

BRIO: Bringing Order to Abstractive Summarization

Yixin Liu¹, Pengfei Liu², Dragomir Radev¹, Graham Neubig²

¹Yale University, ²Carnegie Mellon University

{yixin.liu, dragomir.radev}@yale.edu, {pliu3, gneubig}@cs.cmu.edu

Abstract

Abstractive summarization models are commonly trained using maximum likelihood estimation, which assumes a deterministic (one-point) target distribution in which an ideal model will assign all the probability mass to the reference summary. This assumption may lead to performance degradation during inference, where the model needs to compare several system-generated (candidate) summaries that have deviated from the reference summary. To address this problem, we propose a novel training paradigm which assumes a non-deterministic distribution so that different candidate summaries are assigned probability mass according to their quality. Our method achieves a new state-of-the-art result on the CNN/DailyMail (47.78 ROUGE-1) and XSum (49.07 ROUGE-1) datasets. Further analysis also shows that our model can estimate probabilities of candidate summaries that are more correlated with their level of quality.¹

1 Introduction

Neural methods for abstractive summarization (Rush et al., 2015; Nallapati et al., 2016; Chopra et al., 2016; Lewis et al., 2020; Zhang et al., 2020) formulate summarization as a sequence-to-sequence (Seq2Seq) problem (Sutskever et al., 2014), learning to generate the summary in an autoregressive manner. Such models are commonly trained with maximum likelihood estimation (MLE), maximizing predictive probability of the reference output given the gold sub-sequence before it. However, during inference the model must also generate the output based on possibly erroneous previous steps. This can hurt model performance, a phenomenon often called exposure bias (Bengio et al., 2015; Ranzato et al., 2016). To maintain reasonable performance even in the case of a sub-sequence with errors, we argue that the

¹We have made our code, results, and trained models publicly available at <https://github.com/yixinL7/BRIO>.

System	R-1	R-2	R-L	Acc.(%)
High	53.99	29.85	51.12	100.00
Low	33.48	10.85	30.45	0.00
BART	44.88	21.68	41.92	54.80
Ours	50.10	26.29	47.19	79.63

Table 1: Accuracy of different abstractive summarization systems w.r.t ranking the quality of candidate summaries on CNNDM dataset. **Acc.** stands for the frequency of the model assigning higher probabilities to better candidate summaries. The candidate summaries are generated by a pre-trained model (BART), and we select the best and the worst candidates (w.r.t. ROUGE scores) for each of the samples. **High** and **Low** represent the average performance of the best and worst candidates respectively. R-1/2/L are the ROUGE-1/2/L scores. The original BART only achieves 54.80% accuracy.

model must accurately estimate relative quality of different generated outputs, since effective inference requires comparison among these candidates.

To understand whether existing models can accurately perform such relative comparisons, we conducted a preliminary study on pre-trained BART (Lewis et al., 2020), first generating two candidate summaries from the model and observing whether a higher probability is assigned to the candidate with a higher ROUGE (Lin, 2004) score. As Tab. 1 shows, the accuracy is far from ideal. This is likely due to the fact that MLE training only encourages the model to assign high probability to the reference summary, and is agnostic about any relative comparison between non-reference summaries. However, we argue that it is also important for the order of model scores to be coordinated with the actual quality metrics by which the summaries will be evaluated – higher model scores should indicate better quality summaries. In the following we will refer to models that have such scores as “coordinated” for conciseness.

We introduce a training paradigm which requires the abstractive model to be able to be accurate with respect to predicting the tokens in the reference summaries and coordinated with respect to

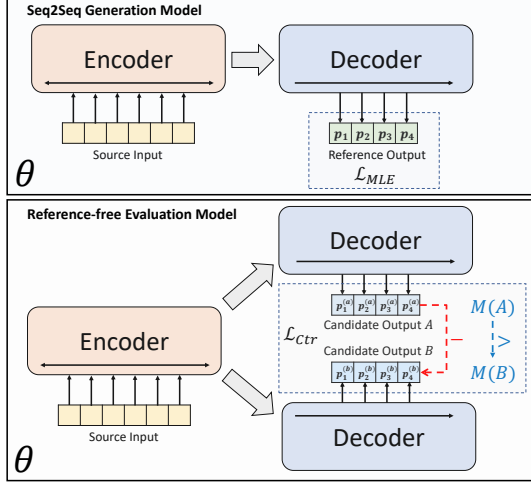


Figure 1: Comparison of MLE loss (\mathcal{L}_{MLE}) and the contrastive loss (\mathcal{L}_{Ctr}) in our method. MLE assumes a **deterministic** (one-point) distribution, in which the reference summary receives all the probability mass. Our method assumes a **non-deterministic** distribution in which system-generated summaries also receive probability mass according to their quality. The contrastive loss encourages the order of model-predicted probabilities of candidate summaries to be coordinated with the actual quality metric M by which the summaries will be evaluated. We assign the abstractive model a **dual** role – a single model could be used both as a **generation** model and a reference-free **evaluation** model.

the *candidate* summaries. In other words, we give the abstractive model a **dual role**: as a *generation* model, it generates the output summaries in an autoregressive way; as an *evaluation* model, it can be used to score the quality of candidate summaries by estimating a probability distribution over candidate outputs. The generation model is trained using the standard MLE loss, but to train the evaluation model we introduce a *contrastive loss* (Hadsell et al., 2006) defined over different candidate summaries generated by pre-trained abstractive models (Fig. 1), following previous work on ranking-based or contrastive learning (Hopkins and May, 2011; Zhong et al., 2020; Liu et al., 2021b).

Our main contribution is to change the target distribution of abstractive models from a one-point deterministic distribution assumed by MLE training to a non-deterministic distribution in which candidate summaries are also assigned probability mass according to their quality. The new SOTA performance on CNN/DailyMail (Hermann et al., 2015) and XSum (Narayan et al., 2018) datasets demonstrated the effectiveness of our method. Our in-depth analysis also found that the abstractive models trained using our method can estimate the candidate summary quality more accurately, in concert with the the objective of our training paradigm.

2 Neural Abstractive Summarization

The goal of abstractive summarization is to create a function g that takes a source document D and generates an appropriate summary S

$$S \leftarrow g(D) \quad (1)$$

Training Objective Neural abstractive summarization models aim to learn a neural model g that results in good summaries. Maximum likelihood estimation (MLE) is the standard training algorithm. It aims to maximize the likelihood of the reference summary S^* , i.e.,

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_i \log p_{g_\theta}(S^{*(i)} | D^{(i)}; \theta) \quad (2)$$

where θ denotes the parameters of g and p_{g_θ} denotes the probability distribution entailed by these parameters. The summation is over the training set and $\{D^{(i)}, S^{*(i)}\}$ is the i -th training sample.

For a specific sample $\{D^{(i)}, S^{*(i)}\}$, Eq. 2 is equivalent to minimizing the sum of negative log-likelihoods of the tokens $\{s_1^*, \dots, s_j^*, \dots, s_l^*\}$ in the reference summary S^* whose length is l , which is the cross-entropy loss:

$$\mathcal{L}_{xent} = - \sum_{j=1}^l \sum_{\mathbf{s}} p_{\text{true}}(s | D, S_{<j}^*) \log p_{g_\theta}(s | D, S_{<j}^*; \theta) \quad (3)$$

where $S_{<j}^*$ denotes the partial reference sequence $\{s_0^*, \dots, s_{j-1}^*\}$ and s_0^* is a pre-defined start token. p_{true} is a one-hot distribution under the standard MLE framework:

$$p_{\text{true}}(s | D, S_{<j}^*) = \begin{cases} 1 & s = s_j^* \\ 0 & s \neq s_j^* \end{cases} \quad (4)$$

In practice, label smoothing (Szegedy et al., 2016) is a widely used and effective technique that modifies the target distribution in Eq. 4 to a "soft" label by assigning probability mass β to other tokens:

$$p_{\text{true}}(s | D, S_{<j}^*) = \begin{cases} 1 - \beta & s = s_j^* \\ \frac{\beta}{N-1} & s \neq s_j^* \end{cases} \quad (5)$$

where N is the size of the dictionary.

Inference and Exposure Bias During inference, the abstractive model g is used to generate the candidate summary in an autoregressive manner. It is intractable to enumerate all the possible candidate outputs, so in practice methods such as beam search are used to reduce the search space.

One important step in search is estimating the probability of the next word s_t given the previous predicted sequence $S_{<t}$:

$$p_{g\theta}(s_t|D, S_{<t}; \theta) \quad (6)$$

Comparing Eq. 6 with Eq. 3, the major difference is that during inference the model makes new predictions based on its **own** previous predictions $S_{<t}$ instead of the *reference* $S_{<t}^*$. As a result, even if the generation model g achieves very high accuracy w.r.t. Eq. 3, once $S_{<t}$ starts to deviate from S^* , there is the risk that the performance of g will significantly degrade. This problem has been identified as the **exposure bias** (Bengio et al., 2015).

3 Coordinating Abstractive Models

Eq. 6 implies that the abstractive model g should be able to assign higher estimated probability to the better candidate summary during inference. However, this intuition is not directly captured in the standard MLE objective used in training – a model obtaining zero MLE loss would assign zero probability to any candidate summary different from the reference. **This is obviously improper for any task where multiple reasonable generations may exist** (Khayrallah et al., 2020), and also does not say anything about the ordering of two imperfect references. **We therefore advocate for making the alternative assumption that the probability of one candidate should be well-correlated with its quality as evaluated by an automatic metric M .** Since it is intractable to enumerate all the possible candidate outputs, we only require our model to be able to accurately predict the ranking order of a set of the most probable candidate summaries \hat{S} , which are its own beam search results. In order to achieve this objective, we slightly modify the conditions of Eq. 5, maintaining the general functional form, but instead specifying the *marginal* probability of the non-reference candidates S to be β , and **encouraging coordination of probabilities and qualities among non-reference candidates as follows**:

$$\begin{cases} p_{\text{true}\dagger}(S|D) = 1 - \beta & S = S^* \\ \sum_{S \in \hat{S}} p_{\text{true}\dagger}(S|D) = \beta & S \neq S^* \\ p_{\text{true}\dagger}(S_i|D) > p_{\text{true}\dagger}(S_j|D) & \forall S_i, S_j \in \hat{S}, \\ & M(S_i) > M(S_j) \end{cases} \quad (7)$$

We next describe precisely how we encourage coordination through *contrastive learning*.

Contrastive Learning for Coordination The candidate quality measure M can be defined in

many ways. In this work we define it as the ROUGE (Lin, 2004) score of a candidate summary S_i given the reference summary S^* . To coordinate a pre-trained abstractive model, we 1) use it to generate different candidate summaries with various levels of quality,² then 2) encourage the model to assign higher estimated probabilities to better candidates by fine-tuning the model with a **contrastive** loss, following the previous work (Hopkins and May, 2011; Zhong et al., 2020):

$$\mathcal{L}_{ctr} = \sum_i \sum_{j>i} \max(0, f(S_j) - f(S_i) + \lambda_{ij}) \quad (8)$$

where S_i and S_j are two different candidate summaries and $\text{ROUGE}(S_i, S^*) > \text{ROUGE}(S_j, S^*)$, $\forall i, j, i < j$. λ_{ij} is the margin multiplied by the difference in rank between the candidates, i.e., $\lambda_{ij} = (j - i) * \lambda$. $f(S_i)$ is the length-normalized estimated log-probability³

$$f(S) = \frac{\sum_{t=1}^l \log p_{g\theta}(s_t|D, S_{<t}; \theta)}{|S|^\alpha} \quad (9)$$

where α is the length penalty hyperparameter.

This loss gives the abstractive model a dual purpose, first as a reference-free *evaluation* model, which can be used in a **two-stage** summarization pipeline, where it is used to score the candidates generated by a pre-trained *generation* model and select the final output from them. However, since the autoregressive generation depends on both the **token-level prediction accuracy** and **sequence-level coordination**, the model fine-tuned with the contrastive loss alone can no longer be used as a *generation* model.

Multi-task Fine-tuning Following Edunov et al. (2018), we combine the contrastive (Eq. 8) and cross-entropy (Eq. 3) losses to preserve the **generation** ability of the pre-trained abstractive model:

$$\mathcal{L}_{mul} = \mathcal{L}_{xent} + \gamma \mathcal{L}_{ctr} \quad (10)$$

where γ is the weight of the contrastive loss. We note that the contrastive and the cross-entropy loss can effectively complement each other – since the contrastive loss is defined on the sequence level, the token-level cross-entropy loss serves as a normalization to ensure that the model could assign balanced probability mass across the whole sequence.

²This is achieved by using diverse beam search (Vijayakumar et al., 2018).

³We length-normalize as it is standard in comparing hypotheses in neural sequence generation (Cho et al., 2014).

4 Related Work

Training Methods of Seq2Seq Models In order to align the training objective and evaluation metric, structured losses have been used for the Seq2Seq model training. Among them, margin-based losses (Herbrich et al., 1999; Taskar et al., 2004; Gimpel and Smith, 2010), which require the model to assign higher probability to the better output, are a major category. Many margin-based losses used in modern seq2seq models (Wiseman and Rush, 2016; Edunov et al., 2018) assume a deterministic (one-point) distribution: a model can achieve zero loss if it can assign a much higher probability to the (pseudo)-reference, regardless of relative comparisons of other candidate summaries. By contrast, our method has a non-deterministic assumption (Eq. 7), which focuses on the pair-wise ranking of a set of candidate summaries.

One main challenge of directly optimizing a Seq2Seq model with quality scores of the output is that the discrete sampling process makes the loss non-differentiable. To circumvent this problem, reinforcement learning has been used to reformulate the conditional text generation tasks (Ranzato et al., 2016; Bahdanau et al., 2016; Li et al., 2016; Paulus et al., 2018; Li et al., 2019). Compared to this school of methods, our method is based on supervised learning, and it is more stable and less sensitive to the design choices (e.g. reward shaping), which are well-known challenges of reinforcement learning methods. Minimum risk training (Shen et al., 2016; Wieting et al., 2019) and other on-line sampling based methods (Bengio et al., 2015; Norouzi et al., 2016; Zhang et al., 2019) belong to another school of methods used to circumvent the problem of non-differentiability. However, they also exhibit similar problems of stability as reinforcement learning.

Contrastive Learning Recently, contrastive learning (Hadsell et al., 2006) has been introduced into several conditional text generation tasks, such as machine translation (Yang et al., 2019; Pan et al., 2021), text summarization (Cao and Wang, 2021; Xu et al., 2021; Sun and Li, 2021), and other tasks (Uehara et al., 2020; Cho et al., 2021; Lee et al., 2021b). Among these application scenarios, most work deployed contrastive learning in the latent representation space, following the framework proposed in Chen et al. (2020). However, in this work we adopt contrastive learning over the discrete space of the generated texts.

Besides, instead of constructing the contrastive learning examples by rule-based methods (e.g. perturbing the reference output), we use the generation models to construct the examples, which makes the contrastive learning task closer to the generation task. Sun and Li (2021) also adopted contrastive learning on the generated texts. However, their formulation belongs to the margin-based losses. We have discussed the difference between our method and the margin-based losses in the previous paragraphs.

Discriminative Reranking Discriminative reranking has been widely studied for conditional generation tasks (Shen et al., 2004; Och et al., 2004; Wan et al., 2015; Mizumoto and Matsumoto, 2016). Some recent works (Liu and Liu, 2021; Lee et al., 2021a) have also explored discriminative reranking of candidates from neural natural language generation models, which adopt large pre-trained language models (e.g. BERT (Devlin et al., 2019)) as the reranker. In this work, we factorize the Seq2Seq model (e.g., BART) trained on the same dataset as the reranking model, which maximizes the parameter sharing across two stages. Besides, our approach contributes an instance of leveraging large pre-trained Seq2Seq models as a quality estimation model (Yuan et al., 2021).

5 Experiments

5.1 Experimental Settings

Datasets We mainly use three datasets in our experiments (statistics in Appendix A).

CNN⁴ (Hermann et al., 2015) is a large scale news dataset. Following Nallapati et al. (2016), we treat the news articles as the source documents and the associated highlights as the summaries.

XSum⁵ (Narayan et al., 2018) is a highly abstractive dataset of articles from the British Broadcasting Corporation (BBC).

NYT⁶ (Sandhaus, 2008) contains articles from the New York Times and the associated summaries. We follow Kedzie et al. (2018) for data preprocessing and splitting, and use the associated archival abstracts as the summaries.

Baselines We choose a variety of related models with strong performance as baselines. BART (Lewis et al., 2020) and PEGASUS (Zhang et al., 2020) are both large pre-trained Seq2Seq

⁴<https://cs.nyu.edu/~kcho/DMQA/>

⁵<https://github.com/EdinburghNLP/XSum>

⁶<https://catalog.ldc.upenn.edu/LDC2008T19>

LMs standard in the literature. **GSum** (Dou et al., 2021) is built on BART, and improves performance by using additional guidance from an extractive summarizer. **SimCLS** (Liu and Liu, 2021) introduces a two-stage framework where the pre-trained BART model is used to generate candidates and a pre-trained RoBERTa (Liu et al., 2019) model is fine-tuned as an evaluation model to score the candidate summaries and select from them. It achieves state-of-the-art performance on both CNNDM and XSum. **GOLD** (Pang and He, 2021) uses offline reinforcement learning to train the BART model by treating the reference summaries as the demonstrations, a different formulation that can also improve the performance of the original BART. **SeqCo** (Xu et al., 2021) and **ConSum** (Sun and Li, 2021) are two recent methods that aim to leverage contrastive learning to improve the performance of the abstractive summarization model (BART).

Implementation Details In the following experiments, we use either BART or PEGASUS as a backbone. We label our proposed methods **BRIO**, with two variants: (1) **BRIO-Ctr** is fine-tuned with the contrastive loss (Eq. 8) only; (2) **BRIO-Mul** is fine-tuned with the multi-task loss (Eq. 10). We use **BRIO-Ctr** as an evaluation model that scores different candidate summaries generated by a Seq2Seq abstractive model and selects the final output from them, and **BRIO-Mul** as a standard Seq2Seq model that takes the source documents as input and generates the output in an autoregressive manner. Further details are in Appendix B.

5.2 Results

The results are shown in Tab 2. For CNNDM and NYT we use **BART** as the backbone model while for XSum we use the pre-trained **PEGASUS** model as our base model since it achieves better performance than BART. We have the following observations:

(1) BRIO-Ctr outperforms *SimCLS*, its counterpart as an evaluation model in a two-stage summarization framework. Specifically, both BRIO-Ctr and SimCLS are used to score the candidate summaries generated by a Seq2Seq abstractive model (BART). The final outputs are selected based on those scores. We attribute BRIO-Ctr’s superior performance to its use of the same model architecture (BART) for both candidate generation and scoring, while SimCLS uses RoBERTa as the evaluation model. As a result, BRIO-Ctr maximizes the parameter sharing between the two stages, and preserves the power of the Seq2Seq model pre-trained

System	R-1	R-2	R-L
CNNDM			
BART*	44.16	21.28	40.90
PEGASUS*	44.17	21.47	41.11
GSum*	45.94	22.32	42.48
ConSum*	44.53	21.54	41.57
SeqCo*	45.02	21.80	41.75
GOLD- <i>p</i> *	45.40	22.01	42.25
GOLD- <i>s</i> *	44.82	22.09	41.81
SimCLS*	46.67	22.15	43.54
BART [‡]	44.29	21.17	41.09
BRIO-Ctr	47.28 [†]	22.93 [†]	44.15 [†]
BRIO-Mul	47.78[†]	23.55[†]	44.57[†]
XSum			
BART*	45.14	22.27	37.25
PEGASUS*	47.21	24.56	39.25
GSum*	45.40	21.89	36.67
ConSum*	47.34	24.67	39.40
SeqCo*	45.65	22.41	37.04
GOLD- <i>p</i> *	45.75	22.26	37.30
GOLD- <i>s</i> *	45.85	22.58	37.65
SimCLS*	47.61	24.57	39.44
PEGASUS [‡]	47.46	24.69	39.53
BRIO-Ctr	48.13 [†]	25.13 [†]	39.84 [†]
BRIO-Mul	49.07[†]	25.59[†]	40.40[†]
NYT			
BART [‡]	55.78	36.61	52.60
BRIO-Ctr	55.98	36.54	52.51
BRIO-Mul	57.75[†]	38.64[†]	54.54[†]

Table 2: Results on CNNDM, XSum and NYT. On NYT we only reported our own results due to different data pre-processing. [†]: significantly better than the baseline model ($p < 0.01$). *: results reported in the original papers. [‡]: results from our own evaluation script. R-1/2/L are the ROUGE-1/2/L F₁ scores.

on the same dataset.

(2) BRIO-Mul is able to establish the new state-of-the-art performance on CNNDM. Notably, the previous state-of-the-art model, *GSum*, takes additional guidance as input and needs a separate encoder to encode the guidance information, while BRIO-Mul uses the same parameterization of BART. Compared to other methods (ConSum, SeqCo, GOLD) that aim to improve upon BART, BRIO-Mul performs much better, showing the effectiveness of our training method.

(3) Since on XSum we use PEGASUS instead of BART as the base model, the result shows that our method is not restricted to the specific choice of the base model.

5.3 Analysis

We further perform some in-depth analyses from diverse perspectives on the CNNDM dataset to gain more insights into our proposed method.

Coefficient (γ)	R-1	R-2	R-L
0 (BART)	44.29	21.17	41.09
0.1	45.08	21.63	41.71
1	46.01	22.22	42.68
2	46.36	22.79	43.07
5	46.91	23.03	43.63
10	47.22	23.31	43.94
100	47.78	23.55	44.57
1000	46.83	22.17	43.68
$+\infty$ (BRIO-Ctr)	47.28	22.93	44.15

Table 3: Model performance with different γ coefficients weighting the contrastive loss (Eq. 10) on CNNDM. BRIO-Ctr is trained with the contrastive loss only, which no longer preserves its generation ability. We report its performance when it is used as an evaluation model to select from candidate summaries. R-1/2/L are the ROUGE-1/2/L F_1 scores.

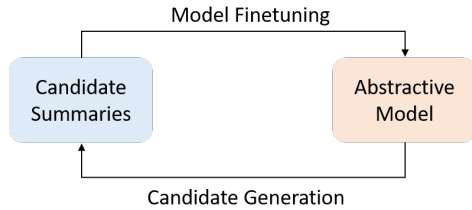


Figure 2: Loop of candidate generation and model finetuning.

System	R-1	R-2	R-L
BART	44.29	21.17	41.09
BRIO-Mul	47.78	23.55	44.57
BRIO-Loop	48.01[†]	23.80[†]	44.67[†]

Table 4: Results on CNNDM when the pre-trained model are fine-tuned twice. **BRIO-Loop** is trained on the candidates generated by **BRIO-Mul**. [†]: significantly better than the baseline (BART) ($p < 0.01$). R-1/2/L are ROUGE-1/2/L F_1 scores.

Coefficients of the Multi-Task Loss The multi-task loss (Eq. 10) used to train our model contains two parts: the cross-entropy loss and the contrastive loss. As shown in Tab. 3, as the weight of the contrastive loss (γ) increases, the model’s performance improves. However, the cross-entropy loss is still necessary to preserve the model’s ability as a generation model. We argue that this is because the token level accuracy is still important during the autoregressive generation process, where the individual tokens are predicted sequentially. In addition, we also found that the model tends to achieve the best performance (w.r.t the ROUGE scores on the development set) faster with a higher γ . Specifically, it requires less than one entire epoch to achieve the best performance on CNNDM, making our approach an efficient fine-tuning method.

Generation-Finetuning as a Loop Since the fine-tuned model (BRIO-Mul) is still able to gen-

Beams	BART		BRIO-Mul	
	R-1	R-2	R-1	R-2
4	44.29	21.17	47.78	23.55
10	43.83	20.76	47.98	23.81
20	43.53	20.49	48.07	23.92
50	43.06	20.05	48.18	24.01
100	42.79	19.76	48.23	24.09

Table 5: Results on CNNDM with different beam widths (the number of beams) used in beam search. The default beam width is 4. R-1/2 are the ROUGE-1/2 F_1 scores.

erate, we can use it to generate a new set of candidates in the same way as we used the pre-trained BART model, and continue fine-tuning it on this newly created set of candidates (Och, 2003). Fig. 2 illustrates this iterative process. The results shown in Tab. 4 illustrate that this new model (BRIO-Loop) outperforms BRIO-Mul. Besides, the model reached the best performance very quickly, showing the potential of adopting our method in an online framework where the new candidates are dynamically generated from the current model. We leave this direction for future work.

Increasing the Beam Width While theoretically a larger beam width (i.e. the number of candidates maintained during beam search) would allow more candidates to be considered and therefore increase the upper bound of the performance, in practice model performance may be lower if the beam width is too large. The reason for this phenomenon is closely related to the low sequence-level coordination of the generator. Specifically, increasing the beam width may introduce candidates with lower quality (Stahlberg and Byrne, 2019), and the generator may not be able to differentiate them from high-quality candidates.

In Tab. 5, we compare the performance of the pre-trained BART and our model (BRIO-Mul) with different beam widths used during inference. We observe that the performance of BART goes down as the beam width increases. On the other hand, our model is able to achieve better performance with a larger number of beams, demonstrating that our training method can improve the coordination of the model by encouraging the model to assign estimated probabilities to candidate summaries well-correlated with their quality.

Training with Different Evaluation Metrics In the previous experiments, we used ROUGE as the evaluation metric to define the target ordering of the candidate summaries (Eq. 7). To evaluate our method’s performance beyond ROUGE,

System	R-1	R-2	R-L	BS
BART	44.29	21.17	41.09	27.38
BRIO-Mul (R)	47.78	23.55	44.57	32.11
BRIO-Mul (B)	47.53	23.22	44.37	32.59

Table 6: Results on CNNDM using different evaluation metrics as M in Eq.7. BRIO-Mul (R) is trained with candidate summaries ordered by ROUGE scores, while BRIO-Mul (B) is trained with candidate summaries ordered by BERTScore. R-1/2/L are ROUGE-1/2/L F_1 scores. BS denotes BERTScore.

System	Unigram	Bigram
Reference	.1110	.4865
BART	.0101	.0924
BRIO-Mul	.0262	.2381

Table 7: Ratio of novel n -grams of different models on CNNDM. Novel n -grams are those that appear in the summaries but not in the source documents.

we use a model-based semantic similarity metric, **BERTScore** (Zhang* et al., 2020),⁷ as the evaluation metric M in Eq.7 to compare the performance of different candidate summaries. Then, we trained another version of BRIO-Mul based on the order of candidate summaries calculated by BERTScore.

The results in Tab. 6 show that (1) Our model can significantly improve the model performance when either ROUGE or BERTScore is used as the target evaluation metric for ordering candidate summaries. This suggests that it is possible to use our method to optimize any specific target metric, making our method an alternative to reinforcement learning or minimum risk training. (2) Our model that is trained on one evaluation metric (e.g. BERTScore) also achieves improvement on another metric (e.g. ROUGE) compared with the baseline model, which indicates that the improvement made by our model is not from exploiting the potential weaknesses of individual metrics. Besides, this result also demonstrates a non-trivial degree of agreement between ROUGE and BERTScore.

Novel n -grams We compare the ratio of novel n -grams in reference, BRIO-Mul’s, and BART’s summaries. As Tab. 7 shows, our model is more “abstractive” compared to BART, although reference summaries still contain more novel n -grams. This is likely due to the fact that our model is optimized at the sequence-level, allowing more freedom for paraphrasing and compression.

We further investigate the relation of the “abstractiveness” and model performance by com-

⁷https://github.com/Tiiiger/bert_score. We use its default version for English texts.

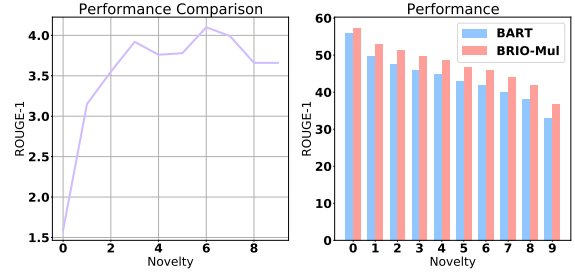


Figure 3: Performance comparison (BART v.s. BRIO-Mul) w.r.t. reference summary novelty. The x-axis represents different buckets of test examples grouped by reference summary novelty (Eq. 11). Larger x-coordinates correspond to examples of which the reference summaries have higher novelty. The left figure shows the performance improvement of our model compared with the baseline model, while the right one shows model performance.

	Own	PEGASUS
BART	.0470	.1205
BRIO-Mul	.1839[†]	.2768[†]

Table 8: Rank Correlation between the model’s estimated probabilities of the candidate summaries and the quality scores (ROUGE) of the candidate summaries on CNNDM. **Own** stands for the candidates generated by the models themselves, while **PEGASUS** stands for the candidates generated by the pre-trained PEGASUS model. [†]: significantly better than the baseline model (BART) ($p < 0.01$).

paring our model (BRIO-Mul) with the baseline model (BART) on different buckets of test examples grouped by the “novelty” of the reference summaries,⁸ i.e.,

$$\text{Novelty}(D, S^*) = \frac{\sum_{g \in G_{S^*}} \mathbb{1}(g \notin G_D)}{|G_{S^*}|} \quad (11)$$

where D and S^* are the source document and reference summary respectively, G_D and G_{S^*} are the sets of bigrams in D and S^* , $\mathbb{1}$ is the indicator function. The results in Fig. 3 show that when novelty is higher, (1) all models’ performance decreases; (2) our model achieves larger improvement over the baseline model.

Rank Correlation We computed the rank correlation between the **estimated probabilities** of the candidate summaries calculated by the generators and the **quality scores** of the candidate summaries. We use Eq. 9 to calculate the estimated probabilities⁹ and we use ROUGE-1 as the quality score metric of the candidate summaries. We calculate

⁸The calculation is performed using ExplainaBoard (Liu et al., 2021a). <https://github.com/neulab/ExplainaBoard>.

⁹We found the value of the length penalty factor α in Eq. 9 by maximizing the rank correlation on the validation set.

Dataset	System	ECE	Acc	Conf
CNNDM	BART	.4097	.3711	.7365
	BRIO-Mul	.2719	.4271	.6652
XSum	PEGASUS	.2369	.4688	.6990
	BRIO-Mul	.1423	.4744	.5881

Table 9: Expected Calibration Error (ECE), accuracy (Acc) and confidence (Conf) on the test set of CNNDM and XSum.

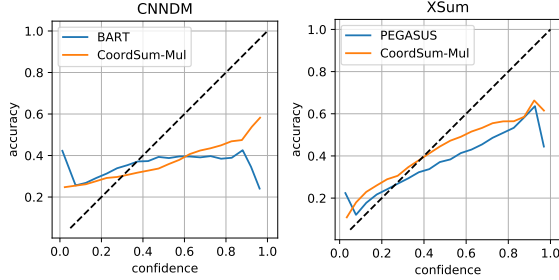


Figure 4: Reliability graphs on the CNNDM and XSum datasets. The accuracy of model’s predictions is plotted against the model’s confidence on these predictions.

Spearman’s rank correlation for each sample, and use the average score as the overall correlation,

We investigated two specific settings: 1) ranking candidate summaries generated by a different model (PEGASUS); 2) ranking candidate summaries generated by themselves (BART & BRIO-Mul). We use 16 candidates in total for calculation. As Tab. 8 shows, our model achieves better rank correlation on the candidate summaries generated by both itself and the independent model. This suggests that our model can better estimate the quality of candidate summaries.

5.4 Token-level Calibration

Calibration requires that a model’s confidence on its predictions is equal to the accuracy of these predictions (Guo et al., 2017). Previous work (Müller et al., 2019; Kumar and Sarawagi, 2019; Wang et al., 2020) has found that a more calibrated text generation model tends to have better performance, and techniques like label smoothing can improve both the token-level calibration and sequence-level accuracy (i.e. the ability of generating better results). One intuitive explanation of this phenomenon is to interpret the model’s estimated probability of a generated summary as the product of the model’s confidences on a series of token-level predictions. Then, since a more calibrated model’s confidence estimates better the accuracy of its predictions, the model’s estimated **probability** of one sequence should be more indicative of

the **quality** of this sequence, which is essential for the beam search during inference. However, the relation of token-level calibration and sequence-level performance remains inconclusive (Müller et al., 2019).¹⁰ For example, a generator that always predicts a uniform distribution over all tokens would be perfectly calibrated, however, such a model would not generate high-quality outputs.

We investigate this relation from the opposite direction by evaluating whether our model (BRIO-Mul), which is trained to have better **sequence-level** performance, would also be more calibrated at the **token-level** compared with the baseline models that are trained using MLE and label smoothing. We follow previous work by using the Expected Calibration Error (Naeini et al., 2015) (ECE) as the evaluation metric of calibration:

$$ECE = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)| \quad (12)$$

where the samples are grouped into M equal-width buckets by confidence (conf), B_m denotes the m -th bucket, and n is the total number of samples. Following Wang et al. (2020), we evaluate model calibration on the **system-generated** summaries during inference and use the tercom toolkit¹¹ to assign labels (correct/incorrect) to the system-generated summaries based on the reference summaries.

The results in Tab. 9 show that BRIO-Mul is better calibrated compared to BART, suggesting that our method helps to improve the token-level calibration by explicitly encouraging the model to have more accurate sequence-level probability estimations. The reliability graph is shown in Fig. 4. We found that (1) abstractive models are generally over-confident on their own predictions, (2) models are generally more calibrated on XSum than CNNDM. This is likely due to the fact that XSum has shorter summaries therefore it is less likely to be affected by the exposure bias.

5.5 Few-shot Fine-tuning

The training paradigm proposed in this paper may be extended to any Seq2Seq model. However, it can be a non-trivial overhead to generate the candidate summaries using large neural models on the entire training set. On the other hand, recent work (Raffel et al., 2020; Zhang et al., 2020; Schick and Schütze,

¹⁰In general, better token-level calibration doesn’t guarantee better sequence-level performance.

¹¹<http://cs.umd.edu/~snover/tercom/>

System	Summary
Reference	chelsea forward tammy abraham nets first-half double for chelsea. dominic solanke adds a third late on as chelsea look set to win trophy. manchester city struggle without injured star thierry ambrose. read: mourinho warns his young chelsea players he can not play them all. click here to read our match report from man city 's academy stadium.
BART	tammy abraham scored twice in the first half to give chelsea the lead. isaac buckley-ricketts levelled the game for manchester city. dominic solanke scored late on to put a gloss on the scoreline. click here to read sportmail's player ratings from the youth cup final.
BRIO-Mul	chelsea beat manchester city 3-1 in the youth cup final at the etihad stadium. tammy abraham scored twice in the first half to give chelsea the lead. dominic solanke scored late on to seal the win for the home side.
Reference	alejandro valverde won ahead of julian alaphilippe and michael albasini. chris froome finished 123rd after a crash during the final 12 kilometres. team sky's sports director gabriel rasch praised froome for finishing. rasch said froome was 'banged up' but expects to ride tour de romandie.
BART	movistar rider alejandro valverde won fleche wallonne on wednesday. team sky's chris froome fell in the final 12km but finished the race. philippe gilbert pulled out of the race after a bad crash 50km from the end. click here for more cycling news.
BRIO-Mul	alejandro valverde defended his fleche wallonne title in belgium on wednesday. movistar rider finished ahead of julian alaphilippe and michael albasini. team sky's chris froome fell in the final 12km of the race but finished in 123rd. froome was involved in a crash but finished the race despite being 'banged up'
Reference	manuel pellegrini won the premier league and capital one cup last season. city currently sit fourth in the league table - 12 points behind chelsea. pellegrini's contract expires at the end of the 2015-16 season. city players have been impressed with vieira's work with the youth team. pep Guardiola is city's first-choice to succeed pellegrini at the etihad.
BART	manuel pellegrini's future at manchester city is under scrutiny. patrick vieira is highly-respected among the city players. city's first-choice managerial option is bayern munich boss pep Guardiola. click here for all the latest manchester city news. click here for more premier league news.
BRIO-Mul	manchester city players have backed patrick vieira to replace manuel pellegrini as manager of the club. the frenchman is highly-respected among the players at the etihad stadium. pellegrini's future at the club is under scrutiny after a disappointing season. city's first-choice manager is current bayern munich boss pep Guardiola.

Table 10: Case Study on CNNDM. BRIO-Mul learns to ignore the noise pattern ("click here") while BART cannot.

Dataset	System	R-1	R-2	R-L
CNNDM	BART	44.29	21.17	41.09
	BRIO-Few	45.81	21.91	42.61
XSum	PEGASUS	47.46	24.69	39.53
	BRIO-Few	47.95	24.89	39.71

Table 11: Few-shot Fine-tuning. **BRIO-Few** is trained on only 100/1000 training examples on CNNDM and XSum respectively. R-1/2/L are ROUGE-1/2/L F_1 scores.

2021; Fabbri et al., 2021) has shown that few-shot learning can be an effective fine-tuning method of pre-trained models for text generation tasks.

Therefore, we investigate our model’s performance in a few-shot setting. Specifically, we randomly sample 100/1000 examples from the training set of CNNDM/XSum, and fine-tune the models that are pre-trained using MLE loss on those examples. More training details can be found in Appendix C. The results are shown in Tab. 11. All experiments are repeated three times, and the reported results are the average performance. The results indicate that our model can achieve improvement over the baseline model under the few-shot learning setting with a small computational overhead.

5.6 Case Study on CNNDM

Tab. 10 presents an interesting pattern we observed when comparing the results of BRIO-Mul and BART, which demonstrates that our method helps the abstractive model to filter out noise patterns in the original data. Specifically, some of the reference summaries (331/11490) in CNNDM contains

the phrase “click here”, pointing to a hyperlink, and 103 source documents also contain this phrase. BART picked up this pattern, and generates this phrase in 96 output summaries. On the contrary, our model learns to ignore this noise pattern and never generated it across the whole test set, likely because it identified that generated candidates with this pattern rarely achieve a high ROUGE score, and downweighted the probability accordingly.

6 Conclusion and Future Work

In this work, we presented a new training paradigm that assigns candidate outputs probability mass according to their quality using contrastive learning. While our method has achieved significant improvement on abstractive summarization, we note several directions for the future work to explore. First, since our method makes no assumptions specifically about the summarization task, it can be extended to other conditional text generation tasks such as machine translation. Second, it is possible to apply our method in a reinforcement learning setting, where the candidate summaries are dynamically generated. Finally, in experiments we only used diverse beam search to generate the candidate summaries, but it is likely that other candidate generation methods could yield further improvements.

Acknowledgements

We thank the anonymous reviewers for valuable feedback and helpful suggestions.

References

- Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. [An actor-critic algorithm for sequence prediction](#). *CoRR*, abs/1607.07086.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, page 1171–1179, Cambridge, MA, USA. MIT Press.
- 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*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. [A simple framework for contrastive learning of visual representations](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [On the properties of neural machine translation: Encoder-decoder approaches](#). In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.
- Woon Sang Cho, Yizhe Zhang, Sudha Rao, Asli Celikyilmaz, Chenyan Xiong, Jianfeng Gao, Mengdi Wang, and Bill Dolan. 2021. [Contrastive multi-document question generation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 12–30, Online. Association for Computational Linguistics.
- Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. [Abstractive sentence summarization with attentive recurrent neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. 2018. [Classical structured prediction losses for sequence to sequence learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.
- Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021. [Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.
- Kevin Gimpel and Noah A. Smith. 2010. [Softmax-margin CRFs: Training log-linear models with cost functions](#). In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736, Los Angeles, California. Association for Computational Linguistics.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. [Dimensionality reduction by learning an invariant mapping](#). In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR ’06*, page 1735–1742, USA. IEEE Computer Society.
- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 1999. Support vector learning for ordinal regression. In *International Conference on Artificial Neural Networks*, pages 97–102.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read

- and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.
- Mark Hopkins and Jonathan May. 2011. [Tuning as ranking](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. [Content selection in deep learning models of summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.
- Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. [Simulated multiple reference training improves low-resource machine translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Aviral Kumar and Sunita Sarawagi. 2019. [Calibration of encoder decoder models for neural machine translation](#). *CoRR*, abs/1903.00802.
- Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021a. [Discriminative reranking for neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7250–7264, Online. Association for Computational Linguistics.
- Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021b. [Contrastive learning with adversarial perturbations for conditional text generation](#). In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. [Deep reinforcement learning for dialogue generation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.
- Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. [Deep reinforcement learning with distributional semantic rewards for abstractive summarization](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaichen Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021a. [ExplainaBoard: An explainable leaderboard for NLP](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 280–289, Online. Association for Computational Linguistics.
- 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, Zi-Yi Dou, and Pengfei Liu. 2021b. [RefSum: Refactoring neural summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1437–1448, Online. Association for Computational Linguistics.
- 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 (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.
- Tomoya Mizumoto and Yuji Matsumoto. 2016. [Discriminative reranking for grammatical error correction with statistical machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1133–1138, San Diego, California. Association for Computational Linguistics.
- Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. [When does label smoothing help?](#) In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated

- probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2901–2907. AAAI Press.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Mohammad Norouzi, Samy Bengio, zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. 2016. [Reward augmented maximum likelihood for neural structured prediction](#). In *Advances in Neural Information Processing Systems*, volume 29, pages 1723–1731. Curran Associates, Inc.
- Franz Josef Och. 2003. [Minimum error rate training in statistical machine translation](#). In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.
- Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. [A smorgasbord of features for statistical machine translation](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. [Contrastive learning for many-to-many multilingual neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.
- Richard Yuanzhe Pang and He He. 2021. [Text generation by learning from demonstrations](#). In *International Conference on Learning Representations*.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *International Conference on Learning Representations*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. [Sequence level training with recurrent neural networks](#). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.
- Evan Sandhaus. 2008. *The New York Times Annotated Corpus*. LDC corpora. Linguistic Data Consortium.
- Timo Schick and Hinrich Schütze. 2021. [Few-shot text generation with natural language instructions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. [Discriminative reranking for machine translation](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. [Minimum risk training for neural machine translation](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.
- Felix Stahlberg and Bill Byrne. 2019. [On NMT search errors and model errors: Cat got your tongue?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.
- Shichao Sun and Wenjie Li. 2021. [Alleviating exposure bias via contrastive learning for abstractive text summarization](#). *CoRR*, abs/2108.11846.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Proceedings of the 27th International Conference*

- on *Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.
- C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. [Rethinking the inception architecture for computer vision](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Los Alamitos, CA, USA. IEEE Computer Society.
- Ben Taskar, Carlos Guestrin, and Daphne Koller. 2004. [Max-margin markov networks](#). In *Advances in Neural Information Processing Systems*, volume 16. MIT Press.
- Yui Uehara, Tatsuya Ishigaki, Kasumi Aoki, Hiroshi Noji, Keiichi Goshima, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2020. [Learning with contrastive examples for data-to-text generation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2352–2362, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. [Diverse beam search for improved description of complex scenes](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Xiaojuan Wan, Ziqiang Cao, Furu Wei, Sujian Li, and M. Zhou. 2015. Multi-document summarization via discriminative summary reranking. *ArXiv*, abs/1507.02062.
- Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. [On the inference calibration of neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.
- John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. [Beyond BLEU: training neural machine translation with semantic similarity](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.
- Sam Wiseman and Alexander M. Rush. 2016. [Sequence-to-sequence learning as beam-search optimization](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2021. [Sequence level contrastive learning for text summarization](#). *CoRR*, abs/2109.03481.
- Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. [Reducing word omission errors in neural machine translation: A contrastive learning approach](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. [BARTScore: Evaluating generated text as text generation](#). In *Thirty-Fifth Conference on Neural Information Processing Systems*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. [Bridging the gap between training and inference for neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.
- Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Datasets Statistics

Datasets	# Examples			Avg. Words	
	Train	Valid	Test	Doc.	Sum.
CNNDM	287K	13K	11K	791.6	55.6
XSum	203K	11K	11K	429.2	23.3
NYT	44K	5.5K	6.4K	1320.2	123.4

Table 12: Datasets Statistics.

B Implementation Details

We use diverse beam search (Vijayakumar et al., 2018) to generate 16 candidates for each data sample. On CNNDM and XSum, we use the pre-trained BART¹² and PEGASUS¹³ models from the *Transformers* (Wolf et al., 2020) library as the base abstractive models for candidate summary generation and model finetuning respectively. On NYT, we first fine-tuned a BART model¹⁴ with MLE training as the base abstractive model, since our data pre-processing is slightly different from the previous work and there are no available pre-trained checkpoints. We use 4 NVIDIA RTX 3090 GPUs for the model training, and the average running time for one epoch is around 20 hours. We use the Adam optimizer (Kingma and Ba, 2015) with learning rate scheduling for the model training:

$$lr = 2 \times 10^{-3} \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5})$$

where warmup denotes the warmup steps, which is set to 10000, step is the number of updating steps, lr is the learning rate.

We set the length penalty factor α in the scoring function (Eq. 9) to the same value as used in the original beam search. We search the value of the margin λ in the contrastive loss (Eq. 8) within the range $[1 \times 10^{-5}, 1]$, and decide the value based on the model performance on the validation set. We also performed extensive search for the coefficient γ in Eq. 10. The specific hyper-parameter setting is reported in Tab. 13.

We use the standard ROUGE (Lin, 2004) Perl package¹⁵ for evaluation. The command line parameters are ‘-c 95 -r 1000 -n 2 -m’. Before the

¹²The checkpoint is “facebook/bart-large-cnn”, containing around 400M parameters.

¹³The checkpoint is “google/pegasus-xsum” containing around 568M parameters.

¹⁴The checkpoint is “facebook/bart-large”.

¹⁵<https://github.com/summanlp/evaluation/tree/master/ROUGE-RELEASE-1.5.5>

Datasets	λ (Eq. 8)	α (Eq. 9)	γ (Eq. 10)
CNNDM	0.001	2.0	100
XSum	0.1	0.6	100
NYT	0.001	2.0	100

Table 13: Hyper-parameter Setting.

ROUGE evaluation, the reference summaries and system outputs are lower-cased and tokenized.¹⁶

C Details of Few-shot Fine-tuning

On CNNDM, we randomly select 100 examples from the training set for fine-tuning. On XSum, we found that at least 1000 examples are needed for the model to achieve better performance compared to the baseline model. All experiments are repeated three times. We randomly select 1000 examples from the original validation set for hyper-parameter selection. We use the Adam optimizer with the learning rate set to 1×10^{-6} . The model is trained for 15 epochs on CNNDM and 10 epochs on XSum.

¹⁶PTB tokenizer is used for tokenization. <https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/process/PTBTokenizer.html>