Learning From the Source Document: Unsupervised Abstractive Summarization

Haojie Zhuang[1], **Wei Emma Zhang**[1], **Jian Yang**[2], **Congbo Ma**[1], **Yutong Qu**[1], **Quan Z. Sheng**[2]
[1]The University of Adelaide, Adelaide, Australia
[2]Macquarie University, Sydney, Australia
{haojie.zhuang,wei.e.zhang,congbo.ma,yutong.qu}@adelaide.edu.au
{jian.yang,michael.sheng}@mq.edu.au

Abstract

Most of the state-of-the-art methods for abstractive text summarization are under supervised learning settings, while heavily relying on high-quality and large-scale parallel corpora. In this paper, we remove the need for reference summaries and present an unsupervised learning method **SCR** (**S**ummarize, **C**ontrast and **R**eview) for abstractive summarization, which leverages *contrastive learning* and is the first work to apply contrastive learning for unsupervised abstractive summarization. Particularly, we use the true source documents as *positive source document examples*, and strategically generated fake source documents as *negative source document examples* to train the model to generate good summaries. Furthermore, we consider and improve the writing quality of the generated summaries by guiding them to be similar to human-written texts. The promising results on extensive experiments show that SCR outperforms other unsupervised abstractive summarization baselines, which demonstrates its effectiveness.

Reference summary	Julia Vakulenko has reached her first final on the WTA Tour at Bell Challenge . The Ukrainian third seed will face Lindsay Davenport after beating Julie Ditty. Former world No. 1 Davenport defeated Russian second seed Vera Zvonareva .
Human-written source document	After eliminating Vera Zvonareva with 7-6, 6-4, 2-6, 6-3, Lindsay Davenport, two-time Olympic medalist and three-time World champion, would face Julia Vakulenko, which is their first duel in the final... She had been in the trouble of shoulders injury ...
Reference source document	... Third seed Julia Vakulenko will face comeback queen Lindsay Davenport in her first WTA Tour final at the Bell Challenge on Sunday ... The three-time Grand Slam winner has surged back up the rankings from 234th to 126th after winning her comeback tournament in Bali and then reaching the last four in Beijing ...

Table 1: Given a reference summary, human-written source document, as well as the reference source document. The words in blue are expressing the same message (i.e., paraphrasing) with the reference summary, while the words in red are hallucinations written by human. Green words are information only in the reference source document while human missed it.

1 Introduction

Given a source document, summarization aims at generating a shorter text version that retains the most salient information (See et al., 2017; Rush et al., 2015; Nallapati et al., 2016a; Bhandari et al., 2020; Ma et al., 2022). While extractive summarization methods directly copy and group important sections (e.g., words, phrases or sentences) from the source documents (Dorr et al., 2003; Mihalcea and Tarau, 2004), abstractive summarization involves rewriting and paraphrasing to generate summaries with novel words or sentences (See et al., 2017; Rush et al., 2015; Nallapati et al., 2016a).

Under supervised training settings, abstractive summarization methods require paired data (i.e., document-reference summary pairs) (See et al., 2017). However, obtaining high-quality and large-scale datasets for supervised training is laborious and expensive. Besides, human-annotations (i.e., human-written summaries) are not always the finest, since summarization task is not simple even for humans, which would hinder the construction of high-quality parallel training data.

Therefore, unsupervised abstractive summarization methods are attracting increasing attentions, such as SEQˆ3 (Baziotis et al., 2019), TED (Yang et al., 2020), and Adversarial REINFORCE (Wang and Lee, 2018). These models could be viewed as generative self-supervised methods (Xiao et al., 2021) with a common key component: a *reconstructor* to reconstruct the source inputs. The design of the reconstructor relies on the constraint that a good model-generated summary should be able to perfectly reconstruct the source input document. However, even for humans, it would be very

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4223 - 4234

difficult to reconstruct the source document given the summary. We empirically show the difficulty by asking human to write the source documents given the reference summaries. Table 1 shows an example, including the given reference summary, the source document and the human-written source document (more details are in Section 4.5).

We thus view the reconstruction objective as an over-restriction and propose to apply contrastive self-supervised method (Xiao et al., 2021) to relax this restriction. Our work is inspired and motivated by: it is more feasible for humans to select the true source document (from a pool of documents) given the reference summary, than reconstruction. Given a *good* summary, it is expected to easily select its source document from a document candidate pool, while the candidate pool includes the true source document and fake source documents. The main reason is that a good summary is supposed to be faithful and informative, with all important information to select the true source document. Reversely, we could consider a summary as good if it is easy to select the true source document from the document candidate pool given this summary. In order to prove this intuition, we ask humans to select the true source document given its summary. The results show that the selection accuracy is much higher than given a bad summary (more details are in Section 4.5).

In this work, we propose to leverage the learning signal of "selecting the true source documents" to guide the summary generation under unsupervised settings. Specifically, the model is trained to generate summaries, and then to maximize the semantic similarities between the generated summaries and **positive source document examples**, while minimizing that of generated summaries and **negative source document examples**. We design different types of strategies to obtain negative source document examples. Compared to positive source document examples, these negative source document examples might miss some important contents or have incorrect or irrelevant information. Meanwhile, we also consider the writing quality (e.g., syntactically and grammatically correct, clear writing, readable for humans) by leading the generated summaries similar to human-written texts.

The contributions of this paper are summarized as follows: **(1)** We propose a novel model **SCR** (**S**ummarize, **C**ontrast and **R**eview) for **unsupervised abstractive summarization**. To the best of our knowledge, this is the first work on applying contrastive learning for unsupervised abstractive summarization. **(2)** We design different strategies to generate negative source document examples for contrastive learning. To the best of our knowledge, this is the first work on generating negative examples on source documents (instead of summaries) for unsupervised abstractive summarization. **(3)** The experiment results show that SCR outperforms other unsupervised abstractive baselines, which demonstrates its effectiveness.

2 Related Work

2.1 Unsupervised Abstractive Summarization

Unsupervised abstractive summarization models are trained without document-reference summary pairs as the training data. West et al. (2019) applied the Information Bottleneck principle for unsupervised sentence summarization by generating the summary that could best predict the next sentence of the source sentence. While Févry and Phang (2018) applied denoising auto-encoder for unsupervised extractive sentence compression, (Baziotis et al., 2019) presented a sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression, where the model was trained to compress the input sentence to a summary and then reconstruct the input sentence, additionally with language model prior loss and topic loss. Similarly, Wang and Lee (2018) trained a generator and a reconstructor for unpaired abstractive summarization, and a discriminator to make the generated summary human-readable. A pretrained unsupervised summarization model was proposed by (Yang et al., 2020), which leveraged the lead bias in news articles for pretraining on large-scale data. and used theme modeling and denoising for finetuning. The goal of the reconstruction (Baziotis et al., 2019; Yang et al., 2020; Wang and Lee, 2018) is to ensure that the generated summaries could keep the core and important information. However, we consider it as an over-restriction and instead applied contrastive learning to generate summaries that could match the source documents semantically.

2.2 Contrastive Learning in Summarization

Leveraging negative examples for training has been investigated in summarization. The SimCLS framework was proposed in Liu and Liu (2021), which includes candidate generation and evaluation via contrastive learning. Xu et al. (2021) viewed the

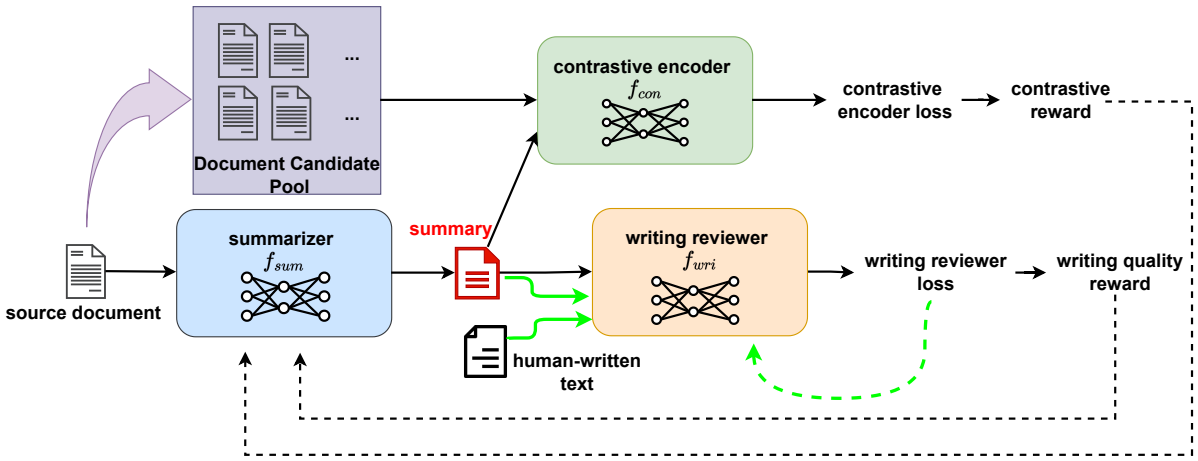

Figure 1: The overall framework of our proposed model SCR, including summarizer, contrastive encoder and writing reviewer. The solid arrows represent inputs and outputs, while the dashed arrows mean learning signals for optimization. The green arrows represent the writing reviewer training. At testing phase, we only use the summarizer to generate the **summary**.

source document, reference summary and the generated summary from the model as the same representation and tried to minimise their distances in similarity space during training. Wu et al. (2020) developed a novel method for summary quality evaluation (without reference summary) that could cover both linguistic and semantic aspects. Cao and Wang (2021) presented a novel method to improve the faithfulness and factuality in abstractive summarization via contrastive learning, while constructing negative examples with different methods for training. Instead of constructing negative samples based on the summaries as in (Cao and Wang, 2021), Zheng et al. (2021) generated augmented examples based on the source documents as document augmentation, and proposed a framework to perform both contrastive learning and the summary generation in a supervised learning setting. Zheng et al. (2021) is the only work that we found on generating augmented examples based on the source documents, where the authors viewed the augmented examples as positive examples and aimed at enhancing the model's denoising ability. In contrast, we view the modified version of source documents as negative examples and targeted on unsupervised abstractive summarization.

3 Methodology

The framework of the proposed SCR for unsupervised abstractive summarization is shown in Fig. 1, which includes a *summarizer* f_{sum}, a *contrastive encoder* f_{con} and a *writing reviewer* f_{wri}. The summarizer aims at generating summaries given the source documents, with the learning feedback from the contrastive encoder and writing reviewer: (1) The summarizer and contrastive encoder are trained via contrastive learning, which leads the generated summary to be faithful and informative. (2) The summarizer and writing reviewer are optimized in an adversarial manner (Goodfellow et al., 2014), where the summarizer is updated to generate summaries that could have high quality scores from the writing reviewer when the writing reviewer is fixed. It would encourage the generated summaries to be high-quality in writing.

Finally, both the contrastive encoder and writing reviewer guide the summarizer to generate good summaries under unsupervised learning settings.

3.1 Summarizer

Given a source document $d = \{x_1, x_2, ..., x_S\}$ with S tokens and x_i represents the i-th token, the goal of summarizer f_{sum} is to generate a summary $\hat{s} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_T\}$ with T tokens, where $\hat{y}_j$ is the j-th token in the generated summary, without the need for reference summary.

We adopt the sequence-to-sequence architecture to build the summarizer. As the Transformer-based architectures have been proven to be successful and effective in language generation tasks (Vaswani et al., 2017; Devlin et al., 2019; Lewis et al., 2020; Zhang et al., 2020), we implement the summarizer by adopting the encoder and decoder of Transformer, following the standard Transformer architecture design in Vaswani et al. (2017). We use 6 layers and 8 attention heads in encoder and decoder.

Different from the decoders under supervised learning settings that use tokens from the reference summary as input, we use the token from the previous time step as the input of the decoder for the current time step.

Finally the generated summary would be the input to the contrastive encoder f_{con} and writing reviewer f_{wri}, and the learning signals from the contrastive encoder and writing reviewer would lead the optimization of the summarizer. To generate each token and output the final summary, there is a sample process over the probability distribution $P_\theta(\hat{y}_i|\hat{y}_1, \hat{y}_2, ..., \hat{y}_{i-1}, d)$ for generation at each time step, where θ is the parameters of f_{sum}. As sampling is a non-differentiable process, it is not feasible to directly apply gradient descent based method to optimize f_{sum}. We use policy gradient (Sutton et al., 1999; Yu et al., 2017) to update the parameters θ of summarizer f_{sum} (details in Section 3.4).

3.2 Contrastive Encoder and Contrastive Learning

The contrastive encoder f_{con} takes the generated summary $f_{sum}(d)$ and document candidate pool $C = \{c_1, c_2, ..., c_K\}$ (with 1 positive example and K-1 negative examples) as inputs, while it is trained to maximize the semantic similarities between the generated summaries and positive source document examples in the semantic space, as well as to minimize that with the negative source document examples. This contrastive learning process aims to guide the generated summaries to be faithful and informative.

3.2.1 Document Candidate Pool

For each source document, we set itself as the positive source document example, and generate K-1 negative source document examples with the following strategies:

Insertion. We randomly select m existing sentences (sampled from the dataset) and insert them into a random position in the positive example. We generate 3 different negative examples for $m = 1, 2$ respectively, and thus 6 negative examples in total.

Deletion. We randomly select m sentences in the positive example and delete them. Also, we generate 3 different negative examples for $m = 1, 2$ respectively, and thus 6 negative examples in total.

Replacement. We randomly select m sentences in the positive example and replace them with other m sentences that are randomly sampled from the dataset. We get 3 different negative examples for $m = 1, 2$ respectively, and have 6 negative examples totally.

Entity Swap. We randomly select m named entities in the positive example and replace them with other randomly selected entities of the same entity type from the dataset. This yield 3 different negative examples for $m = 40\%, 80\%$ respectively to obtain 6 negative examples in total.

Combination. We randomly select multiple (≥ 2) methods from above to generate negative example in a random order (e.g., replacement $\rightarrow$ insertion $\rightarrow$ entity swap). We repeat this process for 6 times and generat 6 different negative examples.

3.2.2 Contrastive Encoder Training

We utilize the Transformer (Vaswani et al., 2017) to build the contrastive encoder. Particularly we employ the encoder part of the Transformer to learn representations of document examples. To have better representation learning, we use two different encoders to encode the generated summary (shorter text) and document examples (from C) (longer text), respectively, denoted as f_s^{enc} (6 layers and 8 attention heads) and f_d^{enc} (12 layers and 12 attention heads). We set the final hidden state for the token [*CLS*] (the first token of all sequences input) as the representation, which yields:

$$\mathbf{v}_{\hat{s}} = f_s^{enc}(f_{sum}(d)), \mathbf{v}_{c_i} = f_d^{enc}(c_i), c_i \in C \tag{1}$$

where $\mathbf{v}_{\hat{s}}$, $\mathbf{v}_{c_i}$ are the embeddings for the generated summary and document example respectively.

Denote the positive source document example in C as c^+ and negative source document example as c^- (so that each c_i in C is either c^+ or c^-) , their embeddings hence would be: $\mathbf{v}_{c^+} = f_d^{enc}(c^+)$ and $\mathbf{v}_{c^-} = f_d^{enc}(c^-)$.

For each generated summary $\hat{s}$, we have the contrastive encoder loss for training:

$$l_{cl}^{\hat{s}} = -\log \frac{\exp(\cos(\mathbf{v}_{\hat{s}}, \mathbf{v}_{c^+})/\tau)}{\sum_{c_i} \exp(\cos(\mathbf{v}_{\hat{s}}, \mathbf{v}_{c_i})/\tau)} \tag{2}$$

$$\sum_{c_i} \exp(\cos(\mathbf{v}_{\hat{s}}, \mathbf{v}_{c_i})/\tau) = \exp(\cos(\mathbf{v}_{\hat{s}}, \mathbf{v}_{c^+})/\tau) + \sum_{c^-} \exp(\cos(\mathbf{v}_{\hat{s}}, \mathbf{v}_{c^-})/\tau) \tag{3}$$

where $\exp(\cdot)$ is the exponential function, and $\cos(\cdot,\cdot)$ is the cosine similarity function. τ is the temperature and is set as 1.0. The contrastive encoder is optimized to minimize $l_{cl}^{\hat{s}}$.

3.3 Writing Reviewer

The writing reviewer aims to guide the generated summaries to be high-quality in writing (e.g., syntactically and grammatically correct, clear writing, readable for humans). To achieve an effective guidance, we apply the writing reviewer to estimate the writing quality, where it is trained to maximize the quality score of human-written texts (as *positive writing samples*, dentoed as s^*) as well as to minimize the quality score of the generated summaries (as *negative writing samples*, i.e., $\hat{s}$) by summarizer. We obtain the human-written text s^* with two methods: (1) by randomly sampling a source document from the dataset and setting the first H sentences of sample document as s^*, or (2) by randomly sampling a source document and randomly setting consecutive H sentences in the sampled document as s^*.

Instead of only returning a final score, we expect to obtain a score for each word in the input text, so that the writing reviewer could indicate the writing quality of the current generated word given the previous context, which could be used to estimate for a partially generated summary. Specifically, for an input text $t = \{t_1, t_2, ..., t_W\}$ with W tokens, the writing reviewer outputs a sequence of writing quality score $z = \{z_1, z_2, ..., z_W\}$, where z_i indicates the quality of the sequence $\{t_1, t_2, ..., t_i\}$. We apply a Long short-term memory (LSTM) network to predict the score at each time step. It was implemented as a one-layer LSTM and the dimension of hidden state is set as 512.

To train the writing reviewer f_{wri}, we feed the human-written texts s^* as positive writing samples and the generated summaries $\hat{s}$ as negative writing samples and set the loss function as:

$$l_{wri} = \frac{1}{N}\sum_{N}[f_{wri}(\hat{s}) - f_{wri}(s^*) + \lambda(||\nabla_{\bar{s}} f_{wri}(\bar{s})||_2 - 1)^2] \tag{4}$$

where N is the number of examples in each mini-batch, $(||\nabla_{\bar{s}} f_{wri}(\bar{s})||_2 - 1)^2$ is the gradient penalty (soft version of the Lipschitz constraint (Gulrajani et al., 2017)) applied on the interpolated output $\bar{s}$ between $\hat{s}$ and s^*, and $f_{wri}(\cdot)$ is the mean of score sequence z for the input text ($\hat{s}$, s^*, $\bar{s}$).

3.4 Model Optimization

To address the non-differentiable sampling problem in summarizer, we formulate the summary generation as a reinforcement learning approach (Yu et al., 2017; Rennie et al., 2017; Wang and Lee, 2018). The *agent*, also summarizer f_{sum}, takes *action* to generate the summary, so as to maximize the *reward* from the contrastive encoder f_{con} and writing reviewer f_{wri}. Therefore, the summarizer is optimized with the learning signal (reward) from the contrastive encoder and writing reviewer. Hence, the goal of summarizer training is to minimize the negative expected reward with policy gradient:

$$l_{sum}(\theta) = -\mathbb{E}_{\hat{s}\sim P_\theta}[r(d, \hat{s})] \tag{5}$$

$$\nabla_\theta l_{sum}(\theta) = -\mathbb{E}_{\hat{s}\sim P_\theta}[r(d, \hat{s})\nabla_\theta log P_\theta(\hat{s}|d)] \tag{6}$$

where $r(\cdot,\cdot)$ denotes the reward that is defined in the Section 3.4.1 and 3.4.2, θ is the parameter of f_{sum}, and Eq. (6) is the derivative of Eq. (5) with respect to θ, which avoids the problem of non-differentiable sampling process.

3.4.1 Reward From Contrastive Encoder

The negative loss function for the contrastive encoder $-l_{cl}^{\hat{s}}$ (Eq. (2)) could be set as reward for the summarizer. In order to have stable reward, we apply self-critical sequence training method by having a reward baseline (Rennie et al., 2017; Wang and Lee, 2018). The baseline was set as $-\alpha l_{cl}^{s_{greedy}}$, where s_{greedy} is the summary that is greedy decoded at each time step while $\hat{s}$ is sampled over the probability distribution $P_\theta(\hat{y}_i|\hat{y}_1, \hat{y}_2, ..., \hat{y}_{i-1}, d)$. α is the hyperparameter that is set as 0.75 and gradually increased to 1.0. Thus, we set the reward from the contrastive encoder as:

$$r_{con} = -l_{cl}^{\hat{s}} + \alpha l_{cl}^{s_{greedy}} \tag{7}$$

3.4.2 Reward From Writing Reviewer

As described in Section 3.3, the writing reviewer outputs a quality score for each time step. Therefore, the quality of generating one token t_i could be set as $z_i - z_{i-1}$ (Wang and Lee, 2018), which estimates the writing quality on the i-th time step compared to the $i-1$-th time step. Thus, the reward from the writing reviewer is:

$$r_{wri} = \begin{cases} z_i - z_{i-1}, & \text{if } i \geq 2 \\ z_1, & \text{if } i = 1 \end{cases} \tag{8}$$

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We evaluate our proposed model on two commonly used benchmark datasets for abstractive summarization.

CNN/DailyMail (Hermann et al., 2015; Nallapati et al., 2016b) consists of online news articles and their corresponding multi-sentence abstracts. The average number of words in source documents and summaries is 781 and 56, respectively. We use the non-anonymized version of the dataset which splits training, validation and testing into 287,226/13,368/11,490 data pairs, and follow the pre-processing as in See et al. (2017).

Gigaword Rush et al. (2015) is another news summarization benchmark dataset (a total of 3.8M/189k/1,951 pairs for train/validation/test splits), where the input for the summarization models is the first sentence (averagely 29 words) of the original news, and the output is the headline (averagely 8.8 words) of the news article.

To perform unsupervised abstractive summarization, we only use the source documents from the dataset for training.

4.1.2 Baselines

We mainly compare the proposed model SCR with the unsupervised models. In order to position our model among all the abstractive text summarization models, we further provide the results of supervised learning based models and zero-short learning based models for reference.

Unsupervised learning models. The baselines we compare with are mainly unsupervised abstractive summarization models: SEQˆ3 (Baziotis et al., 2019), TED (Yang et al., 2020), Adversarial REINFORCE (Wang and Lee, 2018), Contextual Match (Zhou and Rush, 2019), HC_article_10 (Schumann et al., 2020), NAUS (Liu et al., 2022).

Supervised learning models. Due to the success of BERT-based (Devlin et al., 2019) models and large-scale pre-training models in a wide range of tasks, we also take these models into account: PEGASUS (Zhang et al., 2020), ProphetNet (Qi et al., 2020), MUPPET (Aghajanyan et al., 2021).

Zero-shot learning models. We consider the zero-shot settings for summarization of large-scale pre-training models, including BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020), BART-LB and T5-LB (Zhu et al., 2021).

4.1.3 Pretraining

We pre-train the summarizer, contrastive encoder and writing reviewer respectively. Specifically, we exploit the lead-bias (Zhu et al., 2021) for summarizer pre-training. We set the first L sentences as the summary to predict. The input for the summarizer is the rest of the document. The pre-training would be helpful for the summarizer to have a basic ability on understanding, as well as inferring. Considering the average sentence count, we set L as 3 for CNN/DailyMail (Hermann et al., 2015). For Gigaword (Rush et al., 2015), the input is a one-sentence source document. We then set the first 8 words as the target summary and the rest of the sentence as the input document for pretraining. We also pre-train the contrastive encoder with the pre-trained summarizer using Eq. (2). For the writing reviewer, we use the generated summaries from the pre-trained summarizer as negative writing samples and human-written texts as positive writing samples for pretraining using Eq. (4).

4.2 Results

We evaluate the quality of the generated summaries automatically with ROUGE F1 score (Lin, 2004), which covers ROUGE-1 score on uni-gram overlap, ROUGE-2 score on bi-gram overlap, and ROUGE-L on the longest common subsequence.

4.2.1 Model Performance

The results on CNN/DailyMail and Gigaword dataset are shown in Table 2. On CNN/DailyMail dataset, our proposed model SCR surpasses all the unsupervised baselines across all ROUGE metrics, including the previous state-of-the-art unsupervised model TED (Yang et al., 2020). Our model SCR outperforms TED by 0.33, 0.59, 1.72 in R-1, R-2, R-L respectively. On Gigaword dataset, SCR has competitive results as other unsupervised baselines. Compared to the previous state-of-the-art unsupervised model NAUS (Liu et al., 2022), SCR is better on R-2 while R-1 and R-L is slightly lower. Overall, the competitive ROUGE scores significantly show that our model is able to generate summaries with higher quality for both document (CNN/DailyMail) and sentence (Gigaword).

Performance Discussion. While our model surpasses other unsupervised abstractive summarization methods, there are some performance gaps between our model and the supervised learning

Model	CNN/DailyMail			Gigaword		
	R-1	R-2	R-L	R-1	R-2	R-L
Supervised						
ProphetNet (Qi et al., 2020)	44.20	21.17	41.30	39.51	20.42	36.69
MUPPET (Aghajanyan et al., 2021)	44.45	21.25	41.4	40.40	20.54	36.21
PEGASUS (Zhang et al., 2020)	44.17	21.47	41.11	39.12	19.86	36.24
Zero-shot						
PEGASUS (Zhang et al., 2020)	32.90	13.28	29.38	23.39	7.59	20.20
BART (Lewis et al., 2020) (Zhu et al., 2021)	32.83	13.30	29.64	22.07	7.47	20.02
T5-LB (Zhu et al., 2021)	38.47	16.62	35.23	24.00	8.19	21.62
BART-LB (Zhu et al., 2021)	40.52	17.63	36.76	25.14	8.72	22.35
Unsupervised						
Adversarial REINFORCE (Wang and Lee, 2018)	35.51	9.38	20.98	28.11	9.97	25.41
Contextual Match (Zhou and Rush, 2019)	14.25	3.10	10.87	26.48	10.05	24.41
HC_article_10 (Schumann et al., 2020)	/	/	/	24.44	8.01	22.21
TED (Yang et al., 2020)	38.73	16.84	35.40	25.58	8.94	22.83
SEQˆ3 (Baziotis et al., 2019) (Zhu et al., 2021)	23.24	7.10	22.15	25.39	8.21	22.68
NAUS (Liu et al., 2022)	/	/	/	**28.55**	9.97	**25.78**
SCR (ours)	**39.06**	**17.43**	**37.12**	28.10	**11.63**	24.14

Table 2: The ROUGE-1, ROUGE-2 and ROUGE-L results on the datasets, including the supervised models, zero-shot models and unsupervised models. The bold scores represent the best performance of unsupervised models.

based models PEGASUS (Zhang et al., 2020), ProphetNet (Qi et al., 2020), MUPPET (Aghajanyan et al., 2021). We believe this is reasonable because the training schemes of these models are usually pretrained on massive text corpora with different objectives and then fine-tuned for downstream tasks under supervised learning setting (MUPPET (Aghajanyan et al., 2021) additionally has the pre-finetuning, after pre-training and before fine-tuning). However, our model SCR can not be optimized directly by the supervision signals from the reference summaries, leading to the performance gaps. Even though, the scores of our model are decreased approximately only by 10% (R-1, R-L) and 19% (R-2) on CNN/DailyMail with comparison to those supervised models, which is impressive. The competitive results indicate that SCR could be applied effectively on zero-resource summarization without any reference summary. In zero-shot settings, models are only pre-trained but without fine-tuning on target datasets, which is different from unsupervised training. Thus it is also not comparable and their results are only for reference. Furthermore, We conduct experiments to test the model's ability on learning transferable features among different datasets. The details and results are in Appendix A.

4.2.2 Ablation Study

To verify the effectiveness of the contrastive encoder and writing reviewer in our proposed model, we conduct ablation study on CNN/DailyMail dataset and the results are shown in Table 3. From the results, all the ROUGE scores have decreased, which demonstrates the importance of the contrastive encoder and writing reviewer. Moreover, training without the contrastive encoder results in a much larger decline, than without the writing reviewer. We discuss the possible reasons as follows.

Without Writing Reviewer. The contrastive encoder aims at making the generated summary close to the source documents semantically, hence the generated summary would possibly keep some important content (e.g. entity). Such summary, although might present some writing issues (e.g., syntactically or grammatically incorrect), could still have a higher ROUGE scores, for the ROUGE scores are calculated mainly based on content.

Without Contrastive Encoder. The goal of the writing reviewer is to ensure that the generated summaries could be high-quaility as human-written texts. Training only with the writing reviewer would not generate summaries that contain the most important information of the source document. Despite being more high-quality in writing, the ROUGE scores would still be lower than training with the contrastive encoder.

Model	R1	R2	RL
w/o contrastive encoder	25.08	9.20	22.01
w/o writing reviewer	30.61	11.03	25.73
full model	39.06	17.43	37.12

Table 3: Ablation study results on CNN/DailyMail dataset.

Settings	R1	R2	RL
Unpaired Training	39.98	18.20	37.66
Unsupervised Training	39.06	17.43	37.12

Table 4: Results of unpaired and unsupervised training on CNN/DailyMail dataset.

Summary	Info	Read	Lede
Reference	8.2	8.6	7.9
Model-generated	8.4	8.2	8.4

Table 5: Results of human evaluation on different aspects: Info(informativeness), Read(Readability) and Lede(Lede-copying behaviour).

4.2.3 Unpaired VS Unsupervised

We notice that our unsupervised training settings are slightly different from the unpaired training settings in Wang and Lee (2018). We view unpaired training as a "lenient version" of unsupervised training as follows.

Unpaired Training. The model has access to the reference summaries in the dataset, but the source documents and reference summaries are unpaired. Under unpaired training, the reference summaries serve as *positive writing samples* for writing reviewer training.

Unsupervised Training. The model has completely no access to the reference summaries in the dataset. The *positive writing samples* are sampled from the source documents in the dataset.

We train our model SCR under unpaired settings to study the differences from unsupervised settings. The results of our model under this two settings are listed in Table 4, showing unpaired training is slightly better than the unsupervised training across R-1, R-2 and R-L. We believe it is reasonable because the model is exposed to the reference summaries for training, while the ROUGE evaluation compares the generated summaries and reference summaries. Moreover, our model SCR outperforms Adversarial REINFORCE (Wang and Lee, 2018) under unpaired settings on CNN/DailyMail dataset across all ROUGE metrics, which is able to demonstrate that SCR is effective under both settings.

4.3 Huamn Evaluation

We conduct the human evaluation to evaluate the quality of the summaries generated by our proposed model. We first sample 30 examples randomly from the test set of CNN/DailyMail dataset and then two volunteers are asked to evaluate and score the quality of the generated summaries and reference summaries. The volunteers don't have information of the given summaries are either generated from the model or the reference summaries. To have a detailed evaluation, the summaries are evaluated based on the following aspects:

Informativeness: is the summary providing all the important information from the source document?

Readability: the writing quality of the summary, including syntactically and grammatically correct, clear writing, fluency, readable for humans.

Lede-copying Behaviour: is the summary simply copying the leading sentences from the source document?

For each aspect, the scores range from 1 to 10 (1 indicates the worst, 10 indicates the best). The results are shown in Table 5, which show that the summaries generated by our proposed model are as good as (even slightly better than) the reference summaries that are written by humans.

4.4 Example Summary

An example summary generated by the proposed model is shown in Table 6, as well as the source document and reference summary. As we can observe from the example summary, the SCR model could capture the salient information from the source document, such as the name of the HBO crime drama *"True Detective"* that is missing in the reference summary. Besides, we also notice that the model could generate novel words or phrases. For example, given *"which premieres June 21"* in the source document, the model could rewrite as *"coming on June 21"* in the output summary.

4.5 Motivation Experiments

Our work is motivated by the observations and findings from the experiments that compares reconstruction and contrastive learning method for *human* summarization. The purpose for conducting these experiments is to demonstrate that it is much more difficult for humans to reconstruct the source documents than to select the true source document given the summary.

Specifically, we randomly select 10 reference summaries from CNN/DailyMail. Two volunteers are asked to write their source document. We also use ROUGE scores (Lin, 2004) to measure the quality of human-written source documents,

Source document	... HBO just whetted our appetite for a new season of "True Detective." The network released a teaser video for season 2 of the critically acclaimed show ... which premieres June 21. Here's the plot synopsis ... The first season starred Matthew McConaughey and Woody Harrelson as a pair of Louisiana State Police detectives investigating the death of a young woman. The crime drama proved to be a runaway hit, and the season 1 finale crashed the HBO Go site in March 2014.
Reference summary	HBO released a teaser video for the new season, starting June 21. The series stars Colin Farrell and Vince Vaughn.
Model-generated summary	HBO released a teaser video for season 2 of "True Detective". Colin Farrell, Vince Vaughn, Rachel McAdams and Taylor Kitsch star in the new season, coming on June 21...

Table 6: An example of model-generated summary by SCR model, as well as the source document and reference summary. Words in the same color (blue or red) are information captured from the source document to the reference or model-generated summary.

which is shown in Table 7 (reconstruction part). The ROUGE scores are pretty low (especially R-2), which means the human-written source documents and true source documents are very different. We hence conclude that reconstructing the source document is difficult even for humans.

Furthermore, we randomly pick 10 document-reference summary pairs from CNN/DailyMail dataset, and construct the document candidate pool (as in Section 3.2.1) for each picked source document. Besides, for each reference summary (*good* summary), we generate *bad* summary by deleting, inserting or replacing random sentences on the reference summary. Compared to the good summary, the bad summary would hence miss some key information, or have irrelevant (or incorrect) contents. Two volunteers are asked to select the true source document from the document candidate pool, given only the good summary or only the bad summary. The accuracy is shown in Table 7 (contrastive learning part), which indicates that if the provided summary is a good one, the selection accuracy (80%) would be much higher than given the bad summary (15%).

From these experiments, we observe that it is much more difficult for humans to reconstruct the source documents than to select the true source document given the reference summary. However, given a bad summary, it becomes much more difficult to have correct selection. These findings motivate us to propose an unsupervised abstractive summarization method that leverages contrastive learning. We believe the experiments could demonstrate our motivation.

Reconstruction	**R1**	**R2**	**RL**
human-written documents	23.68	5.51	20.39
Contrastive Learning	**Selection Accuracy**		
good summary	80%		
bad summary	15%		

Table 7: Results of the motivation experiments.

5 Conclusion

In this paper, we propose SCR (Summarize, Contrast and Review) for unsupervised abstractive summarization. The summarizer is trained to generate summaries and the contrastive encoder guides the generate summaries close to the source documents semantically via contrastive learning. A writing reviewer is applied to ensure the writing quality of the generated summaries. Moreover, we design different strategies to generate negative source document examples for contrastive learning. Results on extensive experiments show the effectiveness of SCR. In future work, we hope to study our model on transferable feature learning and semi-supervised learning with the advantages of paired data.

Limitations

Although our proposed model could learn some transferable features among different datasets (details in Appendix A), we think there is still some improvement space (not as good as we expected). We believe that not being able to learn transferable features well is a limitation of our model, which we leave as our future improvement. Besides, we hope to explore our model on more datasets, espcially on other domain and other language.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811. Association for Computational Linguistics.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, and Alexandros Potamianos. 2019. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference*

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 673–681. Association for Computational Linguistics.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359. Association for Computational Linguistics.

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. 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*, pages 4171–4186. Association for Computational Linguistics.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the 2003 HLT-NAACL Text Summarization Workshop*, pages 1–8.

Thibault Févry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 2018 Conference on Computational Natural Language Learning*, pages 413–422. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 2014 Conference on Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein gans. In *Proceedings of the 2017 International Conference on Neural Information Processing Systems*, page 5769–5779. Curran Associates Inc.

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 2015 International Conference on Neural Information Processing Systems*, page 1693–1701. MIT Press.

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 2020 Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Puyuan Liu, Chenyang Huang, and Lili Mou. 2022. Learning non-autoregressive models from search for unsupervised sentence summarization. In *Proceedings of the 2022 Annual Meeting of the Association for Computational Linguistics*, pages 7916–7929. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 2021 Annual Meeting of the Association for Computational Linguistics and the 2021 International Joint Conference on Natural Language Processing*, pages 1065–1072. Association for Computational Linguistics.

Congbo Ma, Wei Emma Zhang, Mingyu Guo, Hu Wang, and Quan Z. Sheng. 2022. Multi-Document Summarization via Deep Learning Techniques: A Survey. Association for Computing Machinery.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016a. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016b. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410. Association for Computational Linguistics.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings*

of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, pages 7008–7024. IEEE.

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. Association for Computational Linguistics.

Raphael Schumann, Lili Mou, Yao Lu, Olga Vechtomova, and Katja Markert. 2020. Discrete optimization for unsupervised sentence summarization with word-level extraction. In *Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics*, pages 5032–5042. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, page 1057–1063. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 2017 Conference on Advances in Neural Information Processing Systems*, volume 30, page 5998–6008. Curran Associates, Inc.

Yaushian Wang and Hung-Yi Lee. 2018. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187–4195. Association for Computational Linguistics.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 2019 International Joint Conference on Natural Language Processing*, pages 3752–3761. Association for Computational Linguistics.

Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621. Association for Computational Linguistics.

Liu Xiao, Zhang Fanjin, Hou Zhenyu, Mian Li, Wang Zhaoyu, Zhang Jing, and Tang Jie. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2021. Sequence level contrastive learning for text summarization. *arXiv preprint arXiv:2109.03481*.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. TED: A pretrained unsupervised summarization model with theme modeling and denoising. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1865–1874. Association for Computational Linguistics.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the 2017 AAAI Conference on Artificial Intelligence*, page 2852–2858. AAAI Press.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 2020 International Conference on Machine Learning*, pages 11328–11339. PMLR.

Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, Ling Fan, and Zhe Wang. 2021. Enhanced seq2seq autoencoder via contrastive learning for abstractive text summarization. In *2021 IEEE International Conference on Big Data*, pages 1764–1771. IEEE.

Jiawei Zhou and Alexander Rush. 2019. Simple unsupervised summarization by contextual matching. In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics*, pages 5101–5106. Association for Computational Linguistics.

Chenguang Zhu, Ziyi Yang, Robert Gmyr, Michael Zeng, and Xuedong Huang. 2021. Leveraging lead bias for zero-shot abstractive news summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1462–1471. Association for Computing Machinery.

A Transferable Feature Learning

We have experiments to test the model's ability on learning transferable features among different datasets. We first train the model on the CNN/DailyMail dataset (source domain) under unsupervised learning settings, and then evaluated on the Gigaword (target domain) dataset without any fine-tuning, and vice versa.

The results are listed in Table 8, which shows a decrease across all the ROUGE scores with comparison to training and testing on same domain (results of SCR in Table 2) on both datasets. We believe the main reason is the text differences between two datasets (details in Section 4.1.1). The inputs are paragraph documents in CNN/DailyMail, which are much longer than the

Target Domain	R1	R2	RL
CNN/DailyMail	24.65	8.77	22.29
Gigaword	23.10	7.08	19.24

Table 8: Results of transferable feature learning. The model is trained on the source domain and evaluated on the target domain.

one-sentence as inputs in Gigaword. Each reference summary in CNN/DailyMail has averagely 56 words while only 8.8 in Gigaword. Moreover, training on CNN/DailyMail and evaluating on Gigaword would have a smaller decrease than the opposite training-testing datasets. It suggests that the model learns more transferable features from CNN/DailyMail than Gigaword. Document-level summarization training could help more for the model to perform sentence-level summarization. We leave the model's ability on learning transferable features as our future work.