Learn to Copy from the Copying History: Correlational Copy Network for Abstractive Summarization

Haoran Li[1], **Song Xu**[1], **Peng Yuan**[1], **Yujia Wang**[2], **Youzheng Wu**[1], **Xiaodong He**[1], **Bowen Zhou**[1]

[1] JD AI Research
[2] University of California, Berkeley
{lihaoran24, xusong28, yuanpeng29}@jd.com

Abstract

The copying mechanism has had considerable success in abstractive summarization, facilitating models to directly copy words from the input text to the output summary. Existing works mostly employ encoder-decoder attention, which applies copying at each time step independently of the former ones. However, this may sometimes lead to incomplete copying. In this paper, we propose a novel copying scheme named Correlational Copying Network (CoCoNet) that enhances the standard copying mechanism by keeping track of the copying history. It thereby takes advantage of prior copying distributions and, at each time step, explicitly encourages the model to copy the input word that is relevant to the previously copied one. In addition, we strengthen CoCoNet through pre-training with suitable corpora that simulate the copying behaviors. Experimental results show that CoCoNet can copy more accurately and achieves new state-of-the-art performances on summarization benchmarks, including CNN/DailyMail for news summarization and SAMSum for dialogue summarization. Our code is available at https://github.com/hrlinlp/coconet.

1 Introduction

Text summarization techniques (Rush et al., 2015; Chopra et al., 2016; Zhou et al., 2017; Li et al., 2018; Zhang et al., 2018; Li et al., 2019, 2020a,b; Xu et al., 2020a; Yuan et al., 2020) aim to generate a condensed and cohesive version of the input text, enabling readers to grasp the main points without reading the full text. There are two types of summarizers: extractive and abstractive. Extractive methods produce a summary by taking important sentences from the original text and combining these extracts, while abstractive methods involve interpreting and paraphrasing the input when generating a summary. The latter is more similar to how humans would summarize a text, but it is far more challenging to achieve.

Dialogue
Ernest: hey Mike , did you park your car on our street? *Mike*: no, took it into garage today *Ernest*: ok good *Mike*: why? *Ernest*: someone just crashed into a red Honda looking just like yours *Mike*: lol lucky me
Summary
Mike took his car into garage today. Ernest is relieved as someone had just crashed into a red Honda which looks like Mike's.

Table 1: An example from the dialogue summarization task. Highlighted words are copied consecutively from the input. Previously copied words (such as "*just crashed*") can guide the following copying operations (such as"*into a red Honda*").

Currently, the sequence-to-sequence (Seq2Seq) framework has become the mainstream for performing abstractive summarization tasks. However, it suffers from handling out-of-vocabulary words (OOV). As it has been observed that some words in the input text reappear in the summary, one way of coping with the OOV issue is by extracting words from the input text and incorporating them into the abstractive summary. Following this strategy, existing works (Gulcehre et al., 2016; Gu et al., 2016; See et al., 2017) propose the copying mechanism, which copies words from the input sequence to form part of the summary. These models generally regard the encoder-decoder attention as the copying distribution, which we call "attentional copying". They perform copying at each time step independently of the former ones, neglecting the guidance of the copying history. Our work demonstrates that the copying history can provide crucial clues of the copying behaviors for the following time steps and thereby encourage the summarizer to copy more accurately. For example, in Table 1, assuming the source words "*a red*" have been copied, the next

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4091–4101
November 7–11, 2021. ©2021 Association for Computational Linguistics

copying operation for the following word "*Honda*" can be explicitly induced.

In this paper, we propose a novel copying architecture named **Co**rrelational **Co**pying **Net**work (CoCoNet) that can learn to copy from the copying history. We build CoCoNet based on the Transformer-based Seq2Seq architecture (Vaswani et al., 2017) , which has shown superiority in various text generation tasks, such as machine translation and text summarization. More specifically, CoCoNet copies from the input text at each time step by selecting what is relevant to the previously copied word. It keeps track of the prior copying distribution and explicitly models the correlation between different source words by integrating semantic and positional correlations. We obtain the semantic correlations based on the encoder self-attention matrix as Xu et al. (2020b). Inspired by Yang et al. (2018), we represent positional correlations as a Gaussian bias, which considers the relative distances between source words and the scope of the local context when copying. The framework of our model is shown in Figure 1.

Furthermore, we enhance CoCoNet through pre-training with a self-supervised objective of text span generation with copying on the raw text corpora. Motivated by the work of Zhang et al. (2020), which has proven that pre-training resembling the downstream task leads to better and faster fine-tuning performances, we make sure our pre-training simulates the copying behaviors desired for the downstream summarization tasks. We divide each sequence in the corpora into two spans with some overlapping words, and the first span is used to generate the second in pre-training. We measure the overlap between the two spans based on ROUGE scores (Lin, 2004) to ensure that there are enough words to be generated by copying.

Our main contributions are as follows:

- We propose a Correlational Copying Network (CoCoNet) for abstractive summarization. It tracks the copying history and copies the next word from the input based on its relevance with the previously copied one.

- We further enhance CoCoNet's learning of copying through self-supervised pre-training on text span generation with copying.

- CoCoNet achieves new state-of-the-art performances on news summarization and dialogue summarization tasks, and experimental results show that CoCoNet can copy more accurately.

2 Related work

2.1 Copying Mechanism

The copying mechanism is widely used in abstractive summarization. It allows models to directly copy words from the input to the output. Vinyals et al. (2015) present the pointer network that uses attention distribution to select tokens in the input sequence as the output. Luong et al. (2015) propose to copy source words to the target sentence by a fixed-size softmax layer over a relative copying range. Gulcehre et al. (2016) leverage the attention mechanism to predict the location of the word to copy and apply a copying gate to determine whether to copy or not. Gu et al. (2016) propose to predict output words by combining copying and generating modes through a shared softmax function. See et al. (2017) introduce a copying probability to incorporate copying and generating distributions dynamically. Bi et al. (2020) adopt the copy mechanism in the language model pre-training. Existing works do not attempt to calculate the copying distributions based on the copying history, which is our focus.

2.2 Temporal Attention Mechanism

Our proposed copying mechanism is partially inspired by the temporal attention mechanism (Sankaran et al., 2016) that keeps track of previous attention scores and adjusts the future attention distribution by normalization with historical attention scores. This model has been proven effective in the text summarization task (Nallapati et al., 2016). Similar ideas are also adopted by the coverage mechanism for image caption (Xu et al., 2015), machine translation (Tu et al., 2016), and text summarization (See et al., 2017), maintaining a coverage vector to record the attention history to compute future attention distributions. Temporal attention mechanism is designed to avoid repetitive or insufficient attentions. While our work aims to learn a better copying mechanism from the copying history.

3 Model

3.1 Overview

The input of the text summarization task is a longer text, $\mathbf{x} = (x_1, x_2, ..., x_S)$ of S tokens, and the output is a condensed summary, $\mathbf{y} = (y_1, y_2, ..., y_T)$

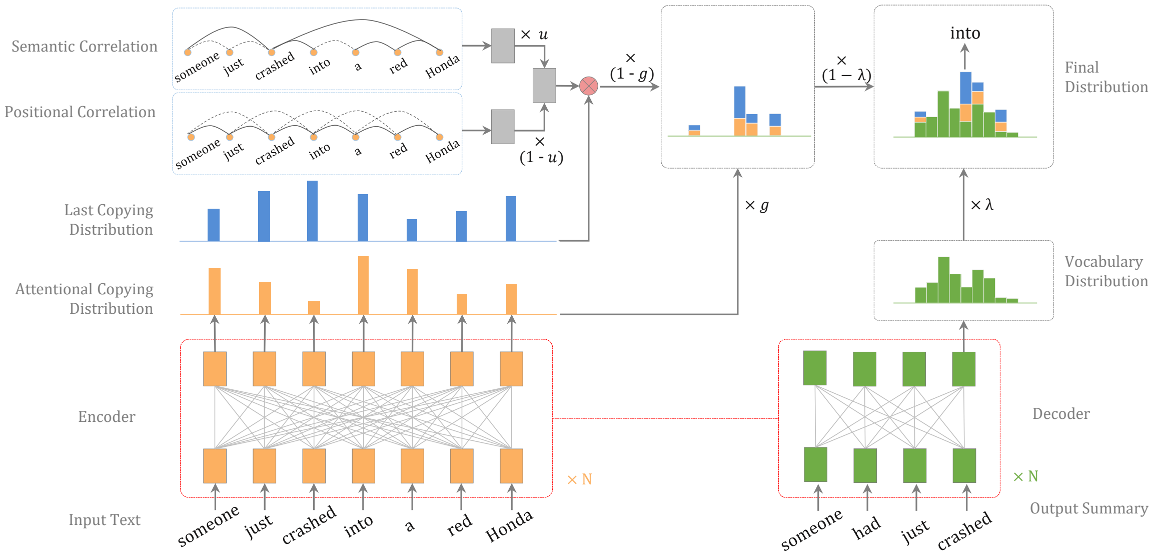


Figure 1: The framework of our model that keeps track of the copying history and copies what is relevant to the previously copied word. The solid lines in *Semantic* and *Positional Correlation* denote stronger correlations than the dashed lines.

of T tokens. The hypothesis of our proposed CoCoNet is that the standard attentional copying mechanism can be enhanced by the copying history. For example, a source word that is relevant to the previously copied one is more likely to be copied at the current time step. We further pre-train CoCoNet with the objective of text span generation with copying, which aims to strengthen the learning of the copying mechanism.

3.2 Transformer-based Seq2Seq Model

We adopt Transformer-based Seq2Seq architecture (Vaswani et al., 2017). The encoder of Transformer is a stack of N identical blocks, and each of them consists of two sublayers: a self-attention layer and a feed-forward layer. The encoder reads and converts the input sequence into the encoder's hidden states, $\mathbf{h}^{\mathbf{enc}}$, as follows:

$$\mathbf{h}^{enc} = f_{enc}(\mathbf{x}) \tag{1}$$

The decoder has similar structures as the encoder, stacking M identical blocks consisting of a self-attention attention layer, an encoder-decoder attention layer, and a feed-forward layer. The decoder's hidden states, $\mathbf{h}^{\mathbf{enc}}$, are generated given the encoder's hidden states and the previously generated words, and then we get the generation distribution based on $\mathbf{h}^{\mathbf{enc}}$:

$$h_t^{dec} = f_{dec}(\mathbf{h}^{enc}, y_{t-1}) \tag{2}$$

$$P_t^{gen}(w) = softmax(W_D h_t^{dec}) \tag{3}$$

The maximum likelihood (ML) training objective aims to minimize the negative log-likelihood of the parameters as follows:

$$\mathcal{L}_{ML} = -\sum_{t=1}^{T} log(P_t^{gen}(y_t)) \tag{4}$$

3.3 Attentional Copying Mechanism

The copying mechanism facilitates the model in predicting output words by integrating copying and generating distributions as follows:

$$P_t(y_t) = \lambda_t \cdot P_t^{gen}(w) + (1 - \lambda_t) \cdot P_t^{attCopy}(w) \tag{5}$$

where λ_t denotes the copying probability, and $P_t^{attCopy}(w)$ denotes the exiting copying distribution that is generally represented as the decoder-encoder attention by existing works as follows:

$$Q_t, K_i, V_i = h_t^{dec} W_Q, h_i^{enc} W_K, h_i^{enc} W_V \tag{6}$$

$$e_{t,i} = \frac{Q_t K_i^T}{\sqrt{d_k}} \tag{7}$$

$$\alpha_t = softmax(e_t) \tag{8}$$

$$P_t^{attCopy}(w) = \sum\nolimits_{i:x_i=w} \alpha_{t,i} \tag{9}$$

$$\lambda_t = sigmoid(W_\lambda \sum\nolimits_i (\alpha_{t,i} \cdot V_i)) \tag{10}$$

where d_k denotes the number of columns of the query matrix Q_t. Note that for the multi-head attention, we can obtain the copy distributions with the average of multiple heads.

3.4 Correlational Copying Mechanism

We propose a correlational copying mechanism that takes advantage of prior copying distributions and, at each time step, explicitly encourages the model to copy the input word that is relevant to the previously copied one. Our hypothesis comes from the observation that a cohesive summary typically has a reasonable language modeling for copying, especially for some important contents. For example, a source word that is relevant to the previously copied one is more likely to be copied at the current time step. As illustrated in Table 1, previously copied words "*just crashed*" are indicative for the following copied words "*into a red Honda*". Therefore, we propose to explicitly learn the language modeling for copying. We maintain a correlational copying distribution transferred from the last copying distribution based on the correlation between different source words:

$$P_t^{coCopy}(w) = \sum\nolimits_{i:x_i=w} \sum\nolimits_{j:x_j \in \mathbf{x}} P_{t-1}^{finalCopy}(x_j) \cdot rel_t(x_j, x_i) \tag{11}$$

$$rel_t(x_j, x_i) = u_t \cdot s_{j,i} + (1 - u_t) \cdot p_{j,i} \tag{12}$$

$$u_t = sigmoid(W_u Q_t) \tag{13}$$

where $P_t^{coCopy}(w)$ denotes the correlational copying distribution, and $P_t^{finalCopy}$ is the final copying distribution to predict output words, served as $P_t^{attCopy}$ in Equation 5. $rel_t(x_j, x_i)$ denotes the correlation score between source word x_j and x_i, integrating semantic correlation $s_{j,i}$ and positional correlation $p_{j,i}$, which we will introduce later. The above process can be regarded as one step of transition in the Markov chain, where the correlation matrix is analogous to the transition matrix. Note that there is no self-transferring for the correlational copy distribution, and thus, the word already obtaining a high copy score will not be copied repetitively.

Then, the correlational copying distribution is used to adjust the current copying distribution, which informs the model of the previously copied one when determining which word to copy now.

$$P_t^{finalCopy}(w) = g_t \cdot P_t^{attCopy}(w) + (1 - g_t) \cdot P_t^{coCopy}(w) \tag{14}$$

$$g_t = sigmoid(W_g \sum\nolimits_i (\alpha_{t,i} + P_t^{coCopy}(x_i)) \cdot V_i)) \tag{15}$$

P_t^{coCopy} is initialized as a zero vector. In the next time step, $P_t^{finalCopy}$ in Equation 14 serves as $P_{t-1}^{finalCopy}$ in Equation 11. In this way, the copying history is maintained recurrently.

3.4.1 Semantic Correlation

Xu et al. (2020b) propose to obtain the centrality score for each source word based on the last encoder self-attention layer. Following this work, we represent the semantic correlation between source words by the encoder self-attention weight:

$$Q_j^E, K_i^E = h_j^{enc} W_Q^E, h_i^{enc} W_K^E \tag{16}$$

$$e_{j,i} = \frac{Q_j^E (K_i^E)^T}{\sqrt{d_k}} \tag{17}$$

$$\alpha_j = softmax(e_j) \tag{18}$$

$$s_{j,i} = \alpha_{j,i} \tag{19}$$

3.4.2 Positional Correlation

Inspired by Yang et al. (2018), we represent the positional correlation as a Gaussian bias, which considers the relative distances between different source words and range of local context suitable for copying:

$$p_{j,i} = \frac{1}{\sqrt{2\pi}\delta_j} e^{\frac{-(pst_j - pst_i)^2}{2\delta_j^2}} \tag{20}$$

$$\delta_j = \frac{|\mathbf{x}|}{2} sigmoid(W_\delta Q_j) \tag{21}$$

where pst_j and pst_i denote the positions for source word x_j and x_i, respectively. δ_j denotes the standard deviation that conditions on the length of the source sequence, i.e., $|\mathbf{x}|$.

Different from Yang et al. (2018), we do not apply the predicted central position, because we argue that the information of relative position is strongly associated with the word correlations. In addition, following Shaw et al. (2018), we perform a relative distance clipping to improve the generalization of our model.

3.5 Correlational Copying Pre-training (CoCoPretrain)

Pre-training with self-supervised objectives on raw text corpora has demonstrated the effectiveness of a broad range of text generation tasks (Song et al., 2019; Dong et al., 2019; Lewis et al., 2020; Zhang et al., 2020). In this paper, we enhance CoCoNet through correlational copying pre-training (CoCoPretrain) on text span generation. The process of

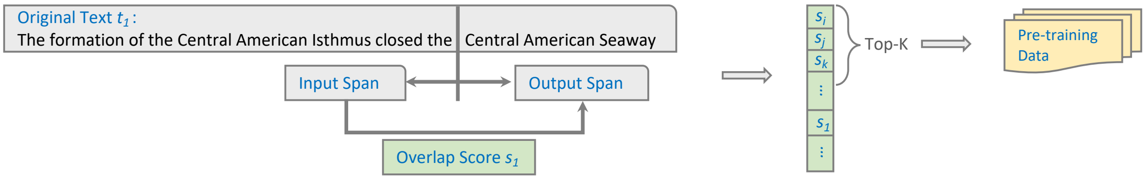


Figure 2: The process of constructing the pre-training data. Given a piece of text, we divide it into an input span and an output span, and we calculate the overlap score of them by Equation 22. The top-K scored span pairs are selected.

constructing the pre-training data suitable for correlational copying is as follows, and an example is shown in Figure 2.

We first divide each sequence in the raw corpora into two continuous spans, and the first longer span is used to generate the second in pre-training. We elaborately select the input text span followed by the output span by maximizing the overlap between the input and output. In this way, our CoCoPretrain objective can be also called overlapped text span generation.

As a measure for overlap, we adopt ROUGE F1 score (Lin, 2004) between the input and output text span. When calculating the ROUGE score, we consider ROUGE-1, ROUGE-2, ROUGE-L, and combinations of them such as:

$$\lambda_1 \cdot ROUGE{-}1 + \lambda_2 \cdot ROUGE{-}2 + \lambda_3 \cdot ROUGE{-}L \tag{22}$$

Specifically, for fair comparison, we use the same pre-training data as BART (Lewis et al., 2020) as our source corpus for CoCoPretrain. We set the length of the input text span and output span to 128 and 32, respectively, After ranking with the ROUGE score, we select the top 20M samples as our final pre-training data.

We believe this data selection strategy towards pre-training can make sure that there are enough output words that can be generated by copying from the input, which resembles the downstream task and learns our proposed correlational copying mechanism better.

4 Experiments

4.1 Dataset

For downstream applications, we conduct experiments on the news summarization task with CNN/DailyMail dataset and on the dialogue summarization task with SAMSum dataset.

CNN/DailyMail dataset (Nallapati et al., 2016) contains 312K news articles paired with multi-sentence summaries. We use the non-anonymized version used in See et al. (2017), which has 287,226 training samples, 13,368 validation samples and 11,490 test samples.

SAMSum dataset (Gliwa et al., 2019) contains 16K chat dialogues with manually annotated summaries, splited into 14,732 training samples, 818 validation samples, and 819 test samples. We use the version of the dataset with artificial separator (Gliwa et al., 2019), in which utterances are separated with "|".

4.2 Experimental Settings

For simplicity, we warm-start the model parameters with the publicly released pre-trained BART (large) model[1] with 12 layers in both the encoder and decoder, and the hidden size is 1024. The learning rate is set to 3e-5, and learning decay is applied. We use Adam optimizer with $\beta_1 = 0.9$, $\beta_1 = 0.999$, and $\epsilon = 10^{-8}$. We use the dropout with a probability of 0.1 and the gradient clipping of 0.1. The hyper-parameters are set to the values used in BART. We use a clipping distance of 16 when computing positional correlation, Our experiments are conducted with 8 NVIDIA A100 GPUs. We continually pre-train our model with CoCoPretrain, which converges within 1M steps using a batch size of 8000. During decoding, we use beam search with a beam size of 4.

4.3 Experimental Results

We evaluate our model with the official ROUGE toolkit (Lin, 2004). We report the F1 score of ROUGE-1, ROUGE-2, and ROUGE-L. Table 2 and Table 3 show the results on CNN/DailyMail and SAMSum dataset, respectively.

4.3.1 Results on CNN/DailyMail

The first block in Table 2 displays the results of models without pre-training.

[1] https://github.com/pytorch/fairseq/tree/master/examples/bart

Models	**RG-1**	**RG-2**	**RG-L**
Methods without Pre-training			
Lead-3	40.34	17.70	36.57
PGNet	39.53	17.28	36.38
DRM	39.87	15.82	36.90
Bottom-Up	41.22	18.68	38.34
DCA	41.69	19.47	37.92
Methods with Pre-training			
MASS	41.38	19.11	38.42
UniLM	43.33	20.21	40.51
BERTSUMEXTABS	42.13	19.60	39.18
SAGCopy	42.53	19.92	39.44
PEGASUS	44.17	21.47	41.11
T5	43.52	21.55	40.69
ProphetNet	44.20	21.17	41.30
PALM	44.30	21.12	41.41
BART (Reported)	44.16	21.28	40.90
BART (Our implement)	44.12	21.21	40.85
BART + Cont. Pre-train	44.15	21.21	40.87
Pre-trained Models + Copying			
BART + AttnCopy	44.26	21.31	40.98
BART + SAGCopy	44.31	21.35	41.00
CoCoNet	44.39	21.41	41.05
CoCoNet - SemCorrelation	44.30	21.33	41.01
CoCoNet - PosCorrelation	44.19	21.27	40.89
CoCoNet + CoCoPretrain	**44.50**	**21.55**	**41.24**

Table 2: ROUGE F1 scores on the CNN/DailyMail dataset. For a fair comparison, we continue pre-training BART with the same pre-training data but without copying mechanism (i.e., **BART + Cont. Pre-train**).

- **Lead-3** baseline that simply selects the first three sentences in the input document.

- **PGNet** (See et al., 2017) is a hybrid pointer-generator model applying an attentional copy mechanism.

- **DRM** (Paulus et al., 2018) is a deep reinforced model with an intra-attention mechanism.

- **Bottom-Up** (Gehrmann et al., 2018) introduces a content selector that identifies which phrases in the document should be included in the summary. The copying is then constrained to the selected phrases.

- **DCA** (Celikyilmaz et al., 2018) is a reinforcement learning model with deep communicating agents, each of which encodes a subsection of the input text.

The second block are the results of models with pre-training.

- **MASS** (Song et al., 2019) pre-trains the Seq2Seq language model (LM) to predict a span of masked tokens.

- **UniLM** (Dong et al., 2019) unifies bidirectional, unidirectional, and Seq2Seq LM pre-training.

- **BERTSUMEXTABS** (Liu and Lapata, 2019) applies BERT in text summarization. It is a two-stage fine-tuned model that first fine-tunes the encoder on the extractive summarization task and then on the abstractive summarization task.

- **SAGCopy** (Xu et al., 2020b) fine-tunes MASS by incorporating the importance score for source words into the copying module.

- **PEGASUS** (Zhang et al., 2020) adopts gap-sentence generation as the pre-training objective.

- **T5** (Raffel et al., 2020) and **BART** (Lewis et al., 2020) are models with denoising Seq2Seq pre-training.

- **ProphetNet** (Qi et al., 2020) proposes to simultaneously predict the future n-gram at each time step for pre-training.

- **PALM** (Bi et al., 2020) incorporates the copy mechanism into the pre-training model.

First, we can find that the models with pre-training outperform most of the models without pre-training, which shows the effectiveness of pre-training. Second, fine-tuning the BART model with attentional copying (i.e., **BART + AttnCopy**) improve the results over the original BART model we implemented (+ 0.14%/0.10%/0.13% for ROUGE-1/ROUGE-2/ROUGE-L). To evaluate the self-attention guided copy model (SAGCopy) (Xu et al., 2020b), we apply the SAGCopy mechanism to the BART model, obtaining superior results over BART (+ 0.19%/0.14%/0.15% for ROUGE-1/ROUGE-2/ROUGE-L). By comparison, the improvement for our proposed **CoCoNet** model is larger (+ 0.27%/0.20%/0.20% for ROUGE-1/ROUGE-2/ROUGE-L), which proves the necessity of the copying mechanism and superiority of the correlational copying over the attentional copying (paired t-test, p-value<0.05). Third, continue pre-training the CoCoNet model (i.e., **CoCoNet**

+ **CoCoPretrain**) leads to the best performance (+ 0.38%/0.34%/0.39% for ROUGE-1/ROUGE-2/ROUGE-L over the BART model). When we continue pre-training BART with the same pre-training data but without copying mechanism (i.e., **BART + Cont. Pre-train**), the result outperforms **BART** with a small margin, indicating that general pre-training with selected data is not effective, and correlational copying is essential for pre-training. Fourth, we study the effectiveness of semantic and positional correlation between source words (i.e., **SemCorrelation** and **PosCorrelation**, respectively), we can observe that semantic and positional correlation are both useful, and depriving positional correlation decreases the performance larger.

4.3.2 Results on SAMSum

The results on the SAMSum dataset are shown in Table 3.

- **Longest-3** takes three longest utterances as the summary.
- **Fast Abs RL** (Chen and Bansal, 2018) is a hybrid extractive-abstractive model with the policy-based reinforcement learning.
- **TransformerABS** (Vaswani et al., 2017) is the basic Transformer-based Seq2Seq model without pre-training.
- **DynamicConv** (Wu et al., 2018) is a dynamic convolution model based on lightweight convolutions.
- **D-HGN** (Feng et al., 2020) is a dialogue heterogeneous graph network modeling the utterance and commonsense knowledge.
- **TGDGA** (Zhao et al., 2020) is a topic-word guided dialogue method based on the graph attention model.

First, we can find that the models with pre-training outperform the models without pre-training to a significant extent, possibly due to the small size of the dataset. Second, similar to the results on the CNN/DailyMail dataset, the CoCoNet has better performances than attentional copying and self-attention guided copying. Third, continue pre-training the CoCoNet model (i.e., **CoCoNet + CoCoPretrain**) achieves the best performance (+ 1.15%/1.41%/1.45% for ROUGE-1/ROUGE-2/ROUGE-L over the BART model). We can find that the improvement is larger than that on the CNN/DailyMail dataset. Looking into the datasets, we observe that the copying phenomenon is more common in the SAMSum dataset, with 14.4% of the source words reappearing in the target summary, as opposed to 10.7% in the CNN/DailyMail dataset. Thus, our proposed CoCoNet can work more remarkably on the SAMSum dataset.

Models	**RG-1**	**RG-2**	**RG-L**
Baseline Methods			
Longest-3	32.46	10.27	29.92
PGNet	37.27	14.42	34.36
Fast Abs RL	41.03	16.93	39.05
TransformerABS	42.37	18.44	39.27
DynamicConv	45.41	20.65	41.45
D-HGN	42.03	18.07	39.56
TGDGA	43.11	19.15	40.49
BART (Our implement)	51.53	26.48	47.22
BART + Cont. Pre-train	51.58	26.49	47.11
Pre-trained Models + Copying			
BART + AttnCopy	52.03	26.69	47.55
BART + SAGCopy	52.12	26.82	47.80
CoCoNet	52.28	26.97	48.14
CoCoNet - SemCorrelation	52.21	26.87	48.01
CoCoNet - PosCorrelation	52.16	26.79	47.94
CoCoNet + CoCoPretrain	**52.68**	**27.89**	**48.67**

Table 3: ROUGE (RG) F1 scores on the SAMSum dataset.

4.4 Human Evaluation

Since the readability (how easy it is to understand) and informativeness (how much important information is captured) are difficult to measure automatically, three expert annotators are involved to conduct manual evaluation. They rate the readability and the informativeness of 100 instances sampled from the test set on a scale of 1 to 5 (with 5 being the best). Results in Table 4 show that **CoCoNet** outperforms **PGNet** and **BART** models. For informativeness, **CoCoNet** receives comparative results as **BART**, but it shows a significant increase in readability comparing to **BART**, suggesting that correlational copying mechanism is crucial to reducing reading difficulty.

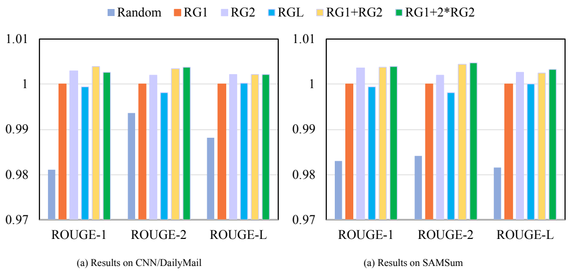
4.5 Effect of Pre-Training Data Selection

We compare various strategies to select pre-training data according to Equation 22 with different values of λ_1, λ_2, and λ_3. The results are shown in Figure 3. Note that the y-axes are normalized by the result of strategy only using ROUGE-1.

First, we find that strategies based on ROUGE

CNN/DailyMail	Informativeness	Readability
PGNet	3.81	3.79
BART	3.97	4.18
CoCoNet + CoCoPretrain	4.01	4.43
SAMSum	Informativeness	Readability
PGNet	3.78	3.25
BART	4.37	4.25
CoCoNet + CoCoPretrain	4.42	4.56

Table 4: Human Evaluation. Two-tailed paired t-test p-value<0.01.



(a) Results on CNN/DailyMail

(a) Results on SAMSum

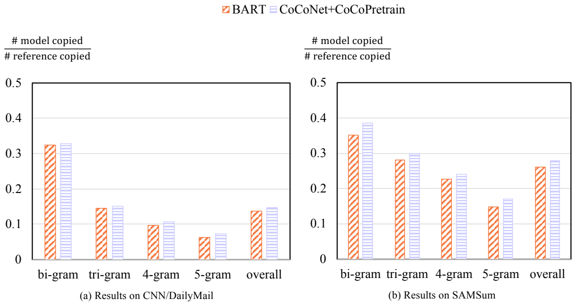
Figure 3: Results of **CoCoNet + CoCoPretrain** model with different pre-training data selection strategies. “RG” is short for “ROUGE”.

are significantly better than Random. Second, among single ROUGE measurements, ROUGE-1 and ROUGE-2 are slightly better than ROUGE-L. Third, combining ROUGE-1 and ROUGE-2 with“λ_1=1 and λ_2=2” achieves the best performance. We can conclude that fitting strategies for pre-training data selection will benefit downstream summarization tasks, and we adopt “ROUGE-1 + 2 * ROUGE-2” in our work.

4.6 Can Our Model Copy More Accurately?

We have demonstrated that CoCoNet improves the summarization model qualitatively and quantitatively. But has our model learned to copy more accurately (especially for the consecutive copying)? Figure 4 shows that the summaries generated by our **CoCoNet+CoCoPretrain** model contain a higher rate of “correct” n-grams (i.e., those that appear both in the input text and reference summary), indicating that learning to copy from the copying history is beneficial to consecutive copies.

On the other hand, we investigate whether our model triggers the over-copying problem (when source words are unnecessarily copied). We find that the average numbers of over-copied words for **BART** and **CoCoNet + CoCoPretrain** are 35.29 and 33.19 on CNN/DailyMail, 8.21 and 7.84 on SAMSum, showing that our model can alleviate over-copying.



(a) Results on CNN/DailyMail

(b) Results on SAMSum

Figure 4: The rate of correctly copied n-grams.

Dialogue
Ernest: hey Mike , did you park your car on our street? *Mike*: no, took it into garage today *Ernest*: ok good *Mike*: why? *Ernest*: someone just crashed into a red Honda looking just like yours *Mike*: lol lucky me
Reference
Mike took his car into garage today. Ernest is relieved as someone had just crashed into a red Honda which looks like Mike’s.
Result of BART
Mike took his car to the garage today. Someone crashed into his car.
Result of CoCoNet
Mike took his car into the garage today. Someone crashed into a red Honda looking like his car.

Table 5: Case study.

4.7 Case Study

Table 5 illustrates an example from the SAMSum dataset. BART generates a summary that is contradictory to the dialogue, saying “*Mike’s car is crashed*”. In fact, the crashed car just looks like Mike’s. By contrast, CoCoNet successfully captures the correlation between “*crashed into*” and “*a red Honda looking like*”. As a result, CoCoNet copies the correct information (highlighted) from the source text through correlational copying and expresses exactly the same idea as the reference.

5 Conclusion

We propose CoCoNet that can take advantage of prior copying distributions and encourage the decoder to copy the source word that is relevant to the previously copied one. We further enhance the copying ability through pre-training with the objective of text span generation. Our model gains new state-of-the-art results on the news summarization and dialogue summarization tasks.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant No. 2020AAA0108600.

References

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. PALM: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. 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 1662–1675, New Orleans, Louisiana.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2020. Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. *arXiv preprint arXiv:2010.10044*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany.

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.

Haoran Li, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020a. Aspect-aware multimodal summarization for chinese e-commerce products. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8188–8195.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020b. Keywords-guided abstractive sentence summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8196–8203.

Junjie Li, Haoran Li, and Chengqing Zong. 2019. Towards personalized review summarization via user-aware sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6690–6697.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, 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.

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

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-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online.

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.

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.

Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, and Abe Ittycheriah. 2016. Temporal attention model for neural machine translation. *arXiv preprint arXiv:1608.02927*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2692–2700.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2018. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 2048–2057.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020a. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020b. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium.

Peng Yuan, Haoran Li, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. On the faithfulness for E-commerce product summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5712–5717, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jiajun Zhang, Yang Zhao, Haoran Li, and Chengqing Zong. 2018. Attention with sparsity regularization for neural machine translation and summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(3):507–518.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339.

Lulu Zhao, Weiran Xu, and Jun Guo. 2020. Improving abstractive dialogue summarization with graph structures and topic words. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 437–449, Barcelona, Spain (Online).

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada. Association for Computational Linguistics.