



Auto-regressive extractive summarization with replacement

Tianyu Zhu¹ · Wen Hua¹ · Jianfeng Qu² · Saeid Hosseini³ · Xiaofang Zhou⁴

Received: 5 July 2021 / Revised: 20 January 2022 / Accepted: 21 September 2022 /

Published online: 9 December 2022

© The Author(s) 2022

Abstract

Auto-regressive extractive summarization approaches determine sentence extraction probability conditioning on previous decisions by maintaining a partial summary representation. Despite its popularity, the framework has two main drawbacks: 1) the partial summary representation is irresolutely denoted by a weighted summation of all the processed sentences without any filtering, resulting in a noisy representation and degrading the effectiveness of extracting subsequent sentences; 2) earlier sentences are biased towards a higher extraction probability due to the sequential nature of sequence tagging. To address these two problems, we propose the Auto-regressive Extractive Summarization with Replacement (AES-Rep), a novel auto-regressive extractive summarization model. In particular, the AES-Rep model consists of two main modules: the extraction decision module that determines whether a sentence should be extracted, and the replacement locator module that enables extracted deficient sentences to be replaced with latter sentences by comparing their expressiveness with respect to the main idea of the document. These modules update the partial summary with explicit actions using elaborated multidimensional guidance. We conduct extensive experiments on the benchmark CNN and DailyMail datasets. Experimental results show that AES-Rep can achieve better performance compared with various strong baselines in terms of multiple ROUGE metrics.

Keywords Extractive summarization · Auto-regressive model · Partial extraction discrepancy · Lead bias

1 Introduction

In recent years, the explosive growth of online textual data necessitates the evolution of document summarization systems, which aim at producing a shorter version of original documents, while preserving the main information. Moreover, since it can facilitate many widespread downstream applications, such as generating news digests, headlines, and automatically writing reports, many efforts have been invested in this task [1, 26].

Document summarization methods can be mainly divided into two categories: abstractive [27, 30, 32] and extractive [7, 9, 29]. In particular, abstractive approaches generate

✉ Wen Hua
w.hua@uq.edu.au

Extended author information available on the last page of the article

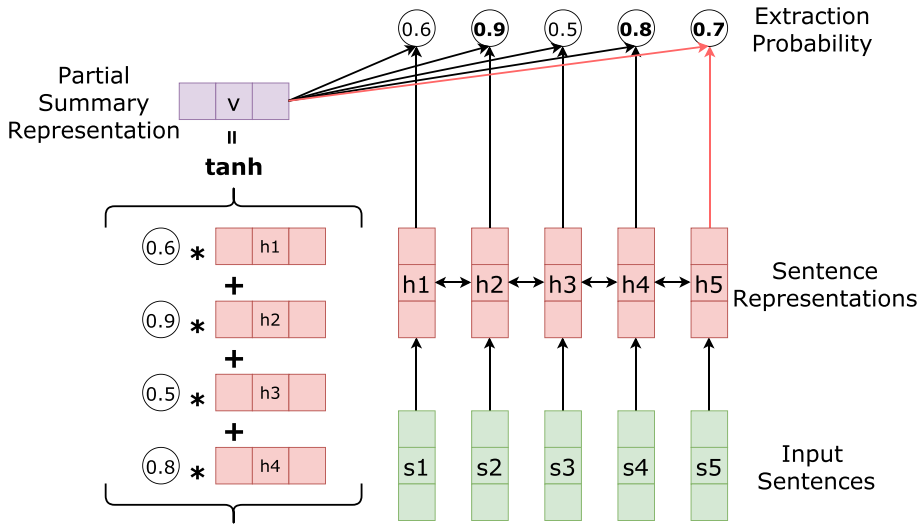


Figure 1 Illustration of the disadvantages in previous auto-regressive extraction models, e.g., SummaRuNer [28]

concise summaries by the techniques of paraphrasing and word replacing, while extractive approaches form summaries by means of identifying and concatenating salient text spans (e.g., sentences) from documents. Extractive methods are usually simpler and more computationally efficient than abstractive ones, and meanwhile, guarantee the syntactic and semantic correctness of the generated summaries [4]. Hence, we focus on extractive summarization in this paper. Moreover, there are two main types of extractive models: auto-regressive and non auto-regressive. Compared with non auto-regressive models [9, 29], auto-regressive extractive summarization [7, 28, 40] is believed to be a more reasonable strategy, which predicts the extraction label of the current sentence taking into account the labels of previously extracted sentences i.e., partial summary. Existing methods [7, 28] construct the partial summary representation through a weighted aggregation of previous sentence representations where the weights are given by their extraction probabilities (Figure 1 illustrates an example).

1.1 Challenges and contributions

An obvious discrepancy of the existing extractive summarization models is that sentence extraction is a straightforward Yes-or-No option, and there is no *partial* extraction. This discrepancy, referred to as *partial extraction discrepancy* hereinafter, constitutes a noisy representation of summaries, degrading the effectiveness of decisions on selecting subsequent sentences. For instance, as demonstrated in Figure 1, although sentences 1 and 3 will not be extracted in the final summary, their representations (noise) are still included into the partial summary representation, and consequently the estimated extraction probability of the subsequent sentences is affected. The fundamental cause of this problem is that the existing workflow is a ranking-based approach, which has to first finish predicting the extraction probabilities of all sentences, and thenceforth collects sentences with Top-K highest extraction probabilities as the summary. In other words, the model is agnostic of

which sentences would be extracted until all the sentences have been processed, thus infeasible to derive the unbiased partial summary representation.

Another disadvantage of the extractive methods is *lead bias* [9, 12, 24], referring that the output summary is mostly composed of the leading sentences. It is due to the sequential nature of the sequence labelling process, where leading sentences are exposed to the model first, and once they are extracted and updated into the partial summary representation, later sentences may be considered redundant and get rejected, regardless of whether these sentences would be a better choice. Consider the illustrative example provided in Figure 1 and Table 1, although sentences {3,4,5} compose a better summary ($\bar{R} = 55.38$), the extractive model may end up with a suboptimal summary {2,4,5} ($\bar{R} = 54.66$). Disregarding that sentence 3 is essentially a better substitution ($\bar{R} = 44.12$), sentence 2 is likely to be extracted due to its informativeness ($\bar{R} = 42.29$). Once sentence 2 is extracted, sentence 3 would be considered as highly duplicated (nearly identical with sentence 2) and get rejected, ending up with a suboptimal summary {2,4,5}.

In this paper, we introduce AES-Rep (Auto-regressive Extractive Summarization with Replacement), a novel auto-regressive extractive model that performs a series of summary update actions to constitute a summary. To address the first problem, unlike the widely-used ranking-based methods, we develop a classification setting to explicitly maintain a partial summary, which is straightforwardly updated using two actions: *IGNORE* the current sentence, or *ADD* it to the partial summary. If *IGNORE/ADD* is selected, the representation of current sentence will be completely excluded/included into the partial summary representation, respectively, preventing the error accumulation and propagation of sentences that would not be extracted. Actually, the requirement of an instant prediction for each sequence sentence poses a great challenge for an ordinary classifier. To achieve that, instead of using a single loss function, we craft attentive loss based on ROUGE distribution to optimize partial features (i.e., attentive document representation).

For the second problem, to realize a fair competition and alleviate the disadvantages brought by sentence position, we introduce the third action: *REPLACE* an extracted sentence with the current sentence, where an external *replacement locator module* is designed to further determine which sentence in the partial summary will be replaced by the current one, and update the partial summary accordingly. More specifically, we incorporate an introspective alignment between alternative sentence representations. This not only imbues our model with reasoning capabilities but enables a fine-grained comparison between

Table 1 Example of lead bias. \bar{R} is the averaged ROUGE-1/2/L F1 score

| Id | Sentence | \bar{R} |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| s2 | Mr Miliband will allow Scotland to set a more generous benefits system than the rest of the UK if he becomes Prime Minister. | 42.29 |
| s3 | Labour leader Ed Miliband will allow Scotland to set a more generous benefits system than the rest of the UK if he becomes Prime Minister. | 44.12 |
| s4 | Mr Miliband will hand Scottish MPs the unprecedented power to set a higher state pension and more generous dole and disability payments in a desperate attempt to reverse the exodus of his voters to the SNP. | 50.07 |
| s5 | He will unveil the radical proposals in his manifesto, due to be published tomorrow, as he attempts to fight back in Scotland – a key General Election battleground. | 48.86 |

Reference summary: Ed Miliband will allow Scotland to set a more generous benefits system. The move is a desperate attempt to reverse the exodus of voters to SNP. He will unveil the proposals in his manifesto, due to be published tomorrow

aligned representations. Additionally, we investigate the distribution of relative distance between valid replacements, and exploit distance information as an indispensable clue for the replacement locator module according to our statistical results. In this way, the final selected sentences are decided by the expressiveness of sentences themselves with respect to the main idea of the document rather than over-exploiting position advantages of the sentences.

Overall, our major contributions in this work are fourfold:

- For the first time, we investigate the problem of partial extraction discrepancy existed in auto-regressive extractive methods and give the fundamental cause of this problem.
- We propose a new extractive summarization framework, which can allocate instant explicit actions to the sequence of sentences in the document and constitute a clean partial summary to facilitate accurate actions on subsequent sentences during extraction.
- We design a replacement locator module. By leveraging introspective alignments and distance information, our model is able to reselect more crucial sentences for expressing the main idea of the document without the limitation of positions of sentences.
- We conduct extensive experiments on widely-used datasets, and the experimental results verify the superiority of our proposed model compared with various strong baselines.

The remaining of the paper is organized as follows: we introduce the details of our proposed model in Section 2 and report the experimental results in Section 3; the current literature of document summarization is discussed in Section 4, followed by a brief conclusion of the work in Section 5.

2 Methodology

In this section, we first provide some preliminary illustrating how we encode sentences with heterogeneous graph following prior works. We then describe the overall workflow of the proposed AES-Rep model. Lastly, we elaborate on how the main components, extraction decision module and replacement locator module work in detail.

2.1 Preliminary

Witnessing the success of applying heterogeneous graph into non auto-regressive summarization, we follow the work [38] to encode sentences.

There are two types of nodes in the heterogeneous graph, namely word nodes and sentence nodes. The sentence node is connected with word nodes contained in the sentence and the word node is connected with its composed sentence nodes. Formally, a document can be represented as a heterogeneous graph $G = \{V, E\}$. Here, the node set V is the union of word nodes $V_w = \{w_1, w_2, \dots, w_m\}$ and sentence nodes $V_s = \{s_1, s_2, \dots, s_n\}$, where m and n denote the number of unique words and sentences respectively. The edge set E contains all connected word-sentence node pairs (w_i, s_j) , representing the connectivity of the heterogeneous graph.

After constructing the heterogeneous graph, we initialize the graph by associating each node with a real-valued vector, which will be progressively updated and refined during the subsequent iterative update phase. The word nodes are initialized with pre-trained word embeddings. For sentences, we use CNN with various filter sizes to capture diverse local n-gram features and then apply LSTM over them to obtain global semantic features. Then, we

initialize sentence nodes by applying a linear transformation over the concatenation of local and global sentence features. H_w^0 and H_s^0 symbolize the initial representations of all word and sentence nodes respectively.

After initialization, the heterogeneous graph updates the node representations by iteratively passing messages between word and sentence nodes. Given the constructed graph G with initial node representations (H_w^0, H_s^0) and edge set E , we apply the Graph Attention Network (GAT) [36] to update semantic node representations. Formally, with h_i signifying the representation of the i -th (word or sentence) node, the GAT works as follows:

$$\begin{aligned} e_{ij} &= \text{LeakyReLU}(\vec{a}^T [Wh_i \| Wh_j]) \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{j' \in \mathcal{N}_i} \exp(e_{ij'})} \\ u_i &= \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} Wh_j \right) \end{aligned} \quad (1)$$

where $\|$ denotes concatenation, σ indicates activation function, \vec{a} , W are trainable parameters, α_{ij} is the attention weight between h_i and h_j and \mathcal{N}_i is the neighbor set of node i containing all j such that $(V_i, V_j) \in E$. The above vanilla attention is extended to multi-headed attention [35], where K independent attention mechanisms are performed and their outputs are concatenated:

$$u_i = \parallel_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k h_j \right) \quad (2)$$

where the superscript k indicates the attention weights α_{ij}^k and the transformation matrix W^k are from the k -th attention mechanism.

We further introduce residue connection [14] to avoid gradient vanishing and Position-wise Feed Forward Network [35] to enhance expressiveness. Then the word and sentence representations are updated in an iterative manner. Each iteration contains a sentence-to-word update and a word-to-sentence update. The t -th iteration can be denoted as follows:

$$\begin{aligned} U_{w \leftarrow s}^{t+1} &= \text{GAT}(H_w^t, H_s^t, H_s^t) \\ H_w^{t+1} &= \text{FFN}(U_{w \leftarrow s}^{t+1} + H_w^t) \\ U_{s \leftarrow w}^{t+1} &= \text{GAT}(H_s^t, H_w^{t+1}, H_w^{t+1}) \\ H_s^{t+1} &= \text{FFN}(U_{s \leftarrow w}^{t+1} + H_s^t) \end{aligned} \quad (3)$$

After T iterations, we collect the ultimate (sentence) node representations $H_s^T = [h_{s1}^T, h_{s2}^T, \dots, h_{sn}^T]$ as sentence representations. For brevity, hereinafter, we neglect the superscript T and subscript s (sentence node indicator), and reuse the symbol h_i to denote the representation of the i -th sentence.

2.2 Overall workflow

Figure 2 illustrates the detailed workflow of AES-Rep. After encoding the document, the *Extraction Decision Module* estimates the extraction affinity for each sentence

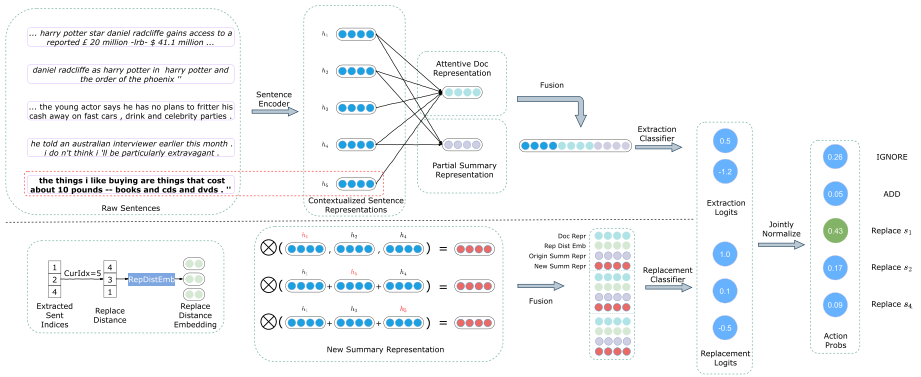


Figure 2 Model Overview. Extracted sentences (1, 2, 4) are shown in italics and the current sentence (5) is shown in bold. Sentence representations are generated by the Sentence Encoder, and other features are subsequently constructed. The extraction logits and the replacement logits are jointly normalized to produce a probability distribution over all the possible actions

based on sentence representation and other auxiliary features. Further, the *Replacement Locator Module* estimates the propensity of replacing each extracted sentence with the current sentence. Then, the raw extraction and replacement logits are jointly normalized to produce a distribution over all the actions, guiding the update of the summary. Eventually, sentences remained in the summary will serve as the output of the document.

Next, we provide formal definitions of the two tasks (i.e., sentence extraction and replacement) and attach a variable table (Table 2) to help readers understand and follow the paper.

Sentence extraction The *Extraction Decision Module* determines whether each sentence in the document should be extracted and added to the summary or not. Given the current sentence s_i , the extraction decision module returns a 2-dimensional vector $e \in \mathbb{R}^2$ indicating the confidence scores of ignoring and extracting the sentence,

Table 2 A variable table that clarifies the dimensions and meanings of the primary variables used in this section

| Variable | Dimension | Description |
|--------------|------------------------|--------------------------------------------------------------------------------------------------|
| $H_{w(s)}^0$ | $m(n) \times d_{w(s)}$ | initial representation of all word(sentence) nodes (m : # unique words, n : # sentences) |
| $H_{w(s)}^t$ | $m(n) \times d_{w(s)}$ | word(sentence) nodes representation from the t -th GAT layer |
| h_i | d_s | representation of the i -th sentence |
| d | d_s | attentive document representation |
| v | d_s | partial summary representation |
| v' | d_s | new summary representation after sent replacement |
| e_i | 2 | propensity of ignoring and extracting the i -th sentence |
| r_j | 1 | propensity of replacing the j -th extracted sentence with the current (i -th) sentence |

respectively. The confidence score would be higher if the corresponding action leads to a higher ROUGE score after updating the summary. More details will be introduced in Section 2.3.

Sentence replacement The *Replacement Locator Module* determines the propensity of replacing each extracted sentence with the current sentence. Given the current sentence s_i and the summary list containing indices of extracted sentences $S = \{c_1, c_2, \dots, c_k\}$, the replacement locator module returns a k -dimensional vector $r \in \mathbf{R}^k$, where r_j has a greater value if the resulting summary after replacing s_{c_j} with s_i ($\{s_{c_1}, \dots, s_{c_{j-1}}, s_{c_{j+1}}, \dots, s_{c_k}, s_i\}$) is of higher ROUGE score compared with other possible replacements. We will elaborate on the details in Section 2.4.

2.3 Extraction decision module

The Extraction Decision Module determines whether each sentence should be extracted and added to the summary. The extraction decision depends on not only the sentence itself but also the document representation and partial summary representation, to extract informative and non-redundant sentences.

Conventional average pooling [28] assumes uniform importance across sentences when constructing document representation, which may not be optimal. Attention mechanism, a technique that differentiate relevant part from others in the input, has achieved promising results in Machine Translation [3, 25] and Document Classification [42]. Considering the fact that sentences contribute differently to the semantics of the document, we apply attention mechanism to attribute higher weights to informative sentences when synthesizing document representation. The *attentive document representation* d is as follows:

$$\begin{aligned} u_i &= \tanh(W_{att}h_i + b_{att}) \\ \alpha_i &= \frac{\exp(u_i^\top u_{att})}{\sum_i \exp(u_i^\top u_{att})} \\ d &= \sum_i \alpha_i h_i \end{aligned} \quad (4)$$

Among the previous equations, α_i denotes the importance of the i -th sentence. $W_{att}, W_d, b_{att}, b_d$ and u_{att} are trainable parameters that would be optimized during model training. The *attentive document representation* is essentially a weighted combination of sentence representations using α_i as weights.

In general, our task is learning to assign appropriate action to each sentence such that the updated summary is of the greatest ROUGE score, where heuristically generated oracle action distribution is provided for supervision. Since the update actions explicitly manipulate the summary, it is feasible to maintain a summary list to track which sentences were extracted, and derive the partial summary representation solely based on extracted sentences, thus avoid the partial extraction discrepancy.

We obtain the partial summary representation v by summing up the sentence representations of extracted sentences and normalizing with the *tanh* function to keep the magnitude remains the same for all time-steps.

$$v = \tanh \left(\sum_{i=1}^k h_{c_i} \right) \quad (5)$$

The final sentence representation of the i -th sentence u_i for extraction decision is the concatenation of the document representation d , the partial summary representation v and the sentence representation h_i . Then u_i is fed into the extraction classification layer for a two-way classification.

$$\begin{aligned} u_i &= [d; v; h_i] \\ e_i &= W_{ext} u_i + b_{ext} \end{aligned} \quad (6)$$

where $W_{ext} \in \mathbf{R}^{2 \times d_{ext}}$, $b \in \mathbf{R}^2$ are trainable parameters.

2.4 Replacement locator module

Given the summary list $S = \{c_1, c_2, \dots, c_k\}$ containing indices of extracted sentences, the Replacement Locator Module determines the propensity of replacing each extracted sentence with the current sentence.

Formulation To model the propensity of replacement, we pair each extracted sentence s_{c_j} with the current sentence s_i for a binary classification, where the output r_j has a greater value if after replacing s_{c_j} with s_i , the resulting summary $\{s_{c_1}, \dots, s_{c_{j-1}}, s_{c_{j+1}}, \dots, s_{c_k}, s_i\}$ is of higher ROUGE score compared with other possible replacements.

Sentence pair representation To determine whether current sentence s_i is a good replacement for extracted sentence s_{c_j} , we construct the resourceful sentence pair representation sp with features that we believe are useful for the replacement classifier to make correct decisions.

First, we need to consider which candidate (s_{c_j} or s_i) is more relevant to the main point of the document, so we introduce the attentive document representation d into the sentence pair representation.

In addition to the document representation, we need to consider their relation to the remaining sentences in the partial summary (excluding s_{c_j}), that is, which sentence better complements the remaining sentences. Therefore, we construct two summary representations, namely the original summary representation v (identical to partial summary representation) and the new summary representation v' .

$$\begin{aligned} v &= \tanh \left(\sum_{i=1}^k h_{c_i} \right) \\ v' &= \tanh \left(h_i + \sum_{i=1, i \neq j}^k h_{c_i} \right) \end{aligned} \quad (7)$$

Moreover, we obtain the introspective alignment to capture the *interaction* between the two candidate summary: i) element-wise product that amplifies or dampens the matching signals between the two representations; ii) element-wise difference that measures the distance between the two representations.

$$\begin{aligned}
 prod &= v \odot v' \\
 diff &= v - v'
 \end{aligned}
 \tag{8}$$

Additionally, we check the distribution of the distance between two alternative sentences in the oracle labels, and observe that the probability of replacement decreases as the distance between the two sentences increases, as depicted in Figure 3. Therefore, we introduce the *replace distance embedding* $RepDist(i - c_j)$ to mitigate spurious long distance replacement.

Finally, we obtain the resourceful sentence pair representation sp by concatenating all these aforementioned features:

$$sp_{(c_j,i)} = [d; RepDist(i - c_j); v; v'; prod; diff]
 \tag{9}$$

sp is then fed into the final replacement classification layer to obtain the replacement propensity r_j .

$$r_j = w_{rep}sp + b_{rep}
 \tag{10}$$

where $w_{rep} \in R^{1 \times d_{rep}}$ and $b_{rep} \in R$ are trainable parameters.

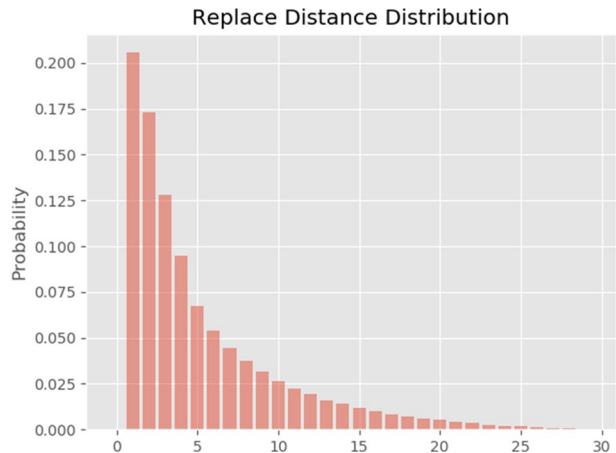
2.5 Loss functions

Attentive loss As the ROUGE score of individual sentence can be interpreted as a measure of sentence importance, we would like the attention scores to approximately match the sentence ROUGE score distribution. The ground truth ROUGE distribution is computed as:

$$P_{rouge}(i) = \frac{r(s_i, ref)}{\sum_{j=1} r(s_j, ref)}
 \tag{11}$$

where ref is the reference summary and r is a ROUGE based scoring function. The attentive loss L_{att} is the KL-Divergence between the attention scores and the ground truth distribution:

Figure 3 Replace distance distribution



$$L_{att}(\theta) = - \sum_{i=1}^n P_{rouge}(i) \log \left(\frac{\alpha_i}{P_{rouge}(i)} \right) \quad (12)$$

Action loss Given the current sentence s_t and the extracted sentence indices $S = \{c_1, c_2, \dots, c_k\}$, we concatenate the outputs of the Extraction Decision Module and the Replacement Locator Module to obtain the raw action logits z , and the extraction logits and the replacement logits are jointly normalized to produce a probability distribution over all the possible actions:

$$z = [e_1, e_2, r_1, r_2, \dots, r_k]$$

$$\hat{P}_i^{(t)} = \frac{\exp(z_i)}{\sum_{j=1}^{k+2} \exp(z_j)} \quad (13)$$

The action loss at timestep t is the KL-Divergence between the action probabilities $\hat{P}^{(t)}$ and the ground truth action distribution $P^{(t)}$ (We will discuss how to generate the ground truth distribution P in Section 3.1) and the action loss for the entire document is the averaged loss across all the timesteps:

$$L_{act}^{(t)}(\theta) = - \sum_{i=1}^{k+2} P_i^{(t)} \log \left(\frac{\hat{P}_i^{(t)}}{P_i^{(t)}} \right) \quad (14)$$

$$L_{act}(\theta) = \frac{1}{n} \sum_{i=1}^n L_{act}^{(t)}(\theta)$$

The final loss of the AES-Rep is the weighted combination of the two losses with a hyper parameter λ controlling the relative contribution of attentive loss:

$$L(\theta) = L_{act}(\theta) + \lambda L_{att}(\theta) \quad (15)$$

In this way, our model is able to consider multiple evidence and finally achieve the global optimal solution.

3 Experiments

We have conducted extensive experiments on the most commonly used datasets to evaluate the performance of our proposed AES-Rep model, and the experimental results are reported in this section.

3.1 Experimental setup

Datasets We evaluate our model on the widely-used CNN/DailyMail¹ dataset and the separated CNN and DailyMail datasets, which contain news articles and their highlights (used as abstractive reference summary). Following previous work [6, 32], we adopt the standard split for the train, validation, test set and obtain the tokenized, non-anonymized dataset by pre-processing. Moreover, we also conduct experiments on WikiHow dataset² [18]. WikiHow is

¹ Available at <https://cs.nyu.edu/~kcho/DMQA/>

² Available at <https://github.com/mahnazkoupae/WikiHow-Dataset>

a large-scale summarization dataset extracted and constructed from an online knowledge base written by different human authors. The articles cover a wide range of topics and represent high diversity styles. We show the statistics of the datasets in use in Table 3.

Evaluation metrics We employ ROUGE [20] as the evaluation metric to measure how the model summary resembles the reference summary by counting the number of overlapping lexical units like n-grams and word sequences. Following the common practice, we report ROUGE-1, ROUGE-2, and ROUGE-L F1 results balancing the precision and the recall, where ROUGE-1 and ROUGE-2 measure informativeness via counting overlapping n-grams and ROUGE-L measures fluency through the longest common subsequence. We leverage the average of all ROUGE F1 variants as the scoring function r :

$$r(\text{sum}, \text{ref}) = \frac{1}{3}(\text{ROUGE-1}(\text{sum}, \text{ref}) + \text{ROUGE-2}(\text{sum}, \text{ref}) + \text{ROUGE-L}(\text{sum}, \text{ref})) \quad (16)$$

We also estimate statistical significance by running our model with different random seeds and performing the t-test between our results and the best baseline performance. We compare p -value with 0.05 and 0.01, and highlight “significant” improvement achieved by our model via * or ** respectively in the following tables.

Model settings We limit the size of vocabulary to 50,000 and initialize the embeddings with 300-dimensional GloVe [31] word vectors. The filter sizes of CNN for extracting local features range from 2 to 7 with 50 feature maps each, and the LSTM for capturing global features is a 2-layer bidirectional LSTM with hidden size 128 in each direction. Following [38], we skip stopwords and 10% words with low TF-IDF values when constructing word nodes. The dimension of word and sentence node representation is set to 300 and 384 respectively. The GAT has 4 attention heads for word nodes and 6 attention heads for sentence nodes. The intermediate hidden size of FFN layer is set to 1536. The number of iterations T is set to 2. The size of replacement distance embedding is set to 384 as well. The weighting factors λ in (15) is set to 0.1. The temperature τ in (17) is set to 0.05 and 0.01 on CNN/DailyMail and WikiHow respectively. To regularize the model, we apply Dropout [34] with probability 0.1 to the output of the first LSTM layer, GAT inputs, GAT attention weights and the intermediate output of FFN. The model is trained with Adam [17] optimizer with batch size 64. For the hyperparameters of Adam, we set the learning rate $lr = 0.0005$, the two momentum coefficients $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$, respectively. Furthermore, we employ gradient norm clipping to rescale the norm to at most 2.0. We train the model for 20 epochs and select the checkpoint based on the averaged ROUGE score and report the evaluation results on the test set.

Table 3 Statistics of summarization datasets: the size of train, valid, test splits and average length of documents and summaries (in terms of word and sentence) are reported

| Dataset | # docs (train/valid/test) | avg. doc. len | | avg. summ. len | |
|-----------|---------------------------|---------------|---------|----------------|---------|
| | | # words | # sents | # words | # sents |
| CNN | 90125/1220/1093 | 756.26 | 32.45 | 45.46 | 3.56 |
| DailyMail | 196959/12147/10396 | 774.86 | 36.60 | 53.68 | 3.83 |
| CNNDM | 287084/13367/11489 | 769.35 | 35.37 | 51.24 | 3.75 |
| WIKIHOW | 168123/6000/6000 | 582.18 | 29.71 | 62.21 | 7.58 |

The length statistics are calculated with respect to the entire dataset

Ground truth generation Since our model is based on a novel setting, there are none handy annotated labels for the training data. To get rid of this, we utilize a greedy approach to construct oracle labels, which is based on the intuition that the action that incurs more ROUGE gain concerning the reference should have a higher probability. Algorithm 1 depicts the details on how we generate the ground truth action distributions based on the human-written abstractive summaries.

For the ground truth action distribution at timestep t , given current sentence s_t and summary $S = \{s_{c_1}, s_{c_2}, \dots, s_{c_k}\}$ containing k sentences, the $gains = [g_1, g_2, \dots, g_{k+2}]$ (lines 6–14) is an array with length $k + 2$ containing the ROUGE score gains of each action (ignore, add, and replace each extracted sentence with the current sentence). The *normalize* function (line 28) is defined as follows to produce a valid probability distribution P :

Input: document $D = \{s_1, \dots, s_n\}$; reference summary ref ; temperature τ
Output: action distributions $distrs$

```

1  $distrs = []$ ;
2  $S_{id} = \{\}$ ;
3  $S = \{\}$ ;
4  $base = 0$ ;
5 for  $t = 1, 2, \dots, n$  do
6    $ignore = 0$ ;
7    $g_{add} = r(S \cup \{s_t\}, ref) - base$ ;
8    $gains = [ignore, g_{add}]$ ;
9    $g_{replace} = -1$ ;
10   $best\_rep = -1$ ;
11   $best\_rep\_idx = -1$ ;
12  for  $id$  in  $S_{id}$  do
13     $g = r(S \setminus s_{id} \cup \{s_t\}, ref) - base$ ;
14     $gains.append(g)$ ;
15    if  $g > g_{replace}$  then
16       $g_{replace} = g$ ;
17       $best\_rep = s_{id}$ ;
18       $best\_rep\_idx = id$ ;
19    end
20  if  $g_{add}$  is the max then
21     $base = base + g_{add}$ ;
22     $S = S \cup \{s_t\}$ ;
23     $S_{id} = S_{id} \cup \{t\}$ ;
24  if  $g_{replace}$  is the max then
25     $base = base + g_{replace}$ ;
26     $S = S \setminus best\_rep \cup \{s_t\}$ ;
27     $S_{id} = S_{id} \setminus best\_rep\_id \cup \{t\}$ ;
28   $distr = normalize(gains, \tau)$ ;
29   $distrs.append(distr)$ ;
30 end
31 return  $distrs$ ;

```

Algorithm 1 Greedy approach to generate ground truth action distributions.

$$P_i = \frac{\exp(g_i/\tau)}{\sum_{j=1}^{k+2} \exp(g_j/\tau)}, i = 1, 2, \dots, k + 2 \quad (17)$$

where τ is a hyperparameter controlling the smoothness of the distribution. After obtaining the action probabilities, the summary is updated by applying the action with the maximum probability (lines 20–27).

Sequentially updating the summary with the current best action in a greedy manner may lead to local optimum, producing suboptimal summary thereby degrading the effectiveness of the downstream summarization model. For this concern, we investigate theoretically and empirically to conclude that the greedy heuristics has minor effect on downstream summarization task. On one hand, finding a globally optimal extraction oracle is computationally expensive. As an approximation, greedy approach has been widely adopted by various competitive systems in generating oracle extraction labels. For example, previous studies [28, 29, 40] maintain extraction oracle by incrementally adding a sentence at a time to maximize its ROUGE, until none of the remaining sentences improve the ROUGE score of the oracle. On the other hand, we include the extraction oracle in Tables 4 and 5 to provide readers a sense of the performance upperbound. Notice that the oracle is ahead of both our model and other competitive

Table 4 Full length ROUGE F1 evaluation(%) on the combined CNN/Daily Mail test set

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | ROUGE-Mean |
|------------------------------|----------------|----------------|---------------|----------------|
| LEAD-3 | 40.43 | 17.62 | 36.66 | 31.57 |
| <i>ORACLE</i> | 57.50 | 33.63 | 53.99 | 48.37 |
| NN-SE | 35.50 | 14.70 | 32.20 | 27.46 |
| SummaRuNNer | 39.60 | 16.20 | 35.30 | 30.37 |
| RNES | 41.25 | 18.87 | 37.75 | 32.62 |
| NEUSUM | 41.59 | 19.01 | 37.98 | 32.86 |
| REFRESH | 40.00 | 18.20 | 36.60 | 31.60 |
| LATENT | 41.05 | 18.77 | 37.54 | 32.45 |
| SUMO | 41.00 | 18.40 | 37.20 | 32.20 |
| HER | 42.30 | 18.90 | 37.90 | 33.03 |
| PACSUM [†] | 40.70 | 17.80 | 36.90 | 31.80 |
| Pointer+BERT [†] | 42.39 | 19.51 | 38.69 | 33.58 |
| Pointer+BERT+RL [†] | 42.69 | 19.60 | 38.85 | 33.71 |
| BERT-ext [†] | 42.29 | 19.38 | 38.63 | 33.43 |
| HSG | 42.31 | 19.51 | 38.74 | 33.52 |
| HSG+Tri-Blocking | <u>42.95</u> | <u>19.76</u> | <u>39.23</u> | <u>33.98</u> |
| PGN | 39.53 | 17.28 | 36.38 | 31.06 |
| DRM | 41.16 | 15.82 | 39.08 | 32.02 |
| BottomUp | 41.22 | 18.68 | 38.34 | 32.75 |
| DCA | 41.69 | 19.47 | 37.92 | 33.31 |
| BERTSumAbs [†] | 41.72 | 19.39 | 38.76 | 33.29 |
| BERTSumExtAbs [†] | 42.13 | 19.60 | 39.18 | 33.64 |
| AES-REP | 43.21** | 19.90** | 39.38* | 34.16** |
| <i>p</i> -value | 5.8e-9 | 4.3e-6 | 0.0203 | 7.1e-7 |

* and ** denote the statistical significance for $p \leq 0.05$ and $p \leq 0.01$, respectively, compared to the best baseline result (marked with underline). [†] indicates the baseline is powered by Pre-trained Language Model

Table 5 Full-length ROUGE F1 on the separated CNN and the Daily Mail test set

| Model | CNN | | | DailyMail | | |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | 1 | 2 | L | 1 | 2 | L |
| LEAD-3 | 28.80 | 11.00 | 25.50 | 41.20 | 18.20 | 37.30 |
| <i>ORACLE</i> | 50.30 | 27.97 | 46.52 | 58.26 | 34.22 | 54.78 |
| NNSE | 28.40 | 10.00 | 25.00 | 36.20 | 15.20 | 32.90 |
| REFRESH | 30.40 | 11.70 | 26.90 | 41.00 | 18.80 | 37.70 |
| SUMO(1layer) | 29.50 | 11.60 | 26.20 | 41.60 | 18.80 | 37.60 |
| SUMO(3layer) | 29.70 | <u>12.00</u> | 26.50 | 42.00 | <u>19.10</u> | 38.00 |
| HER | <u>30.70</u> | 11.50 | <u>27.50</u> | <u>42.70</u> | 19.00 | <u>38.50</u> |
| AES-Rep | 32.66** | 12.88** | 28.52** | 43.85** | 20.11** | 39.94** |
| <i>p</i> -value | 4.4e-6 | 3.1e-4 | 1.1e-4 | 4.6e-7 | 4.6e-7 | 7.9e-7 |

* and ** denote the statistical significance for $p \leq 0.05$ and $p \leq 0.01$, respectively, compared to the best baseline result (marked with underline)

baselines by a considerable margin, thus the performance drop caused by greedy heuristic is a minor concern as the model performance is still far from the upperbound.

Baselines We compare our proposed model AES-Rep with various baselines: **LEAD-3** is a commonly-used baseline method that simply selects the first three sentences in the document as the summary. **NN-SE** [7] and **SummaRuNNer** [28] are two cross-entropy based auto-regressive extractive methods, where extraction-probability-weighted sentence representations are used to construct the partial summary representation. **RNES** [40] is yet another auto-regressive extractive model that combines the cross-sentence coherence and the ROUGE score of the extraction as the reward signal to obtain informative and coherent summaries. **NEUSUM** [48] jointly learns to score and extract sentences in an auto-regressive manner. **REFRESH** [29] is an extractive model that treats document summarization as a sentence ranking problem and uses reinforcement learning to globally optimize the ROUGE score. **LATENT** [43] is a latent variable extractive summarization model that leverages human summaries directly with the help of a sentence compression model. **SUMO** [23] conceptualizes single document summarization as a tree induction problem. **HER** [24] is a non auto-regressive method imitating how humans extract summaries, which formulates the learning process as a contextual bandit and solves it with policy gradient reinforcement learning. **PACSUM** [45] is an unsupervised graph-ranking-based summarization system that uses BERT to capture sentence similarity. **Pointer+BERT** [46] uses a feature-based BERT (without gradient) as encoder to get token embeddings and employs Pointer Network [37] as decoder to pick summary sentences, **Pointer+BERT+RL** [46] introduces reinforcement learning to further optimize the model. **BERT-ext** [2] is yet another architecture based on BERT and Pointer Network, but BERT is utilized to obtain sentence representations directly. **HSG** [38] constructs heterogeneous graphs by introducing semantic nodes of different granularities, thereby enhancing the model's capability to learn cross-sentence relations. **HSG+Tri-Blocking** introduces Trigram Blocking [21] to reduce redundancy in the output summary. For the abstractive models, **PNG** [32] is capable of generating out-of-vocabulary words by directly copying them from the input document.

DRM [30] is trained with a combined loss of supervised learning and policy gradient to mitigate exposure bias and generate readable summaries. **BottomUp** [10] designs a content selector to determine phrases in a source document that should be part of the summary, and then use this selector as a bottom-up attention step to constrain the model to focus on likely phrases. **DCA** [5] address the challenge of encoding a long document by introducing multiple collaborating agents, each of which in charge of a subsection of the input text. **BERTSumAbs** [21] and **BERTSumExtAbs** [21] are two abstractive models based on BERT where the former adopts the default abstractive training protocol while the latter pretrains the encoder with extractive objectives before abstractive training.

3.2 Results and analysis

In this section, we first report the overall results of quantitative evaluation using ROUGE metrics, and then perform an ablation study to examine the effectiveness of each module in the proposed model. Lastly, we do a case study to showcase the decision process of AES-Rep with specific examples.

Overall performance We present the results of AES-Rep together with other selected baselines on the CNN/DailyMail dataset in Table 4. The table is divided into 5 blocks, which respectively report the results of unsupervised baselines (and oracle), auto-regressive extractive baselines, non auto-regressive extractive baselines, abstractive baselines and our model.

When comparing with the unsupervised baselines, our model performs better by a considerable margin. In particular, LEAD-3 only considers the importance of sentence positions in a document and simply uses the first three sentences as the summary. Our AES-Rep model achieves a large increase of the ROUGE scores (2.78%/2.28%/2.72%) over LEAD-3 by allowing later sentences (which might be more topically important) to be added to the summary.

Compared with auto-regressive baselines in Table 4, we observe that AES-Rep surpasses all these models in terms of all ROUGE metrics. Specifically, our model achieves a substantial improvement of (7.71%/5.20%/7.18%) and (3.61%/3.70%/4.08%) over NN-SE and SummaRuNNer concerning ROUGE-1/2/L respectively. We attribute the success to the fix of partial extraction discrepancy and the introduction of the replacement locator module. Our model also shows better performance than RNES and NEUSUM.

For non auto-regressive extractive baselines (mainstream summarization), even HSG+Tri-Blocking which is the state-of-the-art non auto-regressive model (non-BERT-based), AES-Rep demonstrates its performance superiority with significant improvement whose p -value < 0.05 . Note that the reported results are produced by directly evaluating our model without involving any post-processing (e.g. trigram blocking). If we compare AES-Rep with plain HSG without post-processing, the performance gap would grow wider.

Finally, AES-Rep outperforms all the selected abstractive baselines as shown in Table 4. It is worth mentioning that our model surpasses a few BERT-based models, where both extractive and abstractive baselines are included. As a backbone, BERT is pre-trained on enormous corpora containing more than 3300 million words. In contrast, our model is exclusively trained on the summarization dataset, which is much more efficient.

We also conduct experiments on the separated CNN and DailyMail dataset and report separate results in Table 5. For the baselines, we select those that have conducted experiments on the separated dataset and report their results. Note that we do not elaborately tune

hyperparameters per dataset, instead, we reuse the hyperparameters reported in Section 3.1 to examine the versatility of these hyperparameters. As shown in Table 5, AES-Rep consistently outperforms all the competitive baselines on both datasets, where the improvement on each ROUGE metric is quite significant with the p -value < 0.01 .

For the out-of-domain evaluation, we report the experimental results in Table 6. As can be observed from the table, the advantages of AES-Rep over other baselines gets smaller compared with that of CNNDM dataset, mainly due to the switch of domains. AES-Rep fails to obtain a comparable ROUGE-2 score compared with PGN w/ Coverage, since the higher level of abstraction of the dataset makes abstractive methods have advantage over extractive methods. However, AES-Rep still achieves better ROUGE-1 and ROUGE-L scores over extractive baselines, with statistical significance p -value less than 0.01 and 0.05 respectively.

Ablation study We conduct the ablation study by removing each module of the proposed AES-Rep and observing its effect on the model performance. Firstly, to examine the effectiveness of the sentence replacement mechanism, we deactivate the *REPLACE* operation during the oracle generation and training stage and predict IGNORE/ADD action for each sentence (*w/o REPLACE*). Secondly, we keep using the attentive document representation but exclude the attentive loss from the training objectives (*w/o AttLoss*). Thirdly, we replace the attentive document representation with conventional max/average pooled document representation (also ignore the attentive loss L_{att}) and name them as (*w/o AttPool (+MaxPool)*) and (*w/o AttPool (+AvgPool)*), respectively. Finally, we exclude the replacement distance embedding from the replacement classification to examine how much the replacement distance feature contributes to correct classification (*w/o RepDistEmbedding*). The results of the ablation study on CNN/DailyMail and WikiHow datasets are presented in Tables 7 and 8 respectively.

Given the results, we have the following observations: (1) The replacement operation is indispensable under our settings, and disabling REPLACE action results in a significant performance degradation on both CNN/DailyMail and WikiHow datasets. Considering the strategy of assigning appropriate IGNORE/ADD action to maximize the ROUGE score

Table 6 Full-length ROUGE F1 on the WikiHow test set

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | ROUGE-Mean |
|----------------------|----------------|-------------|---------------|--------------|
| LEAD-Doc | 24.46 | 5.56 | 22.62 | 17.54 |
| LEAD-Para | 22.04 | 6.27 | 20.87 | 16.39 |
| TextRank | 27.53 | 7.4 | 20.00 | 18.31 |
| <i>Oracle</i> | 40.84 | 14.40 | 38.15 | 31.13 |
| NNSE | 28.55 | 7.90 | 26.60 | 21.02 |
| SummaRuNNer | <u>28.93</u> | 8.01 | <u>26.97</u> | 21.30 |
| Seq2seq w/ Attention | 22.04 | 6.27 | 20.87 | 16.39 |
| PGN | 27.30 | 9.10 | 25.65 | 20.68 |
| PGN w/ Coverage | 28.53 | <u>9.23</u> | 26.54 | <u>21.43</u> |
| AES-Rep | 29.46** | 7.75 | 27.23* | 21.48 |
| p -value | 0.0005 | 9.9e-7 | 0.0112 | 0.3788 |

* and ** denote the statistical significance for $p \leq 0.05$ and $p \leq 0.01$, respectively, compared to the best baseline result (marked with underline)

Table 7 Ablation studies on the combined CNN/Daily Mail test set

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | ROUGE-Mean |
|-----------------------------|--------------|--------------|--------------|--------------|
| AES-Rep | 43.21 | 19.90 | 39.38 | 34.16 |
| – <i>REPLACE</i> | 41.25 | 18.05 | 37.32 | 32.20 |
| – <i>AttLoss</i> | 42.99 | 19.65 | 39.05 | 33.89 |
| – <i>AttPool (+MaxPool)</i> | 43.08 | 19.78 | 39.21 | 34.02 |
| – <i>AttPool (+AvgPool)</i> | 43.03 | 19.70 | 38.05 | 33.59 |
| – <i>RepDistEmbedding</i> | 43.13 | 19.79 | 39.20 | 34.04 |

We remove various modules one by one to examine their contribution to the full model. ‘-’ means removing the component from the full AES-Rep model

Table 8 Ablation studies on the WikiHow test set

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | ROUGE-Mean |
|-----------------------------|--------------|-------------|--------------|--------------|
| AES-Rep | 29.46 | 7.75 | 27.23 | 21.48 |
| – <i>REPLACE</i> | 25.47 | 5.93 | 23.44 | 18.28 |
| – <i>AttLoss</i> | 28.99 | 7.46 | 26.82 | 21.09 |
| – <i>AttPool (+MaxPool)</i> | 28.79 | 7.46 | 26.57 | 20.94 |
| – <i>AttPool (+AvgPool)</i> | 28.50 | 7.31 | 26.30 | 20.70 |
| – <i>RepDistEmbedding</i> | 29.22 | 7.67 | 27.02 | 21.30 |

We remove various modules one by one to examine their contribution to the full model. ‘-’ means removing the component from the full AES-Rep model

of the resulting summary at each timestep, the summary will be instantly filled up with leading sentences, causing catastrophic performance drop especially when good sentences locate at the beginning of the document but there are better substitutions among subsequent sentences. (2) Plain attentive pooling works well on WikiHow dataset but fails to outperform conventional max pooling and average pooling on CNN/DailyMail dataset. However, once sentence-level ROUGE score distribution is introduced to guide the attention weight distribution, attentive pooling can consistently surpass conventional pooling methods on both datasets. (3) Replacement distance embedding provides evidence for replacement locator module from a different perspective, and removing it brings minor performance decline on both datasets.

Parameter sensitivity analysis We study the robustness of AES-Rep by investigating the performance fluctuations with varied hyperparameters. Specifically, we study the sensitivity of our model to temperature τ in (17) and weighting factor λ in (15). Based on the hyperparameter setup reported in Section 3.1, we conduct standard *one-factor-at-a-time* analysis by varying the value of one hyperparameter while keeping others at their baseline values, and report the new summarization performance achieved. Similar to Table 4, ROUGE-1/2/L and ROUGE-Mean \bar{R} are adopted for evaluation.

Impact of τ The temperature τ is introduced in (17) to control the smoothness of the ground truth action distribution. As can be seen in Figure 4(a), AES-Rep achieves

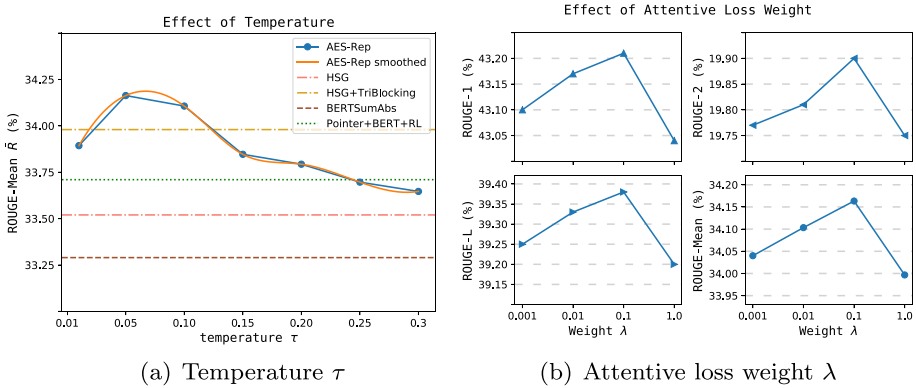


Figure 4 Parameter sensitivity analysis on CNN/DailyMail dataset

the best performance (outperforms HSG+TriBlocking) on CNN/DailyMail given $\tau \in [0.05, 0.1]$, while still achieves comparable performance than other baselines when τ ranges broadly from 0.01 to 0.30. Meanwhile, Figure 5(a) illustrates the AES-Rep achieves its peak on WikiHow when τ is around $[0.010, 0.012]$. In general, AES-Rep benefits from a moderate-ranged τ (the range is dataset dependent, thus requires some tuning), and decreasing or increasing τ hurts performance. We conjecture the reasons are: 1) tiny τ produces nearly one-hot labels. Models trained with these labels fail to distinguish between better and worse non-optimal actions, as they are both labeled as negative. 2) as the magnitude of ROUGE gains (raw logits of softmax) has a magnitude of $0.0x-0.x$, the ground truth distribution produced by large τ nearly degenerates to uniform distribution that provides very little or almost no supervision to the model.

Impact of λ The weighting factor λ is introduced in (15) to control the relative contribution of *Attentive Loss*. We study the impact of $\lambda \in \{0.001, 0.01, 0.1, 1.0\}$. Combining Figures 4(b) and 5(b), the attentive loss weight λ is more robust across datasets, showing a similar trend for all evaluation metrics when increasing from 0.001 to 1.0. Specifically, our model consistently benefits from a relatively larger λ , but when

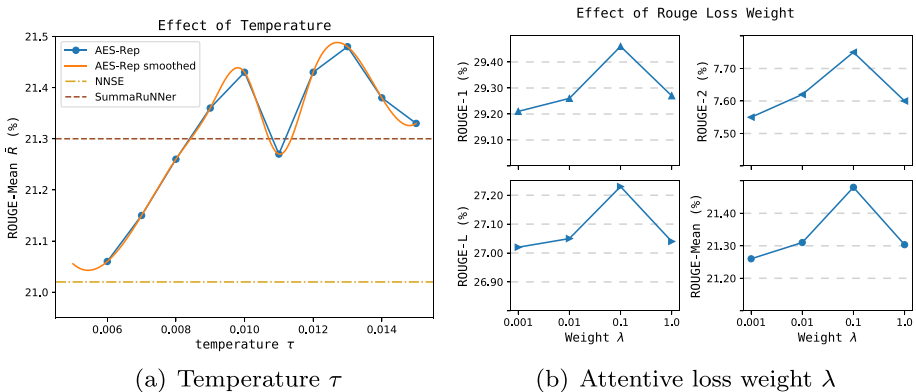


Figure 5 Parameter sensitivity analysis on WikiHow dataset

λ reaches a certain magnitude (1.0 in our case), the improvement tends to stop. This is quite intuitive: on one hand, for tiny λ , the regularization of attention distribution is not sufficiently regularized thus the learning process does not benefit from this auxiliary task. On the other hand, large λ dominates the supervision signal, hindering the model in effectively selecting appropriate actions for sentences. As can be seen from the results, $\lambda = 0.1$ seems to be a good trade-off for attention distribution regularization and effective action classification, where the summarization performance reaches its peak.

Case study Table 9 shows an example of AES-Rep predictions with additional columns to better illustrate the predicted actions. For conciseness, we only list the first a few sentences, with an unimportant sentence 4 skipped and some irrelevant verbose descriptions replaced by ellipses.

The selected document is a news article about a woman posing as a social worker, stabbing a new mother, and kidnapping a baby as her own child, charged with murder and kidnapping a new mother. For the three leading sentences: sentence 0 illustrates that the victim lives with her newborn daughter. Sentence 1 claims how the criminal deceived the victim's boyfriend. Sentence 2 is a supplementary explanation of sentence 1 that the criminal never worked for child-welfare and her identity is faked. The three sentences describe the background of the crime from different aspects with minor duplication between each other. It is worth noting that during training, the model learns to assign the optimal action that leads to the greatest ROUGE scores to each sentence. The leading sentences contain the names of the criminal and the victim and how the criminal defraud the victim's family, which are demonstrated in the reference summary as well, therefore all three leading sentences were extracted. For sentences that received REPLACE action: sentence 5 mentioned the name of the criminal and victim, and listed the charges of the criminal in detail. Compared with Sentence 1, Sentence 6 is more verbose. Although they all mentioned fake identities, Sentence 6 mentioned the baby's name and the follow-up after stealing the baby. Sentence 7 mentioned that the victim's body was found in the closet of the criminal's home. Although the sentence itself did not share too many common words with the reference summary, it still provides crucial information and is worth being extracted into the summary. Sentences with action IGNORE are either totally irrelevant (sentence 4) or less salient compared with selected sentences (other sentences). From this example, we can see that the action selection module distinguishes salient sentences from irrelevant sentences and assigns appropriate actions (ADD/REPLACE and IGNORE) to them, meanwhile, the replacement locator module can correctly locate less informative candidate sentence from the current summary list and replace it with the current sentence. In conclusion, the entire AES-Rep model exhibits its functionality as expected.

4 Related work

In this section, we review the current literature of document summarization in three categories: extractive models, abstractive models, and combined models.

Extractive approaches Extractive summarization aims to identify salient sentences and concatenate them to compose the summary. It usually treats summarization as a sequence labelling task, where the model eventually assigns a binary label to each sentence, indicating its inclusion/exclusion in the output.

Table 9 Example of AES-Rep predictions

| Sentence | Action | Sum | \bar{R} |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-----------|-----------|
| 0: samantha fleming was asleep in the indiana apartment she shared with her boyfriend and newborn daughter when the doorbell rang on april 6 | Add | [0] | 12.88 |
| 1: .. a professionally dressed woman carrying a binder who introduced herself to fleming 's boyfriend as a department of child services worker .. | Add | [0, 1] | 22.39 |
| 2: but authorities say the woman , geraldine r. jones , never worked for the child-welfare agency and that her visit to fleming 's apartment in anderson , about 30 miles north of indianapolis , was part of an elaborate scheme to take the 17-day-old baby and claim the girl as her own | Add | [0, 1, 2] | 25.73 |
| 3: why jones targeted fleming, and why the young mother and her daughter, serenity, left the apartment with her, were key questions in a mystery police continued trying to unravel ... | Ignore | [0, 1, 2] | 25.73 |
| 5: geraldine jones , 36 , of gary , indiana, has been charged with .. murder, kidnapping and criminal confinement over the death of new mom samantha fleming | Repl.0 | [1, 2, 5] | 31.43 |
| 6: investigators say jones pretended to be a social worker in order to steal fleming 's daughter, serenity, and claim the child as her own after faking a pregnancy | Repl.1 | [2, 5, 6] | 34.67 |
| 7: fleming 's body was found by police at the home belonging to jones in in gary , indiana . the body had been doused with bleach , put in a garbage bag that was duct taped and hidden in a plastic bin inside a closet | Repl.2 | [5, 6, 7] | 37.09 |
| 8: anderson police sgt. chad boynton said jones , who was hospitalized for depression and suicidal thoughts , had refused to speak with detectives | Ignore | [5, 6, 7] | 37.09 |

Reference summary: geraldine jones charged with murder , kidnapping , criminal confinement , gary , indiana , woman , 36 , charged in death of samantha fleming , 23 . posed as a social worker to lure fleming and her baby from their home . had called the victim 's mother and lied to her to get information . allegedly stabbed fleming to death and hid body in a closet ; jones had faked a pregnancy and planned to claim the baby girl , serenity . police tracked her to a hospital in texas , where she was visiting her mother . jones is now awaiting extradition to indiana .

NN-SE [7] uses a cascade of CNN and RNN as sentence encoder to generate sentence representations and make extraction decisions on top of these representations. SummaRuNNer [28] employs a similar hierarchical encoder, but its predictions are more interpretable and can be broken down into several abstractive features like information content, salience, novelty and so on. REFRESH [29] treats extractive summarization as a sentence ranking problem and proposes a novel training algorithm to globally optimize the ROUGE evaluation through reinforcement learning. BanditSum [9, 12] formulates extractive summarization as a contextual bandit problem and trains the model with policy gradient to maximize the ROUGE score. RNES [40] combines the cross-sentence coherence and the ROUGE score of the extraction as the reward signal to get informative and coherent summaries. BERTSUMEXT [22] obtains sentence representations from BERT [8] followed by several stacked inter-sentence Transformer [35] and makes decisions on top of that. Self-Supervised [39] and HIBERT [44] propose novel pretraining tasks aiming to capture the global context at the document level, then the model is fine-tuned with the extractive labels. MATCHSUM [47] creates a paradigm shift and formulates extractive summarization as a semantic text matching problem. DISCOBERT [41] is a BERT-based model that prevents introducing redundant or uninformative phrases into summary by extracting finer-grained sub-sentential discourse units as candidates for extractive selection. HSG [38] utilizes semantic nodes of different granularity levels to enrich the cross-sentence relations, thus improving the performance of extractive summarization.

Abstractive approaches Unlike exclusively copying content from the original document in extractive summarization, abstractive models synthesize the summary in a word-by-word manner from scratch, thus may produce novel words and phrases that are not featured in the original document.

AEDRNN [27] uses RNN with attention mechanism as the base model and optimizes the model with several techniques, including adopting structure-aware hierarchical attention and enhancing word vectors with POS/NER tagging information and TF/IDF statistics. CopyNet [13] and PGN [32] enable the model to generate out-of-vocabulary words by directly copying them from the input document. Furthermore, PGN [32] proposes a coverage mechanism to record which words in the document have been attended to and penalize the model for repeatedly attending to same words, seeking to alleviate the generation of repeated content in the output summary. Models solely trained with cross-entropy loss suffer from the exposure bias, DRM [30] utilizes a mixed objective function of supervised learning and reinforcement learning to ease the situation. DCA [49] distributes the task of encoding a long document to multiple collaborating encoders, each in charge of a subsection, and employs a single decoder for the summary generation. ASGARD [16] utilizes structured representation from knowledge graph and designs a reward based on multiple choice cloze test to encourage producing informative and faithful summaries.

Combined approaches Combined approaches utilize both summarization techniques so that the abstractive model benefits from the information produced by the extractive model. In general, the combined approach first uses extractive methods to identify salient text spans, and then uses abstractive methods to generate the summary conditioning on these salient text spans.

UnifiedSum [15] proposed a model that treats sentence extractive probability (from extractor network) as sentence-level attention to re-weight word-level attention distribution (from abstractor network). They also introduced a novel inconsistency loss to penalize the inconsistency between two levels of attention. FastAbsRL [6] exerts an extractor agent to extract salient sentences

from the document. Then an abstractor agent rewrites these sentences into concise summary sentences via compression and paraphrase. Bottom-Up [11] proposes a two-stage bottom-up summarizer. The model first adopts a content selector to identify tokens that should be included in the summary. The summary is generated by a modified pointer generator network whose copy attention distribution is restricted to the summary worthy tokens recognized from the previous step. BERT-Abs [19] applies pre-trained BERT to rank sentence singletons and pairs and then compress or fuse top-ranked instances to summary sentences one after another. SENECA [33] deploys an entity-aware content selection module to collect salient sentences, and then an abstract generation module generates summaries utilizing cross-sentence information.

5 Conclusion

In this paper, we study the two intractable disadvantages in the existing auto-regressive extractive summarization models, i.e., the partial extraction discrepancy and the lead bias, which impair the effectiveness of these models in generating informative document summaries. We then fix the partial extraction discrepancy by explicitly predicting the summary update action for each sentence. Furthermore, we introduce an external replacement locator module to alleviate lead bias by enabling extracted sentences to be replaced by better new sentences. The experimental results on the benchmark CNN and DailyMail datasets show the superiority of AES-Rep compared with the current state-of-the-art baselines.

Acknowledgements This work was partially supported by the Australian Research Council under Grant No. DE210100160.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References


1. Allahyari, M., Pouriye, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: Text summarization techniques: a brief survey. *Int. J. Adv. Comput. Sci. Appl.* **8**(10) (2017)
2. Bae, S., Kim, T., Kim, J., Lee, S.g.: Summary level training of sentence rewriting for abstractive summarization. In: *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp 10–20 (2019)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: *ICLP* (2015)

4. Cao, Z., Wei, F., Li, W., Li, S.: Faithful to the original: Fact aware neural abstractive summarization. *AAAI* **32**(1) (2018)
5. Celikyilmaz, A., Bosselut, A., He, X., Choi, Y.: Deep communicating agents for abstractive summarization. In: *NAACL*, pp 1662–1675 (2018)
6. Chen, Y.C., Bansal, M.: Fast abstractive summarization with reinforce-selected sentence rewriting. In: *ACL*, pp 675–686 (2018)
7. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: *ACL*, pp 484–494 (2016)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL*, pp 4171–4186 (2019)
9. Dong, Y., Shen, Y., Crawford, E., van Hoof, H., Cheung, J.C.K.: Extractive summarization as a contextual bandit. In: *EMNLP*, pp 3739–3748 (2018)
10. Gehrmann, S., Deng, Y., Rush, A.: Bottom-up abstractive summarization. In: *EMNLP*, pp 4098–4109 (2018a)
11. Gehrmann, S., Deng, Y., Rush, A.: Bottom-up abstractive summarization. In: *EMNLP*, pp 4098–4109 (2018b)
12. Grenander, M., Dong, Y., Cheung, J.C.K., Louis, A.: Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. In: *EMNLP-IJCNLP*, pp 6019–6024 (2019)
13. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: *ACL*, pp 1631–1640 (2016)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR*, pp 770–778 (2016)
15. Hsu, W.T., Lin, C.K., Lee, M.Y., Min, K., Tang, J., Sun, M.: A unified model for extractive and abstractive summarization using inconsistency loss. In: *ACL*, pp 132–141 (2018)
16. Huang, L., Wu, L., Wang, L.: Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In: *ACL*, pp 5094–5107 (2020)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *ICLP* (2015)
18. Koupaee, M., Wang, W.Y.: Wikihow: A large scale text summarization dataset. [arXiv:181009305](https://arxiv.org/abs/181009305) (2018)
19. Lebanoff, L., Song, K., Dernoncourt, F., Kim, D.S., Kim, S., Chang, W., Liu, F.: Scoring sentence singletons and pairs for abstractive summarization. In: *ACL*, pp 2175–2189 (2019)
20. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out*, pp 74–81 (2004)
21. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: *EMNLP-IJCNLP*, pp 3721–3731 (2019a)
22. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: *EMNLP-IJCNLP*, pp 3730–3740 (2019b)
23. Liu, Y., Titov, I., Lapata, M.: Single document summarization as tree induction. In: *NAACL*, pp 1745–1755 (2019)
24. Luo, L., Ao, X., Song, Y., Pan, F., Yang, M., He, Q.: Reading like HER: Human reading inspired extractive summarization. In: *EMNLP-IJCNLP*, pp 3033–3043 (2019)
25. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: *EMNLP*, pp 1412–1421 (2015)
26. Moratanch, N., Chitrakala, S.: A survey on extractive text summarization. In: *2017 International Conference on Computer, Communication and Signal Processing*, pp 1–6. *IEEE* (2017)
27. Nallapati, R., Zhou, B., dos Santos C.N., Gülçehre, Ç., Xiang, B.: Abstractive text summarization using sequence-to-sequence rnns and beyond. In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp 280–290 (2016)
28. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: *AAAI*, pp 3075–3081 (2017)
29. Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarization with reinforcement learning. In: *NAACL*, pp 1747–1759 (2018)
30. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. In: *ICLP*. vol abs/1705.04304 (2018)
31. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *EMNLP*, pp 1532–1543 (2014)
32. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: *ACL*, pp 1073–1083 (2017)
33. Sharma, E., Huang, L., Hu, Z., Wang, L.: An entity-driven framework for abstractive summarization. In: *EMNLP-IJCNLP*, pp 3280–3291 (2019)

34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: *NIPS*, pp 6000–6010 (2017)
36. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: *ICLP* (2018)
37. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: *NIPS*, pp 2692–2700 (2015)
38. Wang, D., Liu, P., Zheng, Y., Qiu, X., Huang, X.: Heterogeneous graph neural networks for extractive document summarization. In: *ACL*, pp 6209–6219 (2020)
39. Wang, H., Wang, X., Xiong, W., Yu, M., Guo, X., Chang, S., Wang, W.Y.: Self-supervised learning for contextualized extractive summarization. In: *ACL*, pp 2221–2227 (2019)
40. Wu, Y., Hu, B.: Learning to extract coherent summary via deep reinforcement learning. In: *AAAI*, pp 5602–5609 (2018)
41. Xu, J., Gan, Z., Cheng, Y., Liu, J.: Discourse-aware neural extractive text summarization. In: *ACL*, pp 5021–5031 (2020)
42. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: *NAACL*, pp 1480–1489 (2016)
43. Zhang, X., Lapata, M., Wei, F., Zhou, M.: Neural latent extractive document summarization. In: *EMNLP*, pp 779–784 (2018)
44. Zhang, X., Wei, F., Zhou, M.: Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. In: *ACL*, pp 5059–5069 (2019)
45. Zheng, H., Lapata, M.: Sentence centrality revisited for unsupervised summarization. In: *ACL*, pp 6236–6247 (2019)
46. Zhong, M., Liu, P., Wang, D., Qiu, X., Huang, X.J.: Searching for effective neural extractive summarization: What works and what's next. In: *ACL*, pp 1049–1058 (2019)
47. Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., Huang, X.: Extractive summarization as text matching. In: *ACL*, pp 6197–6208 (2020)
48. Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., Zhao, T.: Neural document summarization by jointly learning to score and select sentences. In: *ACL*, pp 654–663 (2018)
49. Çelikyılmaz, A., Bosselut, A., He, X., Choi, Y.: Deep communicating agents for abstractive summarization. In: *NAACL*, pp 1662–1675 (2018)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Tianyu Zhu¹ · Wen Hua¹  · Jianfeng Qu² · Saeid Hosseini³ · Xiaofang Zhou⁴

Tianyu Zhu
tianyuzhu@uqconnect.edu.au

Jianfeng Qu
jfq@suda.edu.cn

Saeid Hosseini
sahosseini@su.edu.om

Xiaofang Zhou
zxf@cse.ust.hk

¹ School of ITEE, The University of Queensland, Brisbane, Australia

² Soochow University, Suzhou, China

³ Faculty of Computing and IT, Sohar University, Sohar, Oman

⁴ Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, Hong Kong