Alleviating Exposure Bias in Abstractive Summarization via Sequentially Generating and Revising

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

Abstract

Abstractive summarization commonly suffers from exposure bias caused by supervised teacher-force learning, that a model predicts the next token conditioned on the accurate pre-context during training while on its preceding outputs at inference. Existing solutions bridge this gap through un- or semi-supervised holistic learning yet still leave the risk of error accumulation while generating a summary. In this paper, we attribute this problem to the limitation of unidirectional autoregressive text generation and introduce post-processing steps to alleviate it. Specifically, we reformat abstractive summarization to sequential generation and revision (SeGRe), i.e., a model in the revision phase re-inputs the generated summary and refines it by contrasting it with the source document. This provides the model additional opportunities to assess the flawed summary from a global view and thereby modify inappropriate expressions. Moreover, we train the SeGRe model with a regularized minimum-risk policy to ensure effective generation and revision. A lot of comparative experiments are implemented on two well-known datasets, exhibiting the new or matched state-of-the-art performance of SeGRe.

Keywords: Abstractive summarization, Autoregressive language modeling, Reinforcement learning

1. Introduction

Abstractive summarization is a classical neural language generation (NLG) task that aims to condense a long document into a shorter text, retaining only the salient information (Kumar and Chakkaravarthy, 2023; Xie et al., 2023). Recently, the advanced language modeling technologies founded on large-scale corpora significantly boosted abstractive summarization (Lewis et al., 2020; Zhang et al., 2020a), and the approaches that formulate the task as a sequence-to-sequence (Seq2Seq) learning problem have achieved unprecedented outcomes. They commonly train an autoregressive language model (Vaswani et al., 2017) with maximum likelihood estimation (MLE), and the teacher-forcing mechanism (Goyal et al., 2016) is together used to ensure training efficiency and stability. However, such a model predicts each token in summary conditioned on the exact pre-context during training but on its preceding outputs at inference, causing a training-inference discrepancy called *exposure bias* (Bengio et al., 2015; Goodman et al., 2020), which heavily limits summarization performance.

Substituting or augmenting the pure token-level MLE with holistic objectives is widely used to address this problem, which simultaneously gives up supervised teacher-forcing by using an un- or semi-supervised learning paradigm. Typically, reinforcement learning-based methods (Tan, 2023; Pang and He, 2021) train a model to maximize the rewards defined over each candidate summary. A similar objective can also be achieved with contrastive learning (Xu et al., 2022; Liu et al., 2022; Xie et al., 2023), where a model is trained to assign probability mass to candidate summaries according to their quality. However, although whole summaries are generated and optimized in training, which coordinates well with inference, these methods do not change the paradigm that a model makes predictions under erroneous pre-context. As a result, errors are accumulated whenever generating a summary.

As depicted in Figure 1, the mentioned error accumulation (Ross et al., 2011) is inevitable under the existing frameworks. On the one hand, there is no guarantee that the trained model will assign all probability to the exact token at each generation step, no matter the training or testing. Moreover, the autoregressive summary generation is a unidirectional process that features 1) previously generated improper tokens mislead the subsequent tokens, causing a deviation from the reference and gradually increasing lexical and semantic discrepancies. 2) The model can do nothing to fix the already improper tokens at each generation step.

In this paper, we argue that existing solutions for exposure bias are limited due to the unidirectional autoregressive paradigm, and we introduce further post-processing steps from a global view to address this problem. We propose a novel summarization framework - **SeGRe**, which yields an abstractive summary by **Se**quentially **Ge**nerating and **Re**vising. Specifically, SeGRe first generates

† Equal contributions in this work.
* Corresponding author.

LREC-COLING 2024, pages 739–750
20-25 May, 2024.

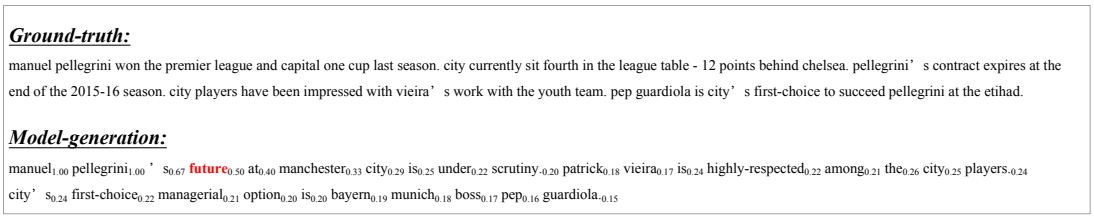

Figure 1: Illustration of error accumulation in autoregressive generation paradigm. Footnote counts ROUGE-1 F1 score up to each word in a summary. After the first improper word, 'future,' was generated, it became apparent that the overlap between the two summaries gradually decreased.

a draft summary, and after that, it re-inputs this summary and refers to the source document to produce polished versions with fewer errors. The main challenge for SeGRe is to learn how to effectively refine a flawed summary within a finite number of revisions. To tackle it, we first encode the model-generated summary with bidirectional attention and then, based on that, assess the degree to which a refined summary improves the original regarding semantic and lexical overlaps with the reference. Finally, we define summary-level revision rewards and optimize the parameters of the SeGRe model with a regularized minimum-risk training (Shen et al., 2016) algorithm.

The idea of our method is a bit similar to the partially autoregressive model, LevT (Gu et al., 2019). However, LevT features a non-autoregressive revision process that predicts definite actions of deletion and (or) insertion on each token in a text. We instead highlight the understanding and enhancement of the generated summary rather than introducing a novel text generation pattern out of the autoregressive scope. Moreover, although we extend attention-based components, our method requires no external systems like predictors used in LevT and can be easily integrated with any holistic learning policy. Our contributions are as follows:

- We introduce a novel abstractive summarization paradigm, SeGRe, which performs summary generation and revision sequentially by a single model to alleviate exposure bias.
- We implement SeGRe with a double-encoder transformer and optimize the model through a minimum-risk training strategy that maximizes the expected revision reward.
- Extensive experiments are conducted on two public datasets to test our methods. Results show that SeGRe matches or outperforms previous state-of-the-art (SOTA) approaches in generating human-like summaries. The reduced error accumulation is also evidenced in human evaluations, where SeGRe generates more faithful facts. Furthermore, we transfer SeGRe to few-shot settings and show its superior robustness.

2. Related Work

2.1. Language Generation

There are mainly two paradigms of language generation: autoregressive and non-autoregressive. Autoregressive (AR) generation (Radford et al., 2019; Brown et al., 2020) refers to a step-by-step process: (1) generating a token conditioned on a given prompt or from scratch, (2) appending the newly generated token to the tail of the sequence, and (3) repeating this process until the EOS token (end-of-sentence) is generated or reach the max length. Non-autoregressive generation (Gu et al., 2019, 2018) instead refers to (iteratively) modifying a sequence of tokens bounded by a fixed length or sampling an entire text from a certain distribution (Li et al., 2022). AR is most widely applied in various text generation tasks, including summarization. However, it inevitably causes error accumulation and fuels the exposure bias. Therefore, our SeGRe introduces an additional revision after the autoregressive generation to alleviate this limitation.

2.2. Solutions of Exposure Bias

Holistic-level learning technologies, including reinforcement and contrastive learning, are widely used to alleviate exposure bias.

Reinforcement learning (Tan, 2023; Roit et al., 2023) gives a model a sequence-level view by rewards that commonly vary from evaluation metrics. Early works are always based on on-policy learning (Paulus et al., 2018), where a model generates a sampled candidate and a greedy-search 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. 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.

Contrastive learning (Hadsell et al., 2006) uses positive and negative sample pairs to train a model to distinguish real data labels. Cao and Wang (2021) build sample pairs by the back-translation method and improve the faithfulness and factual-

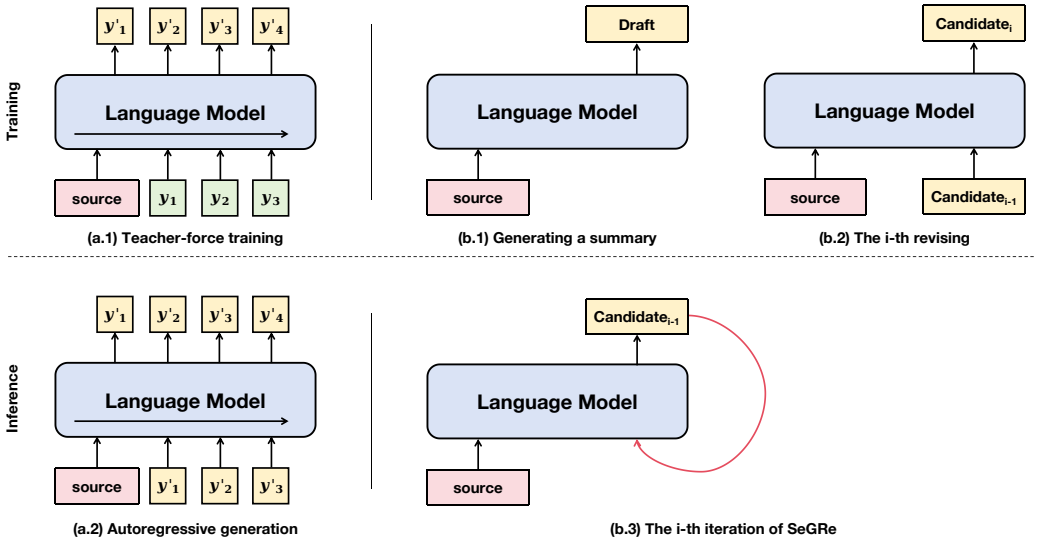

Figure 2: Illustration of the unidirectional autoregressive (AR) generation (a) and our SeGRe (b). The yellow blocks refer to model-generated tokens or summaries, and the green ones are gold references. AR produces a text token by token, while our SeGRe reaches a final summary based on the previous version through successive iterations.

ity of the generated summaries. Xu et al. (2022) contrast semantic similarity among source documents, candidates, and references without building negative samples. In recent years, the ranking method has extended from contrastive learning and achieved state-of-the-art performance in abstractive summarization. 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. A similar work was also done by An et al. (2022). Recently, a lot of improved BRIO variants (Xie et al., 2023; Zhao et al., 2023; Zhang et al., 2022) were proposed in succession. All these methods achieved surprising performance but still inevitably suffered error accumulation caused by autoregressive generation.

3. Background

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

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

For the abstractive paradigm, existing approaches commonly build a model with parameter θ to fit f by approximating the conditional probability $P(Y|X)$ token by token, widely known as autoregressive text generation. Maximum likelihood estimation (MLE) is most used to learn the autoregressive model. It maximizes the probability mass of gold reference assumed by the model, following independent and identically distributed conditions, i.e., $\max_\theta P_\theta(Y|X) = \max_\theta \prod_{t=1}^{l} P_\theta(y_t|Y_{<t}, X)$, where l denotes the length of reference and $Y_{<t}$ subsequence $\{y_1, y_2, \cdots, y_{t-1}\}$.

When it comes to training, the teacher-forcing algorithm (Goyal et al., 2016) is always used, which minimizes the sum of the negative log-likelihoods (NLL) of each token in the summary Y:

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

Though it ensures stable MLE learning, such a trained model depends heavily on the exact preceding content, and it is prone to suffer error accumulation during inference once any improper token is generated in previous steps.

From the probabilistic perspective, a model learns to sample the next token at timestep t from the distribution $P(\cdot|Y_{<t}, X)$, while its goal at inference is to sample from $P(\cdot|Y'_{<t}, X)$. This gap is the so-called exposure bias. SOTA approaches fix it by learning the model to adjust the probability mass $P(Y'|X)$ according to the quality of Y', where Y' refers to a candidate summary. However, the fact is unchanged that a token y'_t is sampled from $P(\cdot|Y'_{<t}, X)$ during inference, and the error accumulation problem is thus still.

Calibrating the flawed distribution $P(\cdot|Y'_{<t}, X) \rightarrow P(\cdot|Y_{<t}, X)$ before sampling - $y'_t \sim P(\cdot|Y_{<t}, X)$ is rational to address this problem. However, calibration requires a global perception of the context, which contradicts the unidirectional nature of autoregressive generation. Moreover, calibrating at

each step is also time-consuming. In this paper, we formulate abstractive summarization as a two-stage task for trading between performance and efficiency, namely generation and revision. As shown in Figure 2 (b.3), after generating a draft summary, the model re-inputs and rewrites it by consulting the source document. Therefore, we unfold the conditional probability of reference summary $P(Y|X)$:

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

where N counts the number of revisions. Y^0 is a draft, Y^N is ideally the reference, and Y^i denotes the refined versions.

4. Method

4.1. Architecture

We implement SeGRe 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 learning, we start our model with a pre-trained single-encoder Transformer, and both encoders share identical initial parameters.

The architecture of SeGRe 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). SeGRe instead adapts series encoders, where the second encoder relies on the output of the first to encode the input text, similar to a Transformer decoder without a sequence mask. Besides, GSum uses the second encoder to encode guidance words, while SeGRe's second encoder is used to encode the candidate summary. Mathematically, SeGRe models the following non-normalized probabilities (logits) during generation and revision, respectively:

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

where E_θ^1, E_θ^2 are the first and second encoders, and D_θ is the decoder. Then, the token y_t^i is sampled from the distribution $softmax(\bar{P})$.

4.2. Learning

To facilitate our illustration, we cast the problems of abstract summary generation and revision to a unified process of sequential iterations. During each iteration, the model draws a candidate Y^{i-1} from the previous outputs[1], estimates its quality,

[1]At the first iteration, i.e., the generation stage, the input is empty.

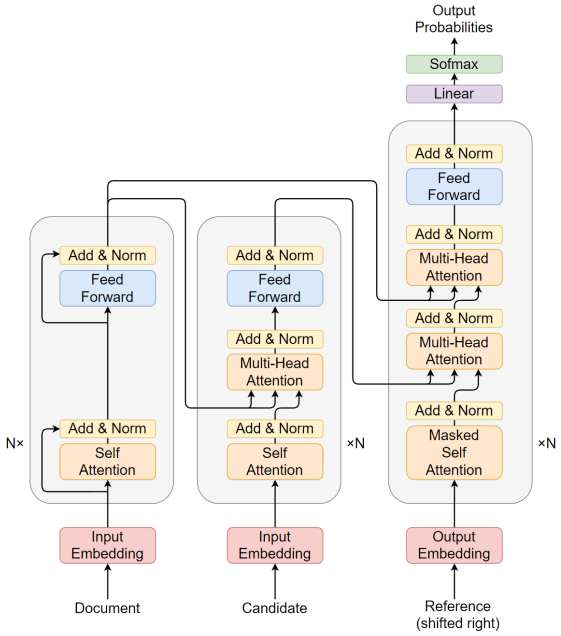

Figure 3: The architecture of SeGRe model.

and produces a new one Y^i. The learning objective is to make the refreshed candidate Y^i as close to the reference Y as possible. For this purpose, we view a candidate from semantics and lexis aspects and define corresponding rewards.

Semantics Rewards To ensure faithfulness, a summary should be logically entailed in the source document (Dreyer et al., 2023), and semantic reasoning is used to detect this relation (Roit et al., 2023). Even so, researchers observed that human tends to write summaries with hallucinatory words to keep abstractiveness (Maynez et al., 2020) despite contradicting summary-document entailment. To echo these insights, we take the reference summary as standard to coordinate the relationship between a candidate and the source document. We introduce a relation recognition function $\mathcal{S}(\cdot,\cdot)$ and assess a candidate on the semantics level:

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

where $\langle \cdot,\cdot \rangle$ distinguishes the gap between two relations. Further, we reward the model according to the gain that a revised candidate surpasses the original on M_s:

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

Lexis Rewards From the lexis view, we encourage a candidate, after revising, to contain more tokens overlapped with the reference. To this end, we reward the model using the function:

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

where $\{\cdot\}$ denotes a token set and $|\cdot|$ 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 SeGRe is to maximize the following expected reward:

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

4.3. Training Strategy

As we learn SeGRe with a maximum expected reward objective, the infinite sampling space makes the expectation in Eq.9 untraceable. Predominant studies adopt the Monte Carlo approach to address this problem, which approximates the real distribution with empirical samples. We follow this idea and further use a minimum-risk training (Shen et al., 2016) algorithm to train our model. At each revision, k candidates $Y^i_{(1)}, \cdots, Y^i_{(k)}$ are sampled from $P_\theta(\cdot|Y^{i-1}, X)$ using beam-search (Vijayakumar et al., 2016), and the model is trained to minimize an expected reward (ER) loss:

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

Relation Recognizing Another challenge we encounter is the instantiation of function $\mathcal{S}$. Given a pair of document and summary (X, Y), the second encoder of SeGRe can be used to encode their correlation:

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

Although parameter-efficient, dynamically learned parameters θ cause the observation of $\mathcal{S}$ to vary sharply as training progresses. Following Zhang et al. (2022) lessons, we introduce momentum-based parameterization to address this. In detail, 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 Training To save the computational costs of generating candidates, we use offline samples during training. A pre-trained summarizer is first fine-tuned with MLE and proceeds to generate k candidate summaries for every document in the training set. Each of these candidates, coupled with the source document, forms a $\{X, Y^{i-1}\}$ pair used for the model training. Note that the generated candidates share varying degrees of errors, and the ones closer to the reference simulate the drafts that have been revised more times. This nature facilitates the training to focus on only one iteration without considering the multi-turn reward. However, simply training towards revision is not guaranteed to retain the functionality of generation. Following (Liu et al., 2022) and (Zhao et al., 2023), we add a regularization term in Eq.9, and the overall loss function is then:

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

4.4. Inference Time

At inference, the model initially conditions only an input source document to generate a draft summary, which is then utilized as an additional condition for the subsequent revision. Once the number of revision steps (N) exceeds 1, the candidate produced from the preceding revision is taken as the input for the next. Beam-search with the width k is employed whenever generating a summary.

5. Experiments

5.1. Datasets

We use two public open-domain datasets to evaluate our method. **CNN/DM** (Hermann et al., 2015; Nallapati et al., 2016) is a widely used news summarization dataset that treats 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.

5.2. Comparison Methods

BART (Lewis et al., 2020) is a pre-trained Transformer model with a denoising objective and is 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.

5.3. 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[2] (on CNN/DM) or bart-large-xsum[3] (on XSum). As for hyperparameters, we set $\xi = 0.5$, $\mu = 0.5$, and $\lambda = 0.05$. We train our model on 4 NVIDIA RTX 3090 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 as $lr^* = lr \cdot \min(\mathcal{S}^{-0.5}, \mathcal{S} \times \mathcal{W}^{-1.5})$, 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 revisions N on each draft is set to 3.

Metrics Following conventions, we use ROUGE-F1 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.

5.4. Main Results

We have the following observations from the automatic evaluation results in Table 1. 1) SeGRe outperforms the base model BART by a large margin on both datasets, proving the superiority of our learning scheme over plain supervised fine-tuning after pre-training. 2) SeGRe shows better scores compared to the similar double-encoder Transformer - GSum. On the one hand, GSum needs an additional system to predict guidance signals. Besides, it suffers further discrepancy other than exposure bias since the quality of guidance in training differs from in inference. On the contrary, our SeGRe needs no extra systems, and the model takes identical behavior, whether during training or inference. 3) Based on ROUGE standards, SeGRe achieves a new SOTA on XSum and matches the recent best performance on CNN/DM. Moreover, SeGRe shows the best BERTScore and BARTScore on both datasets. It is also worth noting that BRIO, SLiC, and SeGRe have similar training loss functions that can be unified as the format of Eq.14. Except for a marginal number of additional parameters, the most difference lies in the learning scheme of summary-level objectives. Our method emphasizes effective revision after unidirectional summarization, which is the main reason for SeGRe's advanced performance.

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

Model	Scale	R-1	R-2	R-L	BS	BaS
CNN/DM						
BART	406M	44.16	21.28	40.90	87.95	-3.91
PEGASUS	568M	44.17	21.47	41.11	85.07†	-3.80†
GSum	473M	45.94	22.32	42.48	-	-
GOLD	-	45.40	22.01	42.25	-	-
SeqCo	-	45.02	21.80	41.75	-	-
BRIO	406M	47.78	23.55	44.57	89.14†	-3.62†
SimMCS	406M	48.16	24.08	44.65	89.20	-3.58
SLiC	2B	47.97	24.18	44.88	-	-
MoCa	406M	48.88	**24.94**	**45.76**	-	-
SeGRe	608M	**48.96**	24.13	44.93	**89.32**	**-3.25**
XSum						
BART	406M	45.14	22.27	37.25	89.63†	-3.64†
PEGASUS	568M	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	568M	49.07	25.59	40.40	89.10†	-3.79†
SimMCS	568M	49.39	25.73	40.49	90.23	-3.77
SLiC	2B	**49.77**	27.09	42.08	-	-
MoCa	568M	49.32	25.91	41.47	-	-
SeGRe	608M	49.42	**27.20**	**42.50**	**92.13**	**-3.61**

Table 1: Automatic evaluation results. †: the results of our reproduction. The best results are in **bold**. The previous best results are highlighted with underline. Scale means the number of model parameters. R-1/2/L: ROUGE-1/2/L F1 scores. BS: BERTScore. BaS: BARTScore-$\mathcal{F}$.

Model	R-1	R-2	R-L	Dist
SeGRe				
SeGRe$_{i=0}$	46.19	22.27	43.69	0.60
SeGRe$_{i=1}$	47.69	23.92	44.68	0.03
SeGRe$_{i=2}$	48.76	24.13	44.93	0.01
SeGRe$_{i=3}$	48.71	24.12	44.89	0.01
SeGRe$_{i=4}$	48.74	24.12	44.91	0.01
SeGRe - *w/o* $\mathcal{L}_{er}$				
SeGRe$_{i=0}$	44.16	21.28	40.90	0.62
SeGRe$_{i=1}$	45.28	22.63	40.96	0.59
SeGRe$_{i=2}$	44.68	21.32	40.57	0.58
SeGRe$_{i=3}$	44.30	22.38	41.34	0.59
SeGRe$_{i=4}$	44.78	21.06	40.00	0.59

Table 2: Ablation study results on CNN/DM. Footnote i denotes the number of revision iterations. Dist: Levenshtein distance. *w/o*: without.

5.5. Ablation Study

The differences between our SeGRe and the traditional baseline mainly lie in holistic learning and the revision mechanism. We first would like to explore each of their functionality. Besides, whether multi-time revisions contribute to generating high-quality summaries is worth knowing, too. We dis-

System	Summary
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.
Draft	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.
SeGRe$_{i=1}$	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.
SeGRe$_{i=2}$	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. A loss of £446 million for the first quarter of this year.
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.
Draft	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.
SeGRe$_{i=1}$	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 this morning. Her mother was last seen with the man's mother.
SeGRe$_{i=2}$	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.

Table 3: Case study on CNN/DM. Content in blue is unfaithful or irrelevant to the reference. The draft is produced by SeGRe$_{i=0}$, and we use red to mark the keywords (vs. the source document) it omits. After being revised, the factuality and abstractiveness of the draft are improved.

Dataset	Model	ECE	Acc	Conf
CNN/DM	BART†	0.4097	0.3711	0.7365
	BRIO-Mul†	**0.2719**	0.4271	0.6652
	SeGRe	0.2633	0.4309	0.6485
XSum	BART†	**0.2369**	0.4688	0.6990
	BRIO-Mul†	0.1423	0.4744	0.5881
	SeGRe	0.1348	0.4805	0.5530

Table 4: Calibration analysis results. ECE: expected calibration error. Acc: accuracy. Conf: confidence. †: the results reported in (Liu et al., 2022).

cuss these issues in the ablation study and list the experimental results in Table 2.

The Effectiveness of Holistic Learning Note that SeGRe$_{i=0}$ - *w/o* $\mathcal{L}_{er}$ represents a SeGRe variant that lacks the revision objective (i.e., only learned with MLE), and SeGRe$_{i=0}$ means our method drops the revision stage. Of the two, SeGRe$_{i=0}$ performed better. We attribute this to the effectiveness of mini-risk training in reducing exposure bias. Also, once giving up revision, our method only differs from RL-based GOLD and CTL-based BRIO in the set of holistic objectives (rewards). SeGRe$_{i=0}$ show the worst results, indicating that the reward function presented in Eq.6 is more effective when performed with revision.

The Effectiveness of Revision We are interested in the effectiveness of revision regarding whether an additional revision stage contributes to improving the summaries' quality and whether more times revisions are recommended. To explore these problems, we first use the normalized Levenshtein distance (Levenshtein et al., 1966) as one of the metrics of revision effectiveness:

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

where $Distance(\cdot, \cdot)$ denotes Levenshtein distance. Then, two insights can be drawn from Table 2. Firstly, we see from the lower part of the Table that revision is useless without holistic learning. Secondly, the revision is only effective in a limited number of times. According to the Levenshtein distance, the impacts of revision are hard to distinguish after three times, and the summaries' quality is even worse. However, from another aspect, it also proves that our method has a determined direction of good summaries within finite steps.

5.6. Calibration Analysis

To verify whether the proposed method effectively reduces the exposure bias, we follow BRIO (Liu et al., 2022) and analyze the expected calibration error (ECE). Table 4 shows calibration results on CNN/DM and XSum testing sets. We use the tercom toolkit [4] to label the SeGRe-generated summaries to align with BRIO works, and the number of buckets is set to 10. These results demonstrated that our model is calibrated on a similar level as BRIO.

5.7. Case Study

To intuitively estimate the degree to which SeGRe alleviates exposure bias, we sample two cases from

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

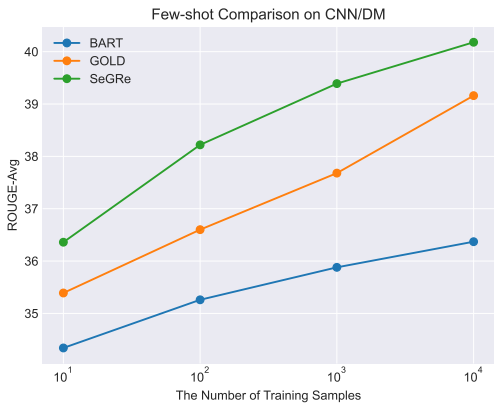

Figure 4: The few-shot comparison among BART, GOLD, and SeGRe. ROUGE-Avg (the average of R-1, R-2, and R-L F_1 scores) is reported.

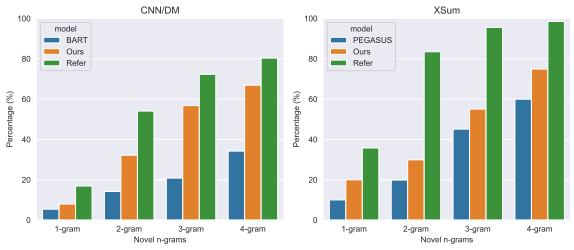

Figure 5: Novel n-grams on CNN/DM (left) and XSum (right) datasets.

the CNN/DM test set and display the outputs from SeGRe in Table 3. The typical cause of exposure bias can be found in the draft summaries, i.e., irrelevant or hallucinatory content. At revision times, the model encodes the candidate summary with bidirectional attention. It allows the model to modify unsatisfied statements after understanding an entire summary and comparing it with the source document. Cases in Table 3 revealed that hallucinatory facts can be fixed during this procedure. More than that, benefiting from maximum expected rewards learning, the model can generate novel tokens not in the previous candidate during revision. It helps to improve the abstractiveness of the final summary, which is emphasized in abstractive summarization. We detailly discuss this feature in section 6.

6. More Analyses

Few-shot Performance Based on the findings in our ablation study, we believe that the revision mechanism introduced in SeGRe makes the model more sensitive to the candidate's quality and can improve flawed candidates within a finite number of rewrites. Therefore, we conduct experiments in few-shot settings to confirm our assumptions. Following

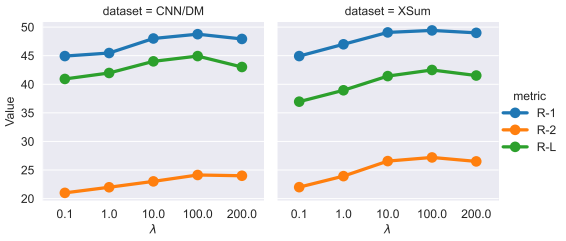

Figure 6: The performance of SeGRe with increasing λ on CNN/DM (left) and XSum (right) datasets.

previous studies, we train SeGRe on CNN/DM by varying the number of training examples from 10 to 10,000 and compare SeGRe with the baseline BART and the RL-based GOLD to make the results convincing. According to Figure 4, SeGRe shows a remarkable few-shot learning capability. SeGRe goes ahead more over the baseline as the training samples increase.

Abstractiveness Since we measure the increment of novel words using lexis rewards R_l, our case study proves the effectiveness of this setting from a textual aspect. Here, we further understand the abstractiveness of the generated summary through 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 5. It can be seen that no matter whether in moderately or extremely abstractive scenarios, our SeGRe can generate more novel n-grams than the baseline. Considering the automatic evaluation and case study results, we conclude that the summaries produced by SeGRe are human-like on both abstractiveness and semantic levels.

The Decide of λ Value To find an optimal factor λ that incorporates the holistic learning objective into the plain MLE, we perform a grid search in $\{0.1, 1, 10, 100, 200\}$. The search process is visualized in Figure 6. Notably, the performance of SeGRe shows a similar trend with varying λ on both datasets. It seems that a too-small factor suppresses holistic learning efficacy. Further, once λ reaches the magnitude of hundreds, varying its value makes inconspicuous effects. We finally set λ to 100 without distinguishing datasets.

The Impact of Beam Width Note that the learning objective of SeGRe is to maximize the expected revision reward calculated over the candidates sampled from $P_\theta(\cdot|Y^{i-1}, X)$. There is a gap between this 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,

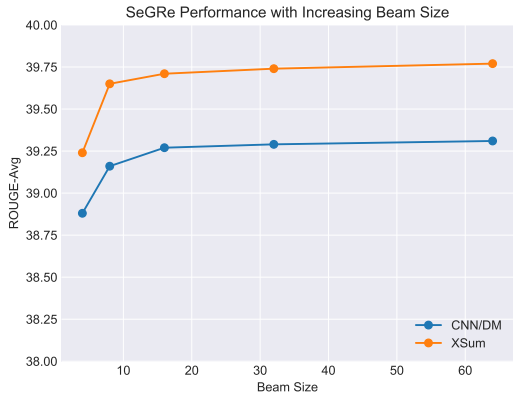

Figure 7: The performance of the SeGRe trained with varying numbers of sampled candidates.

Model	R-1	R-2	R-L
SeGRe	**46.34**	**22.27**	**42.71**
SeGRe-R	46.15	22.24	42.21
SeGRe-B	45.68	21.25	41.59
SeGRe-S	45.72	21.66	42.03

Table 5: The performance of SeGRe and its variants on a CNN/DM subset.

Dataset	R-1	R-2	R-L	BS	BaS
CNN/DM	48.18	24.07	44.69	89.02	-3.18
XSum	49.42	26.33	41.90	92.16	-3.54

Table 6: The performance of SeGRe with a substituted reward function.

in turn, better summarization performance. To validate this hypothesis, we train our model on both datasets while using different beam widths of 4, 8, 16, 32, and 64 to sample candidates. Figure 7 displays the evaluation of each resulting version. Unsurprisingly, increasing the beam width can indeed boost the model's performance. But the gain of ROUGE scores reduces since the k is over 16. We set k to 16 to save computational costs.

Reward Function's Designing We conducted comparison experiments to explore if there are better choices for reward design. Now that the rewards introduced in our method consider lexics and semantics aspects, we individually substituted the original lexics reward to ROUGE-1 F1 score or the original semantics reward to BERTScore, with the other settings unchanged. The resulting models are named SeGRe-R and SeGRe-B, respectively. Also, we substituted both rewards simultaneously and got the model SeGRe-S. It is worth noting that ROUGE-1 F1 and BERTScore are commonly used in previous works (e.g., BRIO-like models), serving as a scoring function. We contrasted the original SeGRe model and its three variants on a subset of CNN/DM, including 20,000/1,000 training/test samples. The results are shown in Table 5. We draw the following three conclusions: 1. Substituting the original lexics reward function to ROUGE-1 F1 score affects little on model performance. 2. Substituting the original semantics reward function to BERTScore clearly damages the performance. It may be because this setting gives up restraining the meaning of the second encoder's outputs. 3. Due to the problem raised in "2," substituting the design of both reward functions leads to a suboptimal SeGRe-S model compared with the original version.

7. Conclusion

In this paper, we focus on improving the existing pattern of alleviating exposure bias in abstractive summarization. Specifically, we introduce SeGRe, which integrates the functions of generation and revision in a single model and produces human-like summaries. We demonstrate the advanced performance of our method through extensive experiments and further unveil the factors that may affect SeGRe's performance through empirical analyses. We plan to embed subtly learned summarization models into SeGRe in future works to further extend our method.

8. Limitations

We omitted to consider the *synonyms issue* when designing our lexis reward function. One reason is that we follow previous studies and mainly evaluate the summarization performance with ROUGE scores, which are insensitive for synonym words. To reach higher scores, the introduced lexis reward only needs to encourage the model-revised summaries to contain more reference tokens without considering their synonyms. Although it does not hinder our SeGRe from achieving advanced evaluation results, we focus on the synonyms issue in improving real-world applications. Here, we look up WordNet (Miller, 1995) to get a synonym set $\{Y^*\}$ for the reference words $\{Y\}$ and further reward our model once a revised summary contains the words in $\{Y^*\}$. The evaluations of the resulting model are shown in Table 6. It seems that the mentioned strategy makes little effort to improve summaries' semantic quality and reversely damages lexical quality (ROUGE scores). Besides, looking up lexicons is time-consuming and contextually unaware. Therefore, dealing with synonyms is still a limitation of our current method, and we will leave this issue as one of our future directions for a deeper study.

9. Acknowledgements

We sincerely appreciate the anonymous reviewers for their valuable suggestions and approval. This work is supported by the Changsha Science and Technology Major Special Project (No.kh2202006).

10. Bibliographical References

Chenxin An, Ming Zhong, Zhiyong Wu, Qin Zhu, Xuanjing Huang, and Xipeng Qiu. 2022. Colo: A contrastive learning based re-ranking framework for one-stage summarization. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5783–5793. International Committee on Computational Linguistics.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6633–6649. Association for Computational Linguistics.

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, Dubrovnik, Croatia, May 2-6, 2023*, pages 2044–2060. Association for Computational Linguistics.

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.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

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.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *NeurIPS*.

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), Virtual Event, August 1-6, 2021*, pages 1065–1072. Association for Computational Linguistics.

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, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

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.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 627–635. JMLR.org.

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*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, pages 5998–6008.

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*.

11. Language Resource References

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.

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.