A General Contextualized Rewriting Framework for Text Summarization

Guangsheng Bao and Yue Zhang, *Member, IEEE*

***Abstract*—The rewriting method for text summarization combines the advantage of extractive and abstractive approaches, improving the conciseness and readability of extractive summaries. Exiting rewriting systems take extractive sentences as the only input and rewrite each sentence independently, which may lose critical background knowledge and break cross-sentence coherence of the summary. To this end, we propose contextualized rewriting to consume the entire document and maintain the summary coherence, representing extractive sentences as a part of the document encoding and introducing group-tags to align the extractive sentences to the summary. We further propose a general framework for rewriting with an external extractor and a joint internal extractor, representing sentence selection as a special token prediction. We demonstrate the framework's effectiveness by implementing three rewriter instances on various pre-trained models. Experiments show that contextualized rewriting significantly outperforms previous non-contextualized rewriting, achieving strong improvements on ROUGE scores upon multiple extractors. Empirical results further suggest that joint modeling of sentence selection and rewriting can largely enhance performance.**

***Index Terms*—Abstractive method, contextualized rewriting, joint model, sentence rewriting, text summarization.**

I. Introduction

Automatic text summarization [1], [2] is the task that compresses input documents into shorter summaries while keeping their salient information. It has been tackled mainly by two basic methods, namely, extractive and abstractive algorithms. The *extractive* method generates a summary by extracting important text pieces (typically sentences) from a document and concatenating them to form the summary [3], [4], [5]. It has the advantage in content selection and faithfulness compared to the abstractive method [6], [7], [8]. However, the extractive sentences may contain irrelevant or redundant information [9], [10], [11] and may have low coherence since the discourse relations and cross-sentence anaphora are not maintained [12], [13].

Source Document: " Success Kid " is likely the Internet 's most famous baby . You 've seen him in dozens of memes , fist clenched in a determined look of persevering despite the odds . Success Kid – now an 8-year-old named Sammy Griner – needs a little bit of that mojo to rub off on his family . *His dad , Justin , needs a kidney transplant* . About a week ago , Laney Griner , Justin 's wife and Sammy 's mother , created a GoFundMe campaign with a goal of $ 75,000 to help cover the medical expenses that go along with a kidney transplant ...

Extractive Summary: *His dad , Justin , needs a kidney transplant* .

Rewritten Summary: " Success Kid " Sammy Griner 's dad , Justin , needs a kidney transplant .

Gold Summary: Justin Griner , the dad of " Success Kid , " needs a kidney transplant .

Fig. 1. Key information from *document context* complements extractive summary, improving its informativeness.

The *abstractive* method uses a conditional language model to generate the summary from scratch token by token [6], [7], [14], producing more fluent and readable content. A line of work proposes a *rewriting* method to combine the advantage of extractive and abstractive methods. It paraphrases the extractive sentences using an abstractive model, removing irrelevant information and normalizing the expressions. Various rewriting systems have been developed, including sentence compression [9], syntax simplification [12], and paraphrasing [10], [15], [16], [17]. The human evaluation shows that rewriting methods can improve the readability and conciseness of extractive summaries [15], [17].

Despite their effectiveness, there are still issues with these rewriting methods. First, these methods generate summaries only according to the extractive sentences. Therefore, if some critical information appears in the document but not in the extractive sentences, it would be impossible for the rewriter to generate them. As an example in Fig. 1 shows, an extractive summary can be created by extracting the salient sentence "*His dad, Justin, needs a kidney transplant.* " A rewriter can rearrange the expression but cannot resolve the entity "*Sammy Griner*" and its mention "*Success Kid*" without considering the document context during rewriting.

Second, these methods rewrite a summary sentence by sentence independently. As a result, the cross-sentence coherence,

Manuscript received 24 July 2022; revised 11 January 2023; accepted 16 April 2023. Date of publication 26 April 2023; date of current version 28 April 2023. This work was supported in part by the Pioneer and Leading Goose R&D Program of Zhejiang under Grant 2022SDXHDX0003, and in part by the National Natural Science Foundation of China under Grant 62161160339. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodan Zhu. *(Corresponding author: Yue Zhang.)*

Guangsheng Bao is with the Zhejiang University, Hangzhou 310007, China, and also with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: baoguangsheng@westlake.edu.cn).

Yue Zhang is with the School of Engineering, Westlake University, Hangzhou 310024, China, and also with the Institute of Advanced Technology, Westlake Institute of Advanced Study, Westlake University, Hangzhou 310024, China (e-mail: yue.zhang@wias.org.cn).

Digital Object Identifier 10.1109/TASLP.2023.3268569

Source Document: ‘ Queen of celebrity ’ is not lifetime role - before Kim Kardashian there was Paris Hilton and now there is another glossy young female fighting for the crown . A model and TV presenter known as the Mexican Kim Kardashian is hoping to dethrone the real Kim after gathering a huge support base on social media .③ Jimena Sanchez , a 30 year-old Mexican model and TV presenter , is four years younger than Kim and works as a sports presenter for the Latin American division of Fox Sports , Fox Deportes .① Scroll down for video . Jimena Sanchez (left) is a model and TV presenter from Mexico who has been labelled the ‘ Mexican Kim Kardashian ’ for her similarity to the reality star (right) .② But her real fame comes from the mass of social media attention that she gets , with one million followers on Twitter and also more than one million Facebook likes on her official page .

Rewritten Summary: Jimena Sanchez is a 30 year-old Mexican model and sports TV presenter .① The star is called ‘ Mexican Kim Kardashian ’ because of their similar looks .② She is now hoping to become the most popular woman on social media .③

Fig. 2. Simple mentions to the entity in *summary context* simplify the complex expression of subjects in extractive sentences and improves its coherence.

such as entity coreference, is not modeled and controlled. Take the case in Fig. 2 as an example. The subjects of the three extractive sentences are redundant and reference to the same entity “*Jimena Sanchez*”, which requires simple mentions such as “*the star*” and “*she*” in the summary sentences for conciseness and better coherence. A rewriter needs to consider the summary context of each sentence in order to remove redundancy and improve its readability.

We propose contextualized rewriting, considering both document and summary context during rewriting, rather than conditioning only on the extractive sentence. In particular, we represent extractive sentences as a part of the input document representation, introducing group-tags to represent the alignment between summary and extractive sentences. Specifically, as Fig. 2 shows, we allocate the group-tags ①, ②, and ③ to the first, second, and third summary sentences. We mark the corresponding extractive sentences using the same group-tags, through which the decoder can locate the corresponding extractive sentence during rewriting.

Based on the scheme, we propose a general seq2seq framework with group-tag alignments for rewriting with an external or a joint internal extractor, representing sentence selection as one of the token prediction steps during decoding. This framework is independent of the implementation details of the seq2seq model, thus common for different abstractive rewriters. We instantiate three rewriters by applying the general framework to BERT [18] and BART [19] models that include two rewriters with an external extractor naming BERT-Rewriter and BART-Rewriter, and one rewriter with an internal extractor naming BART-JointSR.

We evaluate the three rewriters using the popular benchmark dataset CNN/DailyMail. The results suggest that contextualized rewriting can significantly improve both the ROUGE [20] scores and human scores compared with previous non-contextualized rewriting methods. The performance improvement is further enlarged when contextualized rewriting works with the stronger pre-trained model BART. We further prove that the joint modeling of sentence selection and rewriting is much more beneficial than the separate pipeline model in terms of both the efficiency and the effectiveness. To our best knowledge, we are the first rewriting method that improves ROUGE scores and human scores against both extractive and abstractive baselines.

This article significantly extends our conference work [21], which describes an instance of the general framework using BERT. The addition of this article is three-fold. First, we apply contextualized rewriting to the pre-trained model BART and demonstrate the effectiveness of our method on a stronger baseline. Second, we explore the possibility of joint modeling of sentence selection and rewriting, further powering our rewriting method by removing the dependence on an external extractor and demonstrating the advantage of joint modeling. Last, we generalize our contextualized rewriting design to a framework that suits both rewriters with an external extractor and an internal extractor, which can be applied to different seq2seq models. Our code and models are released at.[1]

II. Related Work

Rewriting methods are widely used in NLP applications, converting text from one form to another while keeping the original semantics. In information extraction, rewriting methods are used to generate different expressions of a query to increase the recall of retrieval [22]. In conversational question answering, rewriting methods are used to convert questions from a context-dependent expression to a self-contained expression with referred entities explicitly mentioned and omitted information recovered [23], [24], [25]. In neural machine translation, rewriting methods are used to rewrite the retrieved templates into translations by filling information from source sentences [26]. In text summarization, rewriting methods are used to compress sentences [9], simplify expressions [12], or fill templates [27]. Studies [10], [15], [17] show that rewriting methods can enhance extractive summaries on conciseness and readability while reserving critical information. We further develop these rewriting methods to consider document context and summary context during rewriting.

Recent studies model sentence rewriting as conditioned text generation taking extractive sentences as inputs. Chen and Bansal [10] use an auto-regressive sentence extractor and a

[1][Online]. Available: https://github.com/baoguangsheng/bart-rewriters

seq2seq model with the copy mechanism [14] to rewrite extractive sentences one by one. They tune the extractor using reinforcement learning with reward signals from rewritten sentences and rerank rewritten summaries to avoid repetition. Bae et al. [15] adopt a similar strategy but use a BERT document encoder and reward signals from the whole summary. Wei et al. [16] use a BERT document encoder to generate sentence embeddings and upon which a binary classifier is trained to select sentences. They use a Transformer decoder [28] equipped with the copy mechanism to generate summary sentences. Xiao et al. [17] use a pointer network to select sentences and make a decision of copying or rewriting accordingly. If the model chooses to rewrite, a vanilla seq2seq model will be used to rewrite the sentence. The model decisions on sentence selection and copying/rewriting are tuned using reinforcement learning. Compared with these methods, our rewriting method is *computationally simpler* for both training and inference since we do not use reinforcement learning and the copy mechanism as above. Furthermore, as mentioned earlier, in contrast to these pure sentence rewriting methods, we consider the *document context* and the *summary context* during rewriting, thereby improving information recall and cross-sentence coherence. In addition, we propose a joint model of internal extractor and rewriter, reducing the total model size, the training cost, and the inference cost of the pipeline approach.

Some hybrid extractive and abstractive summarization models align with our work. Hsu et al. [29] propose inconsistency loss to encourage the consistency between word-level and sentence-level attentions. Cheng and Lapata [13] extract important words first and use them to constrain a language model to generate a summary. Gehrmann et al. [11] propose a bottom-up method, selecting important words using a neural classifier and restricting the copy source of a pointer-generator network to the chosen words to generate a summary. Li et al. [30] first generate some keywords and then generate summary accordingly. Saito et al. [31] extract keywords using a saliency model before summary generation. Dou et al. [32] use a summary-level extractive model to guide an abstractive model to produce summary. Similar to our method, they use extracted content to guide the abstractive summarizer. However, different from these methods, which focus on word level or summary level, we investigate *sentence-level* constraints for guiding abstractive *rewriting*.

Our rewriting method can also be regarded as an abstractive summarizer [6], [7], [14] with attention guidance using group-tags, where the group-tags are generated from the extractive summary. In comparison with the vanilla abstractive model, the advantages are three-fold. First, extractive summaries can guide the abstractive summarizer with more salient information. Second, the training difficulty of the abstractive model can be reduced when important contents are marked in the inputs. Third, the summarization procedure is made more interpretable by associating a crucial source sentence with each target sentence.

III. Problem Formulation

We formulate contextualized rewriting as conditioned text generation that maximizes the probability of the summary given an input document. We explore two alternatives of contextualized rewriters: a rewriter with extractive summaries provided externally and a rewriter with an internal extractor. The relative advantage of the former is generality, which can be used on top of arbitrary extractive summarizers, while the advantage of the latter is model independence.

Formally, we use X to denote an input document and Y an output summary

$$X = \{X_i\}|_{i=1}^{|X|} \text{ and } Y = \{Y_j\}|_{j=1}^{|Y|},$$

where X_i denotes each sentence in X and $|X|$ the number of sentences. Y_j denotes each sentence in Y and $|Y|$ the number of sentences in Y. Here, we use a sentence-level notation instead of word-level or token-level for the simplicity of discussions because we focus on sentence-level relations between the input document, the extractive summary, and the rewritten summary.

A. Rewriter for External Extractive Summary

Given extractive summary E, our contextualized rewriter rewrites it into a final summary Y that

$$\hat{Y} = \arg\max_{Y} P(Y|E, X), \tag{1}$$

where Y and E have the same number of sentences.

As shown in Fig. 3, we use external extractor to select sentences from the input document X and obtain the extractive summary E that

$$E = \{E_j = X_k | X_k \in X\}|_{j=1}^{|E|}, \tag{2}$$

where $|E|$ denotes the number of sentences, which is equal to $|Y|$. Each sentence E_j is from a document sentence X_k. For instance, given a document $X = \{X_1, X_2, \ldots, X_6\}$, an extractive summary could be $E = \{E_1 = X_3, E_2 = X_5, E_3 = X_2\}$ that the first extractive sentence E_1 is from the third document sentence X_3, and so forth.

The contextualized rewriter rewrites each extractive sentence E_j into a summary sentence Y_j given the document context and summary context. Therefore, the problem formulated in (1) can be further detailed as

$$\hat{Y} = \arg\max_{Y} \prod_{j=1}^{|Y|} P\left(Y_j | \underbrace{E_j}_{\text{rewriting source}}, \underbrace{E_{<j}}_{\text{extractive context}}, \underbrace{Y_{<j}}_{\text{summary context}}, \underbrace{X}_{\text{document context}}\right). \tag{3}$$

It is worth noting that contextualized rewriting as expressed in (3) is different from previous non-contextualized rewriters [15], [16], [17], which do not use contextual information from X but directly calculate $P(Y_j|E_j)$.

B. Joint Internal Extractor and Rewriter

The external extractor can be integrated into contextualized rewriter, resulting in a self-contained rewriter, which models both sentence extraction and rewriting in one seq2seq model. The joint problem can be defined as

$$\hat{Y} = \arg_Y \max_{Y,E} P(Y, E|X), \tag{4}$$

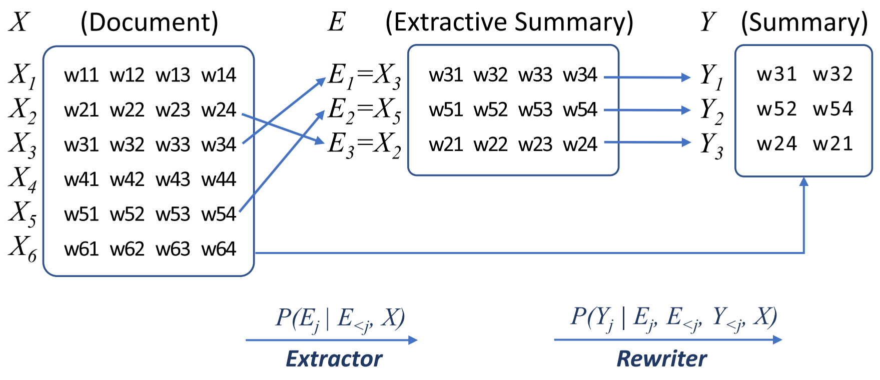

Fig. 3. Contextualized rewriting involves an extractor selecting sentences from the input document and a rewriter generating a summary according to both the selected sentences and the document. (Each line in the rectangles represents a sentence, and each element w# denotes a word.).

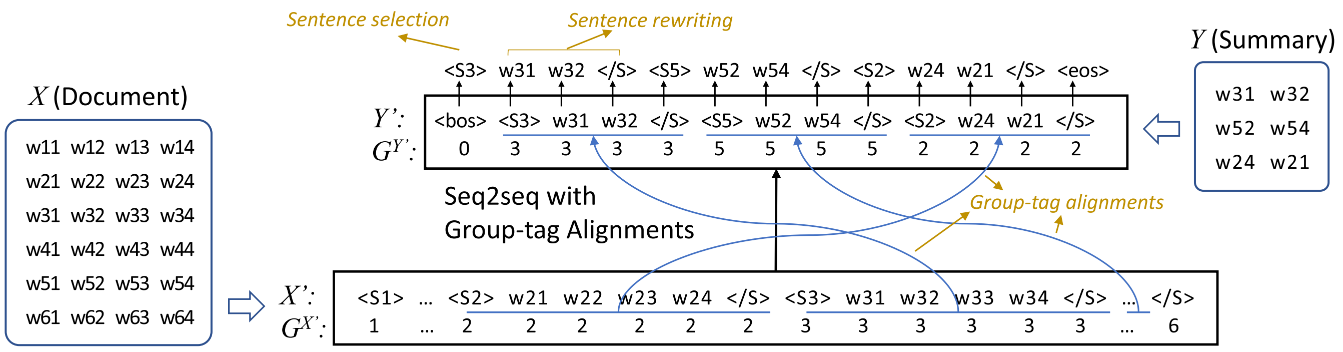

Fig. 4. A general framework for contextualized rewriting which involves group-tag alignments, sentence selection, and sentence rewriting. Identifier tokens <Sk> are used to represent the beginning of sentences. The special tokens <bos> and <eos> are the begin-of-sequence and the end-of-sequence, respectively.

where E is the selected sentences from X, which is defined as the same as (2). The equation can be further expanded as

$$\hat{Y} = \arg_Y \max_{Y,E} \prod_{j=1}^{|Y|} P(Y_j, E_j | Y_{<j}, E_{<j}, X), \tag{5}$$

where both the sentence selection E_j and the rewriting Y_j depend on the history of selection and rewriting.

Given that the generation of Y_j depends on the sentence selection of E_j, we separate the two decision steps as

$$\hat{Y} = \arg_Y \max_{Y,E} \prod_{j=1}^{|Y|} P$$
$$(Y_j | E_j, E_{<j}, Y_{<j}, X) P(E_j | E_{<j}, Y_{<j}, X), \tag{6}$$

in which the current sentence selection E_j depends on the previously selected sentences and rewritten sentences, while current sentence rewriting Y_j depends on the current sentence selection and the history.

It is worth noting that the internal extractor $P(E_j|E_{<j}, Y_{<j}, X)$ in (6) is different from the external extractor $P(E_j|E_{<j}, X)$ in Fig. 3 because of the joint modeling of Y and E. In addition, comparing detailed expressions in (3) and (6), we could find that they share the same rewriting part $P(Y_j|E_j, E_{<j}, Y_{<j}, X)$, for which we will propose a general implementation in next section.

IV. A General Framework: Seq2seq With Group-Tag Alignments

The overall structure of our model is shown in Fig. 4, which can be seen as a conceptually simple extension of the standard seq2seq model with group-tag-guided alignments. The alignments are expressed in two group-tag sequences, identifying each sentence alignment by the same group-tag in both sequences. We extend both the input document and output summary with special sentence identifiers so that we can generate group-tag sequences accordingly. We transfer the sentence selection problem into a problem of predicting the sentence identifier tokens, which can be integrated into the seq2seq framework easily.

A. Group-Tag Alignments

We introduce sentence identifier tokens such as <S1> and <S2> to represent the beginning of each sentence, extending document X to X' and summary Y to Y'.

1) Group-Tag Generation: We generate group-tag sequences according to these identifiers. For a sentence beginning with <Sk>, we assign group-tag k to each token in that sentence. For example, we assign group-tag to each token of a document like "<S3/>3 w31/3 w32/3 </S>/3 <S5>/5 w52/5 w54/5 </S>/5". Formally, we use $tokens = \{w_i\}|_{i=1}^{N}$ to denote the mixed sequence of word tokens and identifier tokens. Given

Algorithm 1: Generate Group-Tags From a Mixed Sequence of Word Tokens and Identifier Tokens.

```
   Input: $tokens = \{w_i\}|_{i=1}^{N}$
   Output: $tags = \{t_i\}|_{i=1}^{N}$
1: function GroupTag$tokens$
2:   $t \leftarrow 0$    ▷ default group-tag.
3:   for $i \leftarrow 1$ to $N$ do
4:     if $w_i ==$ "<S$k$>" then
5:       $t \leftarrow k$    ▷ new group-tag for new sentence.
6:     $t_i \leftarrow t$    ▷ group-tag for the current sentence.
7:     for $w_i ==$ "</S>" do
8:       $t \leftarrow 0$    ▷ reset group-tag if a sentence ends.
9:   return $tags$
```

the $tokens$ sequence, we generate the group tag sequence $tags = \{t_i\}|_{i=1}^{N}$ by Algorithm 1.

2) Alignment Representation: As Fig. 4 shows, we use group-tag sequences $G^{X'}$ and $G^{Y'}$ to represent the extractive sentences E and its one-one mapping to summary sentences Y'. Take the case in the figure as an example, the $G^{X'} = \{1, \ldots, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, \ldots, 6\}$ and $G^{Y'} = \{3, 3, 3, 3, 5, 5, 5, 5, 2, 2, 2, 2\}$, in which the tokens in Y' corresponding to group-tag 3 are generated by rewriting the tokens in X' corresponding to the same group-tag 3.

Given group-tag sequences $G^{X'}$ and $G^{Y'}$, we could remove E from (3) and reformulate contextualized rewriting as

$$P(Y_j|E_j, E_{<j}, Y_{<j}, X) = \prod_k P(Y'_k|G^{Y'}_{<k}, Y'_{<k}, G^{X'}, X'), \tag{7}$$

where Y'_k denotes the k-th token in Y', $Y'_{<k}$ the tokens before k, and $G^{Y'}_{<k}$ the group-tags before k that k ranges from the beginning to the end of sentence Y_j. Note that the probability on the left of the equation is expressed on sentences, while that on the right is expressed on tokens. We obtain group-tag sequence by Algorithm 1, that

$$G^{X'} = \textsc{GroupTag}(X'),$$
$$G^{Y'}_{<k} = \textsc{GroupTag}(Y'_{<k}).$$

3) Group-Tag Representation: In the encoder-decoder framework, we convert $G^{X'}$ and $G^{Y'}$ into vector representations through a shared embedding table, which is randomly initialized and jointly trained with the encoder and decoder. The vector representations of $G^{X'}$ and $G^{Y'}$ are used to enrich vector representations of X' and Y', respectively. As a result, all the tokens tagged with k in both X' and Y' have the same vector component, through which content-based addressing can be done through the attention mechanism [33]. Here, the group tag serves as a mechanism to constrain the attention from Y' to the corresponding part of X' during decoding. Unlike approaches that modify a seq2seq model by using rules [11], [29], group tags are more flexible and trainable.

B. Document Encoding

We generate group-tags $G^{X'}$ from the input X' and apply the group-tag embeddings upon the document representation to produce the final representation

$$G^{X'} = \textsc{GroupTag}(X'),$$
$$x = \textsc{Encoder}(X'),$$
$$x = x + \textsc{Emb}_{tag}(G^{X'}), \tag{8}$$

where $\textsc{Encoder}$ denotes the encoder of a seq2seq model, and $\textsc{Emb}_{tag}$ denotes the embedding table of group-tags.

C. Summary Decoding

We extend a standard Transformer decoder with group-tag alignments. We generate group-tag sequence $G^{Y'}$ from summary Y' and convert these group-tags into embeddings, adding them to token embeddings and position embeddings for the input representation

$$G^{Y'} = \textsc{GroupTag}(Y'),$$
$$y = \textsc{Emb}_{token}(Y') + \textsc{Emb}_{pos}(Y') + \textsc{Emb}_{tag}(G^{Y'}), \tag{9}$$

where $\textsc{Emb}_{token}$ denotes the token embedding table, $\textsc{Emb}_{pos}$ the position embedding table, and $\textsc{Emb}_{tag}$ the group-tag embedding table.

We predict the generation tokens according to encoder output and decoder history. The rewriting formula in (12) can be implemented by

$$P(Y'_k|G^{Y'}_{<k}, Y'_{<k}, G^{X'}, X') = \textsc{Decoder}(y_{<k}, x), \tag{10}$$

where $\textsc{Decoder}$ denotes the Transformer decoder, y represents the tagged token embeddings, and x the encoder outputs. Because of the same vector components of group-tags in x and y, the decoder addresses the k-th extracted sentence in x when generating the k-th rewritten sentence.

1) Sentence Selection: As shown in Fig. 4, on the decoder side of self-contained rewriter, we use identifier tokens to denote the corresponding extractive sentences (e.g.<S7> and <S1>). Therefore, the problem of sentence selection is transformed into a problem of predicting sentence identifier tokens, which can be modeled as one step in the decoding sequence. Formally, the sentence selection part in (6) can be transformed to

$$P(E_j|E_{<j}, Y_{<j}, X) = P(Y'_k|G^{Y'}_{<k}, Y'_{<k}, G^{X'}, X'), \tag{11}$$

where the k-th token of the extended sequence Y' is the identifier token of j-th selected sentence.

2) Sentence Rewriting: We can see that (7) and (11) have the same form and the k in (11) is followed by the k in (7). We merge them into one decoding sequence, which makes sentence selection first and then do contextualized rewriting. Therefore, a general formula for both (3) and (6) is

$$\hat{Y'} = \arg\max_{Y'} \prod_{k=1}^{K} P(Y'_k|G^{Y'}_{<k}, Y'_{<k}, G^{X'}, X'), \tag{12}$$

where K denotes the number of tokens in the summary Y', and X' is extended from X while Y' is extended from Y.

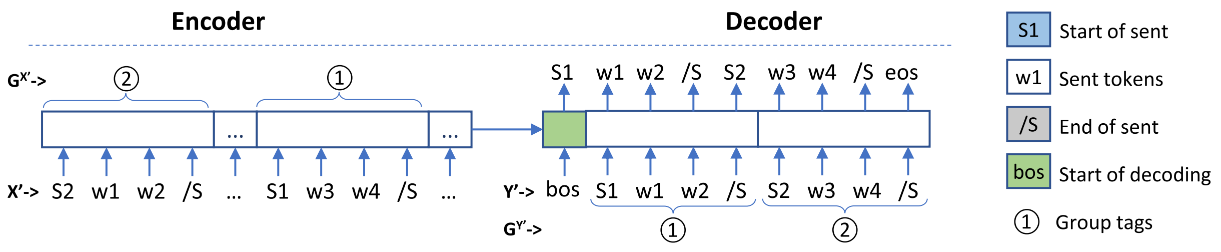

Fig. 5. **BERT-Rewriter** and **BART-Rewriter** use extractive sentences provided by external extractor, such as the sentences denoted by the sentence identifiers <S2> and <S1> which are displayed as "S2" and "S1" in the encoder input. The sentence selection does not play a role in the models as it only predicts incremental identifiers.

The formula is the same for the rewriter with an external extractive summary and the rewriter with an internal extractor, but the group-tag sequences are different. For the former, the group-tag sequence $G^{X'}$ tells where the extractive sentences are located in X'. In contrast, the group-tag sequence $G^{Y'}$ is just naturally ordered and increases per sentence. For the latter, the group-tag sequence $G^{X'}$ is naturally ordered (the number increases per sentence), and the group-tag sequence $G^{Y'}$ indicates the location of the extractive sentences in X'.

D. Training and Inference

We *train* our contextualized rewriter on a pre-processed dataset labeled with oracle extractions E.

1) Oracle Extractions: To generate oracle extraction, we match each sentence in the human summary to each document sentence, choosing the document sentence with the best matching score as the oracle extraction. Specifically, we use the average recall of ROUGE-1/2/L as the scoring function, which follows Wei et al. [16]. Differing from existing work [5], which aims to find a *set* of sentences that maximizes ROUGE matching with the whole summary, we find the best match for each summary sentence. As a result, the number of extracted sentences is the same as the number of sentences in the human summary. This strategy is also adopted by Wei et al. [16] and Bae et al. [15].

2) Loss Function: We train contextualized rewriter using MLE loss, but with a different weight assigned to identifier tokens because they serve for sentence selection

$$Loss = \sum_{j=k}^{|Y'|} -w_j * \log P(Y'_k | G^{Y'}_{<k}, Y'_{<k}, G^{X'}, X'),$$
$$w_j = \gamma \text{ if } Y'_k \in \text{Indentifiers else } 1, \tag{13}$$

where sequence $G^{X'}$ and $G^{Y'}$ are built according to E. *Identifiers* is the set of identifier tokens and γ is a hyper-parameter determined by searching on the development set.

During *inference*, we generate group tags incrementally with the tokens predicted. The group tag of the next decoding step is uniquely determined by the tokens that have been generated.

V. Instances

A. Instance I: BERT Rewriter

We describe our conference work BERT rewriter [21] as a first instance of the framework, which follows the architecture design of Fig. 5, using a pretrained BERT [18] as the document encoder and a randomly-initialized Transformer [28] decoder as the summary decoder. The rewriter extends the abstractive summarizer of [5] with group-tag alignments between the encoder and the decoder.

In order to adapt to the *input* definition of BERT model, we convert the document input X' before feeding it to BERT model. We replace the identifier tokens $<Sk>$ to a general sentence representation token [CLS] and the end-of-sentence token $</S>$ to [SEP], so that the BERT encoder can consume. The randomly-initialized decoder shares the BERT token embedding table. The identifier tokens $<Sk>$ in summary are replaced with [SEP]. During decoding, the first [SEP] is translated as <S1>, the second <S2>, and so on.

Because we use a pre-trained model for the encoder and a randomly-initialized model for the decoder, we *train* them using different learning-rate and warmup schedules that

$$lr_{\text{ENC}} = 0.002 \cdot \min\left(step^{-0.5}, step \cdot warmup_{\text{ENC}}^{-1.5}\right),$$
$$lr_{\text{DEC}} = 0.2 \cdot \min\left(step^{-0.5}, step \cdot warmup_{\text{DEC}}^{-1.5}\right). \tag{14}$$

For *inference*, we constrain the decoding sequence to a minimum length of 50, a maximum length of 200, a length penalty [34] with $\alpha = 0.95$, and a beam size of 5. During beam search, we block the paths on which a repeated trigram is generated, namely Trigram Blocking [35].

B. Instance II: BART Rewriter

As a second instance, we apply the general framework described in Section IV to a pretrained BART [19], extending it with the rewriting mechanism as Fig. 5 describes.

BART-Rewriter relies on an external extractor to select the salient source sentences. During training, the extractive sentences are selected by a matching algorithm that each summary sentence is matched to a source sentence. The group-tags for the summary follow a natural order for each sentence, while the group-tags for the source document may not. During inference, we can use BERT-Ext [21] to select sentences.

C. Instance III: BART-JointSR (BART Joint Selector and Rewriter)

As a third instance, we build a joint model of internal extractor (selector) and rewriter - BART-JointSR. As Fig. 6 shows, similar to BART-Rewriter, we apply the general framework to the

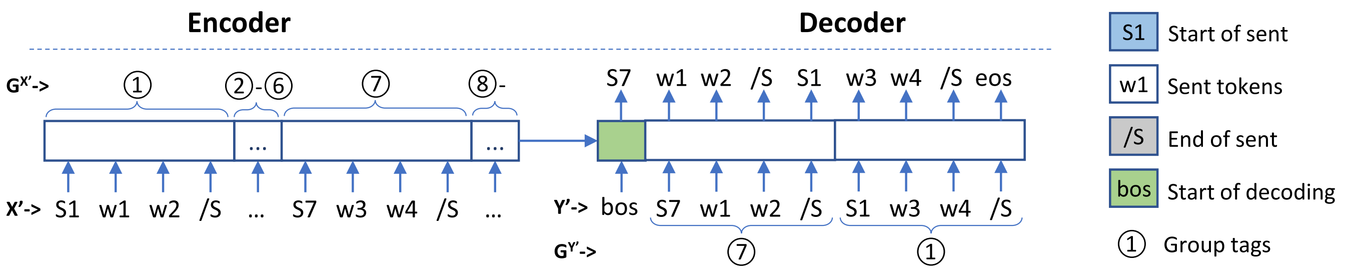

Fig. 6. **BART-JointSR** jointly models sentence selection and rewriting, using identifier tokens such as <S7> and <S1> (displayed as "S7" and "S1") in the decoder output to indicate the selected sentences.

pre-trained BART, but instead of an external extractor, we use an integrated extractor.

To train BART-JointSR, we generate different training inputs that sentence identifiers are generated using a different strategy. We use sentence identifiers with natural order for the input document but sentence identifiers corresponding to the document sentence for the summary. Therefore, the sentence identifiers in the decoder output indicate the sentence selections. During inference, we use beam-search to find the best selection and rewriting combination.

VI. Experimental Setup

We experiment on the standard benchmark dataset CNN/DailyMail [36], a single-document summarization dataset comprising 312,085 online news articles with an average of 766 words per article and human written highlights with an average of 3.75 sentences per sample. We follow the standard splitting of Herman et al. [36], containing 287,227 training samples, 13,368 validation samples, and 11,490 testing samples. We use the non-anonymous version and preprocess the dataset following the BART baseline [19].

We evaluate the quality of summaries using the automatic metric ROUGE [20], reporting ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L). The ROUGE-1/2 represent the n-gram overlap between the generated summary and the gold reference, and ROUGE-L reflects the longest common sub-sequence between the generated summary and the gold reference.

VII. Automatic Evaluation

A. Main Results

We evaluate our rewriters on the dataset CNN/DailyMail, reporting ROUGE-1/2/L (R-1/2/L). Table I compares our model with previous extractive models, abstractive models, and rewriters. In particular, compared to previous BERT based extractive baseline BERTSUMEXT and abstractive baseline BERTSUMEXTABS, our BERT-Rewriter (base) outperforms them for an average of 0.5 and 1.2 ROUGE points, respectively. Compared with previous BERT based rewriters, our BERT-Rewriter (base) achieves the best scores, outperforming BERT+Copy/Rewrite+HRL for 1.0 points on average and especially 1.2 points on ROUGE-L, despite that our rewriter is purely abstractive without leveraging complex techniques such

TABLE I
Main Results on CNN/DailyMail

Method	R-1	R-2	R-L
Extractive			
LEAD-3 [14]	40.34	17.70	36.57
BERTSUMEXT [5]	43.25	20.24	39.63
Abstractive			
BERTSUMABS [5]	41.72	19.39	38.76
BERTSUMEXTABS [5]	42.13	19.60	39.18
BART (large) [19]	44.16	21.28	40.90
RNN-Ext+Abs+RL [10]	40.88	17.80	38.54
BERT-Hybrid [16]	41.76	19.31	38.86
BERT-Ext+Abs+RL [15]	41.90	19.08	39.64
BERT+Copy/Rewrite+HRL [17]	42.92	19.43	39.35
Our Models			
BERT-Ext	41.04	19.56	37.66
+ BERT-Rewriter (base)	43.52	20.57	40.56
+ BART-Rewriter (base)	43.52	20.76	40.61
+ BART-Rewriter (large)	44.26	21.23	41.34
BART-JointSR (large)	**44.72***	**21.78***	**41.70***
Ensemble Methods			
GSum [34]	45.94	22.32	42.48
SimCLS [40]	46.67	22.15	43.54
BRIO [41]	47.78	23.55	44.57

The best scores are in bold, and significantly better scores than bart (large) are marked with * ($p < 0.001$, t-test). Ext and abs denote extractive and abstractive models, respectively, and rl means reinforcement learning.

as copying mechanism and reinforcement learning. These results suggest the effectiveness of our rewriting method and the advantage of contextualized rewriting compared to non-contextualized rewriting.

Using a weak BERT extractor, our BART-Rewriter (large) enhances the scores by a moderate 0.44 on ROUGE-L. However, when an internal extractor is used, our BART-JointSR enhances the performance over BART (large) by 0.56, 0.50, and 0.80 on R-1/2/L, respectively. In comparison with our previous BERT-Rewriter [21], BART-JointSR improves the performance by 1.20, 1.21, and 1.14 on ROUGE-1/2/L, respectively. The results demonstrate that our rewriting method works with stronger pre-trained model and benefits from joint modeling.

The architecture and size of pre-trained models are important factors influencing the performance of our rewriters. As we can see, our BART-Rewriter (base) has similar ROUGE scores as BERT-Rewriter (base). Although the BART-Rewriter is fine-tuned on the seq2seq pre-trained BART and the BERT-Rewriter is fine-tuned on a pre-trained BERT encoder and a randomly initialized decoder, the BART-Rewriter does not

TABLE II
CONTEXTUALIZED REWRITERS WORK WITH DIFFERENT EXTRACTORS, ENHANCING THEIR PERFORMANCE

Method	R-1	R-2	R-L	#W
Oracle	46.77	26.78	43.32	112
+ BERT-Rewriter	52.57 (+5.80)	29.71 (+2.93)	49.69 (+6.37)	63
+ BART-Rewriter	55.01 (+8.24)	32.07 (+5.29)	52.08 (+8.76)	64
LEAD3	40.34	17.70	36.57	85
+ BERT-Rewriter	41.09 (**+0.75**)	18.19 (**+0.49**)	38.06 (**+1.49**)	55
+ BART-Rewriter	41.29 (**+0.95**)	18.49 (**+0.79**)	38.22 (**+1.65**)	53
BERTSUMEXT	42.50	19.88	38.91	80
+ BERT-Rewriter	43.31 (**+0.81**)	20.44 (**+0.56**)	40.33 (**+1.42**)	54
+ BART-Rewriter	43.35 (**+0.85**)	20.70 (**+0.82**)	40.55 (**+1.64**)	50
BERT-Ext	41.04	19.56	37.66	105
+ BERT-Rewriter	43.52 (**+2.48**)	20.57 (**+1.01**)	40.56 (**+2.90**)	66
+ BART-Rewriter	44.26 (**+3.22**)	21.23 (**+1.67**)	41.34 (**+3.68**)	66

The column #w represents the average number of words in summary.

TABLE III
JOINT MODELING OF SENTENCE SELECTION AND REWRITING FURTHER IMPROVES THE PERFORMANCE

Method	R-1	R-2	R-L	#W
BERT-Ext	41.04	19.56	37.66	105
+ BART-Rewriter	44.26	21.23	41.34	66
+ BART-Rewriter with reorder	44.38	21.31	41.43	61
BART-JointSR	44.72	21.78	41.70	63
- extractive summaries	39.45	18.37	35.75	110
- dedup extractive summaries	42.43	20.06	38.64	93
BART-JointSR with two-stage	43.57	20.78	40.57	57
- extractive summaries	37.12	16.81	33.24	131
- dedup extractive summaries	42.06	19.62	38.04	88

The performance will be much lower if we separate selection and rewriting, as the method bart-jointsr with two-stage shows.

outperform the BERT-Rewriter because it has about 14% fewer parameters. Our BART-Rewriter (large) gives much better scores than BART-Rewriter (base), improving the scores by 0.74, 0.47, and 0.73 on ROUGE-1/2/L, respectively. The results suggest that contextualized rewriting method can work with various pre-trained models, taking advantage of a larger model and further enhancing the model performance.

Table I additionally shows the results of ensemble methods in the literature, in which BRIO [38] gives the strongest results. However, the ensemble methods are not directly comparable with the single model. In addition, ensemble models are costly to train. For instance, it takes 20 hours per epoch and total 15 epochs to train BRIO on 4 NVIDIA RTX 3090 GPUs, as compared with 3 hours per epoch and total 5 epochs by our model. We do not take latest BRIO as our baseline, given that it is at a different line of work and its reranking technique is orthogonal to our method.

B. Universality of Rewriter With External Extractive Summary

Contextualized rewriters learn to compress, paraphrase, and abstract the extractive sentences into more concise and readable expressions. Although the rewriters are trained using oracle extractions, the learned rewriters are not limited to these oracle extractive sentences.

We evaluate the rewriters with various external extractors, including LEAD-3, BERTSUMEXT, BERT-Ext, and Oracle. As shown in Table II, the rewriters enhance the quality of summaries generated by all four extractors. In particular, working with the basic extractor LEAD-3, BERT-Rewriter improves the average ROUGE score by 0.91, while BART-Rewriter improves the average score by 1.13. Even working with the best extractor BERTSUMEXT, BERT-Rewriter and BART-Rewriter enhance the average ROUGE score by 0.93 and 1.10, respectively. The improvements on ROUGE-L are most significant, which are more than 1.4 points for all extractors, suggesting a strong enhancement in fluency.

It is worth noting that for BERTSUMEXT, we conduct decoding without Trigram Blocking [35], which blocks beam search paths containing repeated trigrams. As a result, the extractive summaries contain more redundant information. However, when we apply our rewriters to them, the redundancy can be reduced, leading to higher scores.

C. Advantage of Joint Internal Extractor and Rewriter

Our joint rewriter BART-JointSR (large) gives the best ROUGE scores, outperforming BART-Rewriter for 0.46, 0.55, and 0.36 on ROUGE-1/2/L, respectively. We hypothesize that the improvement comes from two advantages of the joint model compared to the pipeline model. First, BART-JointSR models sentence selection in an auto-regressive manner, therefore, the order of selected sentences is optimized, and the redundancy between sentences is reduced. Second, BART-JointSR jointly models sentence selection and rewriting, so that not only rewriting depends on selection history but vice versa. As a result, the sentence selection and rewriting are better matched with each other.

We conduct various experiments to verify our hypothesis. As shown in Table III, we first study the impact of the order of selected sentences. We alter BART-Rewriter model to enable the decoder to decide the sentence order before rewriting, naming BART-Rewriter with reorder. Reordering extractive sentences can improve the ROUGES scores by about 0.1 on average, which is although small but confirms our first hypothesis about sentence order.

We further compare the independent extractor BERT-Ext and the inner auto-regressive extractor in BART-JointSR by concatenating the selected sentences. As the item "extractive summaries" under BART-JointSR shows, the ROUGE scores are much lower than that of BERT-Ext, which seems contradictive with our first hypothesis about reducing redundancy. However, when we remove the repetitive extractive sentences, we obtain much high scores, as the item "dedup extractive summaries" under BART-JointSR illustrates. Given our rewriter's firm information compression and redundancy reduction abilities, these duplicate extractive sentences are rewritten into different summary sentences without redundant information. It suggests that combining an auto-regressive extractor and a contextualized rewriter may be the key to its high performance.

To clarify the impact of the auto-regressive extractor, we have a closer look at the distribution of extractive sentences over the sentence position in input document. As Fig. 7 shows, the sentences extracted by BART-JointSR have better distribution than that by BERT-Ext, which is much closer to the oracle extractions. We further observe that the oracle extractions also contain duplicate sentences, and we compare the distribution

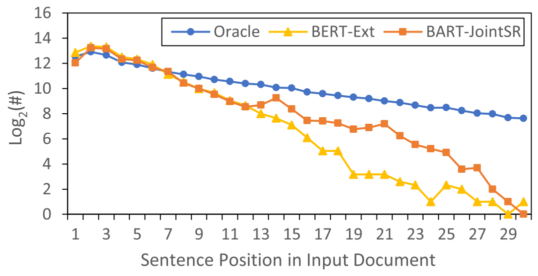

Fig. 7. Distribution of extractive sentences show that the sentences selected by joint rewriter is closer to oracle extractions than BERT-Ext.

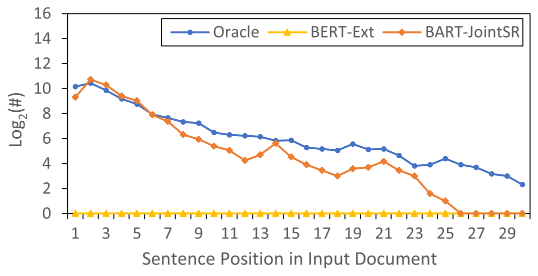

Fig. 8. Distribution of *duplicate* extractive sentences. Joint rewriter generates duplicate extractions and the distribution is close to oracle extractions. BERT-Ext do not generate duplicate extractions, so that we use 0 to denote it.

of duplicate extractive sentences in Fig. 8. We can see that the distribution of duplicate sentences extracted by BART-JointSR is close to that of oracle extractions. In contrast, BERT-Ext only returns identical extractions because it independently makes classification decisions on each sentence.

To verify our second hypothesis about the advantage of joint modeling, we evaluate an alternative of BART-JointSR, naming BART-JointSR with two-stage, where sentence selections and sentence rewriting separated in the decoder. Specifically, for a BART-JointSR decoding sequence "<S3>... <S5>... <S2>...", we put all sentence selection steps together at the beginning of the decoding sequence as "<S3> <S5> <S2> </S> <S3>... <S5>... <S2>...". Therefore, sentence selection does not depend on rewriting, but rewriting still depends on sentence selection. As shown in Table III, BART-JointSR is much better than the two-stage version, outperforming it for more than 1 ROUGE point on R-1/2/L. The results suggest that the extractor can do a better job when it is conditioned on both previously selected and rewritten sentences, which shows the advantage of joint modeling.

VIII. Human Evaluation

The contextualized rewriter can improve extractive summaries in various ways. First, it can recall critical information from the document contexts, thereby increasing its informativeness. Second, it can compress unimportant/redundant information from the summaries, enhancing its conciseness. Last, it models summary discourse, therefore enhancing the readability. In addition, since the rewritten summaries are paraphrases of the extractive summaries, they tend to have fewer hallucinations than summaries generated from scratch using a pure abstractive

TABLE IV
Human Evaluation on Informativeness, Conciseness, Readability, and Faithfulness

Method	Info.	Conc.	Read.	Faith.
BERTSUMEXT	4.01	3.44	3.41	**5.00**
BERTSUMEXTABS	3.87	**3.73**	**4.06**	4.80
BERT-Rewriter	**4.12**	3.69	4.01	4.91
BART	4.19	3.42	4.27	4.89
BART-Rewriter	4.31	3.62	4.21	4.93
BART-JointSR	**4.32**	**4.15**	**4.59**	**4.97**

model. We confirm these hypotheses by conducting human evaluation. In particular, we randomly select 30 samples from the test set of CNN/DailyMail, scoring informativeness, conciseness, readability, and faithfulness. We assess the qualities with a number from 1(worst) to 5(best) by three independent annotators and report the average score.

The results are shown in Table IV. Compared to the extractive baseline BERTSUMEXT, our BERT-Rewriter improves the informativeness, conciseness, and readability by a significant margin while maintaining a high faithfulness. The improvement in informativeness mainly comes from document context, while the improvement in conciseness and readability is mainly contributed by reduced redundancy and improved coherence. Compared with the abstractive baseline BERTSUMEXTABS and BART, our rewriter BERT-Rewriter and BART-Rewriter improve faithfulness and informativeness while keeping the conciseness and readability close. Our joint rewriter BART-JointSR gives much higher scores than BART-Rewriter on conciseness and readability, suggesting the advantage of joint modeling of sentence selection and rewriting, which enhances the coherence between sentences.

IX. Analysis

To better understand where the performance improvement comes from, we further conduct quantitative studies on the ability of our rewriters to reduce redundancy, avoid irrelevance, and enhance coherence.

A. Redundancy

Redundancy has been a significant problem for both extractive and abstractive summarization [39], [40]. Previous work [5], [15], [35] adopts a simple strategy during beam-search to filter out paths with duplicated n-grams. In particular, 3-gram is used by most existing methods, and the strategy is known as Trigram Blocking [35].

As Fig. 9 shows, we compare two strategies for each model: one uses trigram-blocking during beam-search, and another does not. All the models except BART-JointSR give lower ROUGE scores when the trigram-blocking post-process is removed. The extractive model BERTSUMEXT depends on trigram-blocking the most; the ROUGE scores drop by 0.59 on average without it. The rewriting model BERT-Rewriter decreases the gap to 0.22 on average, suggesting that our rewriter can significantly reduce the redundant information in extractive summaries. Despite that trigram-blocking influences BART and BART-Rewriter for about 0.1 ROUGE point, our joint rewriter BART-JointSR

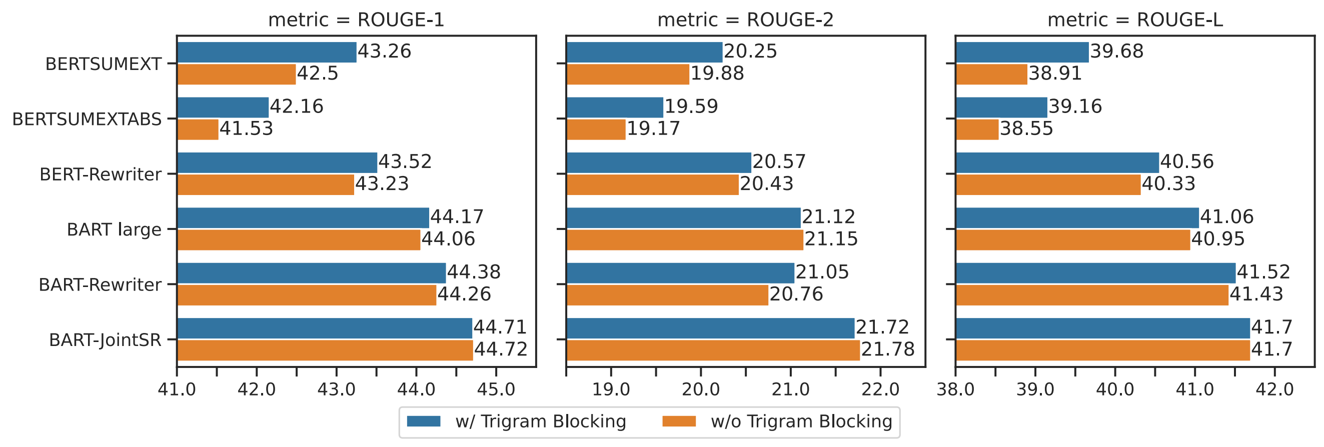

Fig. 9. Models have different abilities to generate non-redundant summaries. The more redundant information is generated by the model, the bigger performance drop will happen without trigram blocking.

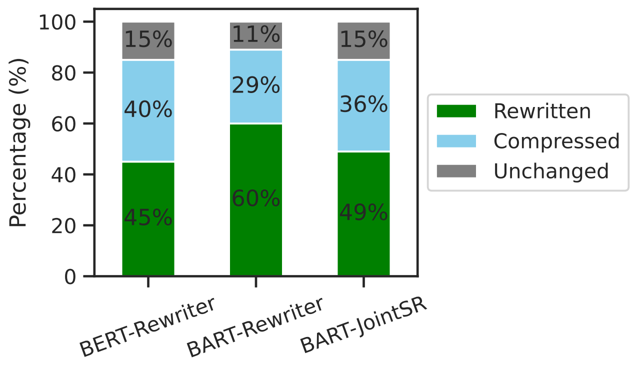

Fig. 10. Proportions of rewritten, compressed, and unchanged sentences after rewriting.

gives the same scores, indicating that the joint modeling of sentence selection and rewriting can fully reduce the redundant information.

B. Compression

Contextualized rewriters can remove unimportant phrases from extractive summaries. For instance, an extractive summary "*they returned to find hargreaves and the girl, who has not been named, lying on top of each other.*" is compressed into "*they returned to find hargreaves and the girl lying on top of each other.*" According to 20 samples of BERT/BART-Rewriter and BART-JointSR generations, all the compressions are on the phrase-level instead of on single words, and most removed phrases are likely unimportant since only 11% of them appear in gold summaries.

Our contextualized rewriters can largely compress the summaries. As the column #W in Table II shows, the oracle summaries are compressed to almost a half by our rewriters, while other extractive summaries are compressed more than one-third. As Fig. 10 shows, most extractive sentences are rewritten or compressed. We obtain these numbers on the test dataset by adopting the edit-sequence-generation algorithm [41] to generate a sequence of word editing actions that maps an extracted

Source Document: British world dressage champion Charlotte Dujardin won the Grand Prix at the World Cup in Las Vegas .① The 29-year-old , and her horse Valegro , who won the world title in Lyon last year , recorded a score of 85.414 percent to finish clear of Dutchman Edward Gal with American Steffen Peters in third .② Her score was short of the 87.129 she recorded in breaking her own world record last year , but there was no wiping the smile off Dujardin 's face ... The last few days , he was actually feeling not quite himself and I was a bit worried . ' But he was feeling much better and I had a really great ride . '

Rewritten Summary: Charlotte Dujardin won the Grand Prix at the World Cup in Las Vegas .① The 29-year-old , and her horse Valegro , won the world title in Lyon last year .②

Swap Group Tags: Charlotte Dujardin won the world title in Lyon last year .① The 29-year-old won the Grand Prix in Las Vegas .②

Fig. 11. Example of the ability to maintain coherence.

summary sentence to the rewritten one. We categorize a sentence as "Rewritten" if the sequence contains an action of adding or modifying, "Compressed" if it contains an action of deleting, and "Unchanged" otherwise. Take BART-JointSR as an example. Only 15% of extractive sentences remain unchanged during rewriting.

C. Coherence

Contextualized rewriters can improve extractive summaries on cohesion and coherence. It is partially illustrated by the higher ROUGE-L scores in the automatic evaluation and higher Readability scores in the human evaluation. Here, we use a case to demonstrate how our rewriter maintains the quality. In particular, the rewriting process of our rewriters is controlled by the extractive input. By altering the extractive input and comparing the rewriting outputs, we can qualitatively study the model's ability. As Fig. 11 shows, we use different sentence orders of extractive summary to test BART-Rewriter. We can see that the athlete's name is mentioned in the first summary sentence, while a nominator is used in the second sentence. When we change the order of the two extractive sentences, as the "Swap Group Tags" section shows, the content of the two summary sentences interchange their positions. However, the athlete's name is still presented in the first sentence, and a nominator is used in the second sentence. These observations demonstrate that our rewriter maintains the cross-sentence anaphora correctly.

X. Conclusion

We investigated contextualized rewriting for automatic summarization, building rewriters that make use of the original source document context, by proposing a general framework naming seq2seq with group-tag alignments and implement three rewriter instances. In addition to the standard rewriting schema, we also build a single joint model with an internal extractor. Results on standard benchmark show that contextual information is highly beneficial for summary rewriting, particularly when sentence selection and rewriting are jointly modeled. Our contextualized rewriters outperform existing non-contextualized rewriters by a significant margin, achieving strong ROUGE improvements upon multiple extractors for the first time. Our framework of seq2seq with group-tag alignments is general and can potentially be applied to other NLG tasks.

Acknowledgment

We appreciate the constructive comments from the anonymous reviewers. We thank Fangfang Su for the insightful discussions.

References

[1] M. Maybury, *Advances in Automatic Text Summarization*. Cambridge, MA, USA: MIT Press, 1999.
[2] O. Tas and F. Kiyani, "A survey automatic text summarization," *PressAcademia Procedia*, vol. 5, no. 1, pp. 205–213, 2007.
[3] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents," in *Proc. 31st AAAI Conf. Artif. Intell.*, S. P. Singh and S. Markovitch, Eds., 2017, pp. 3075–3081.
[4] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 1747–1759.
[5] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3730–3740. .
[6] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 379–389.
[7] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.
[8] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 93–98.
[9] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1998–2008..
[10] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 675–686.
[11] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4098–4109.
[12] B. Dorr, D. Zajic, and R. Schwartz, "Hedge trimmer: A parse-and-trim approach to headline generation," in *Proc. HLT-NAACL 03 Text Summarization Workshop*, 2003, pp. 1–8..
[13] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 484–494.
[14] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.
[15] S. Bae, T. Kim, J. Kim, and S.-G. Lee, "Summary level training of sentence rewriting for abstractive summarization," in *Proc. 2nd Workshop New Front. Summarization*, 2019, pp. 10–20.
[16] R. Wei, H. Huang, and Y. Gao, "Sharing pre-trained BERT decoder for a hybrid summarization," in *Proc. Chin. Comput. Linguistics - 18th China Nat. Conf.*, M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, Eds., 2019, pp. 169–180.
[17] L. Xiao, L. Wang, H. He, and Y. Jin, "Copy or rewrite: Hybrid summarization with hierarchical reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 9306–9313.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[19] M. Lewiset al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[20] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.
[21] G. Bao and Y. Zhang, "Contextualized rewriting for text summarization," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 12544–12553.
[22] L. Dong, J. Mallinson, S. Reddy, and M. Lapata, "Learning to paraphrase for question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 875–886. [Online]. Available: https://aclanthology.org/D17-1091
[23] A. Elgohary, D. Peskov, and J. Boyd-Graber, "Can you unpack that? Learning to rewrite questions-in-context," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*2019, pp. 5918–5924. [Online]. Available: https://aclanthology.org/D19-1605
[24] H. Suet al., "Improving multi-turn dialogue modelling with utterance ReWriter," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 22–31. [Online]. Available: https://aclanthology.org/P19-1003
[25] R. Anantha, S. Vakulenko, Z. Tu, S. Longpre, S. Pulman, and S. Chappidi, "Open-domain question answering goes conversational via question rewriting," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 520–534.
[26] S. Ren, Y. Wu, S. Liu, M. Zhou, and S. Ma, "A retrieve-and-rewrite initialization method for unsupervised machine translation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3498–3504.
[27] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 152–161. [Online]. Available: https://aclanthology.org/P18-1015
[28] A. Vaswaniet al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[29] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 132–141..
[30] C. Li, W. Xu, S. Li, and S. Gao, "Guiding generation for abstractive text summarization based on key information guide network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 55–60.
[31] I. Saito, K. Nishida, K. Nishida, and J. Tomita, "Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models," 2020, *arXiv:2003.13028*.

[32] Z.-Y. Dou, P. Liu, H. Hayashi, Z. Jiang, and G. Neubig, "GSUM: A general framework for guided neural abstractive summarization," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 4830–4842.
[33] S. Garg, S. Peitz, U. Nallasamy, and M. Paulik, "Jointly learning to align and translate with transformer models," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4453–4462.
[34] Y. Wuet al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016. [Online]. Available: https://arxiv.org/abs/1609.08144
[35] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," 2017, *arXiv:1705.04304*.
[36] K. M. Hermannet al., "Teaching machines to read and comprehend," in *Proc. 28th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.
[37] Y. Liu and P. Liu, "SimCLS: A simple framework for contrastive learning of abstractive summarization," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 1065–1072.
[38] Y. Liu, P. Liu, D. Radev, and G. Neubig, "Brio: Bringing order to abstractive summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2890–2903.
[39] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X.-J. Huang, "Extractive summarization as text matching," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6197–6208.
[40] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "A joint sentence scoring and selection framework for neural extractive document summarization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 671–681, 2020.
[41] F. Zhang and D. Litman, "Sentence-level rewriting detection," in *Proc. 9th Workshop Innov. NLP Building Educ. Appl.*, 2014, pp. 149–154.

Guangsheng Bao received the master's degree from the University of Science and Technology of China, Hefei, China, in 2008. He is currently working toward the Ph.D. degree with Zhejiang University, Hangzhou, China, and Westlake University, Hangzhou. His research interests include text summarization, machine translation, text generation, and coreference resolution.

Yue Zhang (Member, IEEE) is currently an Associate Professor with Westlake University, Hangzhou, China. His research interests include natural language processing and computational finance. He has been working on statistical parsing, natural language synthesis, machine translation, information extraction, sentiment analysis, and stock market analysis. He was the recipient of the Best Paper Awards of IALP 2017 and COLING 2018. He is on the Editorial Board of journals and conference, such as the Action Editor of the *Transaction of Association of Computational Linguistics*, and an Associate Editor for the *ACM Transactions on Asian and Low Resource Language Information Processing* and IEEE TRANSACTIONS ON BIG DATA, and as the Area Chair of the COLING 2014–2018, NAACL 2015, 2019, and 2020, EMNLP 2015, 2017, 2019, and 2020, ACL 2017–2020. He gave conference tutorials at NAACL 2010, ACL 2014, EMNLP 2016 and 2018.