been found safe and well. CCTV footage showed Evelin with an unknown man before her disappearance.

from our proposed lexical reward, the model can generate novel tokens absent in the previous summary during reprocessing. It helps to improve the abstractiveness of the generated summary, which is emphasized in abstractive summarization. We'll further discuss this point in section 5.

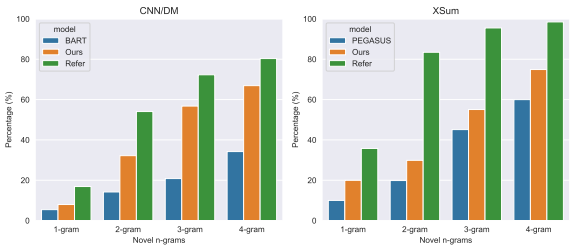

Figure 4: Novel n-grams on CNN/DM (left) and XSum (right) datasets.

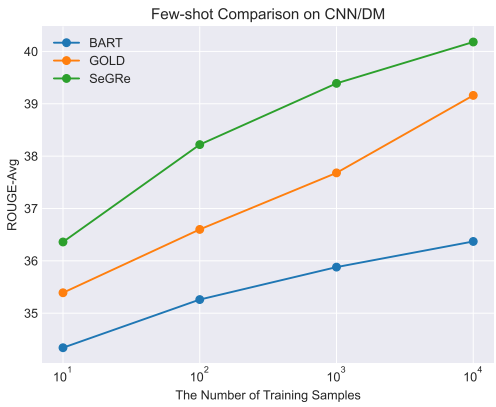

Figure 5: Few-shot performance comparison. ROUGE-Avg (the average of R-1, R-2, and R-L F_1 scores) scores are reported.

5 More Analyses

Abstractiveness. In our learning framework, we measure the increment of novel words using lexis rewards R_l. The case study approves the effectiveness of this strategy from a textual aspect. Here, we further understand the abstractiveness of IARSum-generated summaries through a quantitative analysis. According to previous works (Xie et al., 2023) and (Liu et al., 2022), we rate the percentage of novel n-grams that appear in the generated summary but not in the source document in Figure 4. We find that our IARSum can generate more novel n-grams than the baseline and reference, regardless of whether on moderately or extremely abstractive summarizations. Recalling the automatic evaluation and case study results, we assert that the summaries generated by IARSum closely resemble human written summaries in terms of both abstractiveness and semantic aspects.

Few-shot Performance. Based on the findings in our ablation study, we consider that the IAR generation mechanism introduced in IARSum makes the model more sensitive to the candidate's quality and can improve flawed candidates within a

finite number of iterations. Therefore, we conduct experiments in few-shot settings to confirm our assumptions. Following previous studies, we train IARSum on CNN/DM by varying the number of training samples from 10 to 10,000 and compare IARSum with the baseline BART and the RL-based GOLD to make the results convincing. According to Figure 5a, IARSum shows a remarkable few-shot learning ability. IARSum goes ahead more over the baseline as the training samples increase.

6 Conclusion

In this paper, we focus on improving the existing approaches that alleviate exposure bias suffered in abstractive summarization. Specifically, we introduce a novel iterative autoregressive summarization paradigm, IARSum. It models the generation of an abstract summary as a series of transitions of intermediate results, ranging from coarse to refined quality. IARSum also enables learning rational relations among a document, the reference summary, and candidate summaries under a standard GL framework. Extensive comparison experiments revealed the effectiveness and advancement of our method.

References

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS 2015*, pages 1171–1179.

Shuyang Cao and Lu Wang. 2021. CLIFF: contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6633–6649.

Tanay Dixit, Fei Wang, and Muhao Chen. 2023. Improving factuality of abstractive summarization without sacrificing summary quality. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023*, pages 902–913.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *NAACL-HLT 2021*, pages 4830–4842.

Markus Dreyer, Mengwen Liu, Feng Nan, Sandeep Atluri, and Sujith Ravi. 2023. Evaluating the tradeoff between abstractiveness and factuality in abstractive summarization. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2044–2060.

Sebastian Goodman, Nan Ding, and Radu Soricut. 2020. Teaforn: Teacher-forcing with n-grams. In *EMNLP 2020*, pages 8704–8717.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *NIPS 2016*, pages 4601–4609.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS 2015*, pages 1693–1701.

G. Senthil Kumar and Midhun Chakkaravarthy. 2023. A survey on recent text summarization techniques. In *MIWAI 2023*, volume 14078 of *Lecture Notes in Computer Science*, pages 496–502.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

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 *ACL 2020*, pages 7871–7880.

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

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. *CoRR*, abs/1907.11692.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers)*, pages 1065–1072.

Yixin Liu, Pengfei Liu, Dragomir R. Radev, and Graham Neubig. 2022. BRIO: bringing order to abstractive summarization. In *ACL 2022*, pages 2890–2903.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 1906–1919.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL 2016*, pages 280–290.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP 2018*, pages 1797–1807.

Richard Yuanzhe Pang and He He. 2021. Text generation by learning from demonstrations. In *ICLR 2021*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018*.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *ACL 2023*, pages 6252–6272.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *ACL 2016*.

Jeewoo Sul and Yong Suk Choi. 2023. Balancing lexical and semantic quality in abstractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023*, pages 637–647. Association for Computational Linguistics.

Caidong Tan. 2023. Deep reinforcement learning with copy-oriented context awareness and weighted rewards for abstractive summarization. In *CACML 2023*, pages 84–89.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Jiawen Xie, Qi Su, Shaoting Zhang, and Xiaofan Zhang. 2023. Alleviating exposure bias via multi-level contrastive learning and deviation simulation in abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9732–9747.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022. Sequence level contrastive learning for text summarization. In *AAAI 2022*, pages 11556–11565.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *NeurIPS 2021*, pages 27263–27277.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *ICLR 2020*.

Xingxing Zhang, Yiran Liu, Xun Wang, Pengcheng He, Yang Yu, Si-Qing Chen, Wayne Xiong, and Furu Wei. 2022. Momentum calibration for text generation. *CoRR*, abs/2212.04257.

Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J. Liu. 2023. Calibrating sequence likelihood improves conditional language generation. In *ICLR 2023*.

A Limitations

Although we pioneer present an iterative autoregressive summarization (IAR) mechanism, which suffers little prior bias since it relies on no metrics to measure document-summary semantic or lexical similarity, performing IAR requires a dual-encoder Transformer architecture. This setting is intuitively incompatible with nowadays decoder-only large pretrained Transformer models. On the one hand, this work confirmed the effectiveness of using the IAR mechanism to improve abstractive summarization, also, it left further work for us to adapt the mechanism for large language models.

B More Analyses

B.1 Varying the Beam Width.

Note that the global learning objective of IAR-Sum is to maximize the expected rewards calculated over the candidate summaries sampled from $P_\theta(\cdot|Y^{i-1}, X)$. There is a gap between the learning objective and our training implementation. During training, we are inspired by the Monte Carlo (MC) algorithm and use k candidates to represent the infinite searching space. Intuitively, a larger beam width (k) used in beam search is more adequate to approximate the expected distribution and, in turn, better summarization performance. To validate this assumption, we train our model on both datasets and use different beam widths of 4, 8, 16, 32, and 64 to sample candidates. Figure 6 displays the ROUGE-Avg score of each resulting version.

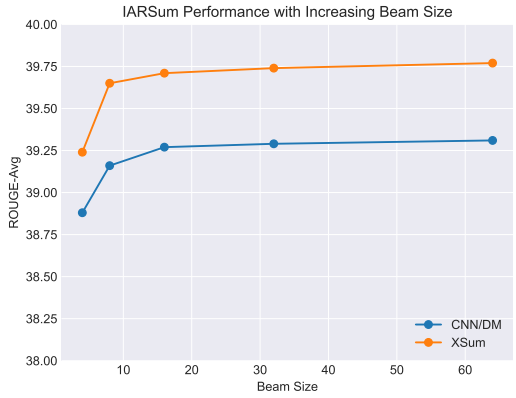

Figure 6: The performance of the IARSum trained with varying numbers of sampled candidates. ROUGE-Avg (the average of R-1, R-2, and R-L F_1 scores) scores are reported.

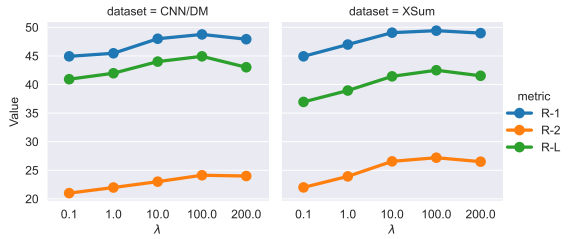

Figure 7: The performance of IARSum with increasing λ on CNN/DM and XSum.

Unsurprisingly, increasing the beam width can indeed boost the model's performance. However, the ROUGE score improvement reduces once the k is over 16. We set k to 16 to save computational costs.

B.2 The Decide of λ Value.

To find an optimal weight coefficient λ that integrates the global learning objective into the token-level MLE, we perform a grid search in $\{0.1, 1, 10, 100, 200\}$. The search process is visualized in Figure 7. Notably, the performance of IARSum with varying λ shows a similar trend on both datasets. It is observed that a too-small weight suppresses global learning efficacy. On the contrary, once λ reaches the magnitude above one hundred, varying its value makes inconspicuous effects. We finally set λ to 100 without distinguishing datasets.