

# Saliency Allocation as Guidance for Abstractive Summarization

Fei Wang<sup>†\*</sup>, Kaiqiang Song<sup>†\*</sup>, Hongming Zhang<sup>‡</sup>, Lifeng Jin<sup>‡</sup>, Sangwoo Cho<sup>‡</sup>  
Wenlin Yao<sup>‡</sup>, Xiaoyang Wang<sup>‡</sup>, Muhao Chen<sup>†</sup> and Dong Yu<sup>‡</sup>

<sup>†</sup>University of Southern California; <sup>‡</sup>Tencent AI Lab, Seattle

{fwang598, muhaoche}@usc.edu

{riversong, hongmzhang, lifengjin, swcho, wenlinyao, shawnxywang, dyu}@global.tencent.com

## Abstract

Abstractive summarization models typically learn to capture the salient information from scratch implicitly. Recent literature adds extractive summaries as guidance for abstractive summarization models to provide hints of salient content and achieves better performance. However, extractive summaries as guidance could be over strict, leading to information loss or noisy signals. Furthermore, it cannot easily adapt to documents with various abstractiveness. As the number and allocation of saliency content pieces vary, it is hard to find a fixed threshold deciding which content should be included in the guidance. In this paper, we propose a novel summarization approach with a flexible and reliable saliency guidance, namely SEASON (Saliency Allocation as Guidance for Abstractive Summarization). SEASON utilizes the allocation of saliency expectation to guide abstractive summarization and adapts well to articles in different abstractiveness. Automatic and human evaluations on two benchmark datasets show that the proposed method is effective and reliable. Empirical results on more than one million news articles demonstrate a natural fifteen-fifty saliency split for news article sentences, providing a useful insight for composing news articles.<sup>1</sup>

## 1 Introduction

Abstractive summarization seeks to generate concise descriptions about synoptic information of longer documents (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017). Tackling this task can provide users with improved dissemination and acquisition of more readable content in long documents. More concretely, it allows for enhanced selection, compression and retrieval of Web-scale textual information that benefits other NLP tasks

<sup>\*</sup>Work done during Fei Wang’s internship at Tencent AI Lab Seattle. The first two authors contributed equally.

<sup>1</sup>Code and model weights are available at <https://github.com/tencent-ailab/season>.

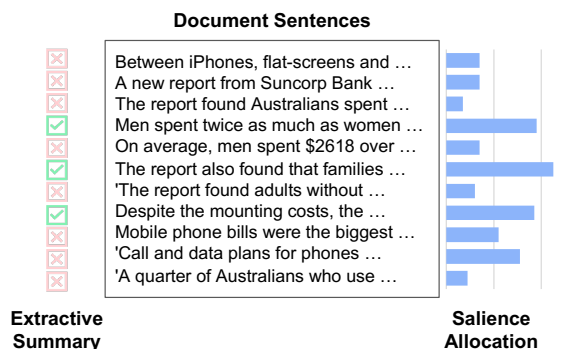


Figure 1: Illustration of different guidance. Extractive summary is a strict guidance consisting of extracted sentences labeled with check-mark. Saliency allocation is a flexible guidance mapping sentences to different saliency degrees shown as a bar chart.

such as machine reading comprehension (Inoue et al., 2021), mention linking (Cheng et al., 2015), claim verification (Yin et al., 2021), and information extraction (Lu et al., 2022).

Abstractive summarization models are typically trained end-to-end using large collections of paired corpora of raw documents and human-written summaries to directly perform sequence-to-sequence generation. In terms of deciding what to include in the generated summaries, these models implicitly learn to capture the salient information from scratch. Accordingly, recent literature has attempted to add auxiliary extractive saliency guidance for abstractive summarization models to give them a higher-level understanding of input documents, among which, extractive summaries appear to provide the most effective guidance (Li et al., 2020; Jin et al., 2020; Dou et al., 2021). Methods following this strategy learn to first perform extractive summarization, then perform abstraction on top of the extractive summaries (Hsu et al., 2018; Pilault et al., 2020; Dou et al., 2021).

However, incorporating extractive summaries as a form of guidance is evidently imperfect, even though it improves the overall performance of ab-

stractive summarization in some cases (Dou et al., 2021): 1) Extractive summaries are not reliable guidance. When there are too many summary-worthy sentences in the document, selecting a part of them may prone to information loss. When there are too few or no summary-worthy sentences, using the selected extractive summaries could be noisy and confusing to the model. 2) Extractive summaries are not flexible to adapt to different cases. The number and allocation of salience content pieces can vary by documents. Rather than extracting a fixed number of sentences, a flexible guidance should select salient content based on document properties. An imperfect selection process may also lead to further model biases, such as positional biases or length biases (Zhong et al., 2019). As the summarization process can differ for distinct documents (Grusky et al., 2018; Koupaei and Wang, 2018), a reliable guidance should allow flexible content selection, and be adaptive to documents with different abstractiveness.

In this paper, we propose a novel summarization approach with a flexible and reliable salience guidance, namely SEASON (Salience Allocation as Guidance for Abstractive Summarization). Salience is the degree to which a sentence contributes to the central idea of a document, and its allocation means how salience is distributed among all sentences in a document. To estimate the salience allocation, a linear classifier is trained on top of the encoder. This estimation is incorporated into the decoder with Salience-Aware Cross-Attention (SACA). It provides the flexibility to decide how much signal to accept from the salience guidance to supervise the abstractive summarization. The ground-truth salience label is assigned to each sentence based on its similarity with the ground-truth summary. Meanwhile, the number of salience degrees and their cut-off thresholds are decided based on the corpus to balance informativeness and prediction accuracy. To further improve the robustness of the summarization model, we apply label smoothing between adjacent salience degrees during training, and use the expectation of salience as a more robust salience estimation.

The technical contributions of this work are three-fold. *First*, we develop a new method for abstractive summarization on Transformer-based encoder-decoder architecture with the allocation of salience expectation as flexible guidance (§3). Our method provides reliable guidance that adapts

well to articles in different abstractiveness (§5.1). *Second*, we show the effectiveness and reliability of our proposed method comparing to the existing methods in both automatic (§4.2) and human evaluation (§5.3). *Third*, empirical results on more than one million news articles show a natural *fifteen-fifty salience split* for news article sentences (§4.3), providing a useful insight for composing news articles.

## 2 Related Work

### Joint extractive and abstractive summarization.

Extractive summarization and abstractive summarization are two general paradigms of text summarization (See et al., 2017; Grusky et al., 2018). Extractive summarization ensures the faithfulness of the generated summary but is not able to properly summarize documents when rephrasing is needed (Liu and Liu, 2009). Abstractive summarization, comparatively, is more flexible but may suffer from hallucination (Maynez et al., 2020).

A series of studies attempt to benefit from the advantages of both paradigms by combining them. Hsu et al. (2018) encourage the word-level attention of an abstractive summarization model and the relative sentence-level extraction probability from an extractive summarization model to be consistent. More recent studies show that conducting abstractive summarization with extractive summaries as a part of the input leads to better performance (Saito et al., 2020; Pilault et al., 2020; Dou et al., 2021). Extractive summarization can also work as an effective content selector for abstractive summarization when summarizing long documents (Manakul and Gales, 2021). Some studies (Gehrmann et al., 2018; Li et al., 2020; Saito et al., 2020) also consider to extract key words or phrases instead of summary worthy sentences as guidance, but their performances are not as good as those using sentences (Dou et al., 2021).

Our work extends the strict extractive summary guidance to a soft guidance of salience allocation. The proposed guidance is more flexible, reliable and adaptive, leading to better performance.

**Selective attention.** Selective attention is a psychological concept referring to the differential processing of simultaneous sources of information (Johnston and Dark, 1986). Incorporating prior knowledge through selective attention is widely explored in natural language processing, especially in recent NLP models with attention mechanism (Lin et al.,

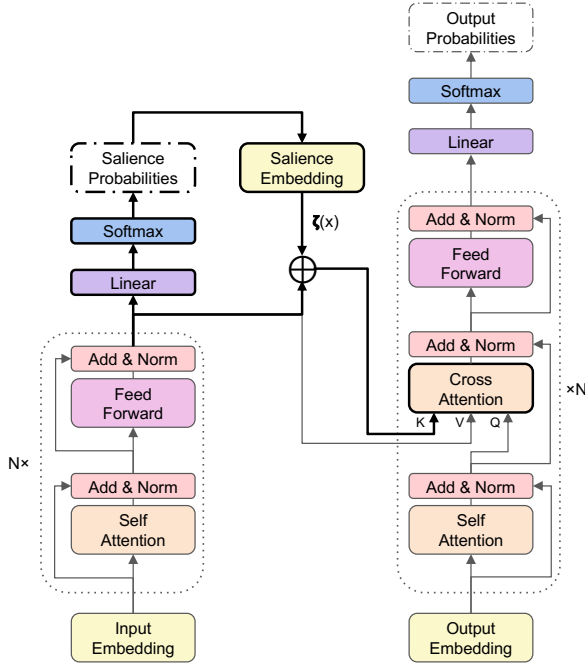


Figure 2: Model architecture of SEASON. The proposed modules are highlighted with bold lines. SEASON adds a saliency predictor on top of the encoder, maps (the expectation of) salience degrees to corresponding embeddings, and adds these saliency embeddings to the key vectors of cross attention.

2016; Sukhbaatar et al., 2019; Pruthi et al., 2020; Beltagy et al., 2020; Wang et al., 2022). To modify the summarization process with selective attention, previous studies either adjust the attention scores based on content selection probabilities directly (Hsu et al., 2018; Saito et al., 2020; Li et al., 2021), or appending selected content in the input (Saito et al., 2020; Dou et al., 2021). Recent studies show that the latter method with sentence-level content selection performs better (Dou et al., 2021).

Different from prior studies, SEASON maps salience degrees to distinct embeddings and adds them to the encoder outputs as key vector for cross-attention. This gives our model the flexibility to decide how much signal to accept from the saliency guidance for supervising the abstractive summarization process. This strategy achieves better performance in comparison with previous saliency-guided selective attention methods.

### 3 SEASON

In this work, we employ a Transformer-based encoder-decoder model for abstractive summarization. As shown in Fig. 2, our model SEASON encapsulates saliency prediction and text summariza-

tion in a single network. We perform multi-task end-to-end training, and inference via one forward pass. During training, the model jointly learns to predict the degree of salience for each sentence and is guided with ROUGE-based ground-truth salience allocation to generate the abstractive summary. During inference, SEASON predicts the expected salience allocation intermediately with the encoder outputs, and uses this predicted information to guide the decoder to generate the summary.

#### 3.1 Problem Formulation

Our assumption comes from an intuition that knowing the content salience allocation helps the model to pay attention to important content and generate more informative summaries. Although the content salience allocation is a built-in attribute of the source document, it is hard for the model to leverage this attribute without direct supervision (Li et al., 2020; Saito et al., 2020; Dou et al., 2021).

Let  $\mathbf{x}$  be the sequence of input tokens in the source document, and  $\mathbf{y}$  be the sequence of the summary tokens, where every token  $x_i$  or  $y_i$  is in the vocabulary  $\mathcal{V}$ . We use  $z_j$ , where  $j \in \{1, \dots, N\}$ , to represent the salience degree of the  $j$ -th sentence in the input document. We define  $o_i$  as the sentence index for the  $i$ -th token, where  $o_i \in \{1, \dots, N\}$ . The salience allocation is defined as  $\zeta(\mathbf{x}) = [\mathbf{f}(z_{o_1}), \dots, \mathbf{f}(z_{o_{|\mathbf{x}|}})]$ .<sup>2</sup> The problem can be formulated as follows:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{k=1}^{|\mathbf{y}|} p_{\theta}(y_k | \mathbf{y}_{<k}, \mathbf{x}, \zeta(\mathbf{x})). \quad (1)$$

In Eq. 1, each token prediction is conditioned on the previously decoded summary tokens, the input tokens in the source document, and the allocation of salience of the source document.

#### 3.2 Saliency Allocation Prediction

To predict salience degrees of input sentences, we slightly modify the encoder input sequence by adding a special token at the beginning of each sentence, obtaining their last-layer hidden states as sentence representations:

$$[\mathbf{h}_1^{sent}, \dots, \mathbf{h}_n^{sent}] = \text{Encoder}(\hat{\mathbf{x}}), \quad (2)$$

where  $\mathbf{h}_j^{sent}$ ,  $j \in \{1, \dots, N\}$ , is the contextualized embedding of the  $j$ -th sentence, and  $\hat{\mathbf{x}}$  is the modified input sequence. Then, sentence representations

<sup>2</sup> $\mathbf{f}(\cdot)$  is a function that maps the sentence salience degree to an embedding vector. In our implementation, we use the ground-truth salience embedding for training, and the expected embedding over the inferred salience distribution for testing.

are fed into a single-layer classification head:

$$P(z_j = l|\mathbf{x}) \propto \exp\left(\frac{\mathbf{w}_l^T \mathbf{h}_j^{sent} + b_l}{\tau}\right), \quad (3)$$

where  $\tau$  is a sharpening coefficient for the salience degree distribution,  $l \in \{1, \dots, L\}$  is the index of salience degree,  $L$  is the number of salience degrees,  $\mathbf{w}_l$  and  $b_l$  are trainable parameters. We provide discussions on  $L$  and  $\tau$  in §4.3 and §5.4 respectively. The design above allows the model predict salience allocation with minimal modifications on the architecture.

### 3.3 Salience-Aware Cross-Attention

To explicitly incorporate the salience allocation into the model, we develop a salience-aware cross-attention (SACA) module. SACA first maps the salience degrees to trainable salience embeddings:

$$\mathbf{f}(z_j) = \mathbf{Emb}(z_j). \quad (4)$$

This operation is intuitive when using ground-truth salience degrees. For predicted salience degrees, SACA needs to perform an estimation on the salience embedding with the inferred salience distribution. A simple **hard** estimation can be achieved by directly taking the embedding of degree  $l$  that maximizes the probability:

$$\mathbf{f}(z_j) = \mathbf{Emb}(\arg\max_l P(z_j = l|\mathbf{x})). \quad (5)$$

However, this direct estimation does not take the uncertainty of prediction into consideration, so we propose the **soft** estimation that calculates the expectation for the salience embedding:

$$\mathbf{f}(z_j) = \sum_{l=1}^L \mathbf{Emb}(z_j = l) P(z_j = l|\mathbf{x}). \quad (6)$$

We compare these two estimation methods comprehensively in §5.4. Next, SACA incorporates the salience allocation in the cross-attention layer to guide summary generation on the decoder side. SACA adds the sentence salience embedding to the encoder hidden state of each token belonging to the sentence as the key state for cross-attention. The cross-attention is formulated as:

$$\text{CrossAttn}(Q, K, V) = \text{MultiheadAttn}(Q, K, V),$$

where the attention query  $Q = \mathbf{h}^{decoder}$  thereof corresponds to the hidden state of the decoder, the attention key  $K = \mathbf{h}^{encoder} + \zeta(\mathbf{x})$  is the sum of the encoder hidden state and the salience embedding,

and the value  $V = \mathbf{h}^{encoder}$  is composed of the original encoder hidden state. In comparison with adding salience scores to cross-attention scores directly, SACA allows the model to learn how much signal to take from the salience guidance.

### 3.4 Learning Objectives

In training, SEASON learns to predict the salience allocation and generate the summary simultaneously. For salience prediction, we use the averaged cross-entropy loss on each predicted sentence:

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{j=1}^N \log P(z_j|\mathbf{x}). \quad (7)$$

In addition, we apply label smoothing (Diaz and Marathe, 2019) to the salience degrees for denoising. Specifically, a probability  $\beta$  is evenly assigned to salience degrees adjacent to the ground-truth degree. Analysis in §5.4 shows its effectiveness comparing with common label smoothing. For summary generation, we use the ground-truth salience allocation as input, and apply the averaged cross-entropy loss on each predicted token as below:

$$\mathcal{L}_{lm} = -\frac{1}{|\mathbf{y}|} \sum_{k=1}^{|\mathbf{y}|} \log p_{\theta}(y_k | \mathbf{y}_{<k}, \mathbf{x}, \zeta(\mathbf{x})). \quad (8)$$

We further combine two loss functions together with a coefficient  $\alpha$  that balances the two:

$$\mathcal{L}_{total} = \mathcal{L}_{lm} + \alpha \mathcal{L}_{cls}. \quad (9)$$

§5.4 shows that SEASON is not sensitive to  $\alpha$ .

## 4 Experiment

In this section, we first describe our experimental setting, including datasets, baselines, evaluation metrics and implementation details (§4.1). Then, we show the model performance on two summarization datasets (§4.2), and provide an insight on salience threshold selection (§4.3).

### 4.1 Experimental Setup

**Datasets.** We evaluate our method on two news summarization datasets. For both datasets, we use the original news article as input and the human-written summary as the ground-truth output. *CN-NDM* (See et al., 2017) consists of news articles and their human-written abstracts from CNN and Daily Mail websites, including 287,226/13,368/11,490 training/validation/test pairs. On average, each article has 781 words and each abstract contains 56



words. *Newsroom* (Grusky et al., 2018) contains news articles and summaries written by authors and editors from 38 major newsrooms published between 1998 and 2017. The dataset includes 995,041/108,837/108,862 training/validation/test pairs. On average, each article has 659 words and each summary has 27 words.

**Metrics.** We report widely used ROUGE metrics (Lin, 2004), including ROUGE-1 (R-1), ROUGE-2 (R-2), and sentence-level ROUGE-L (R-L)  $F_1$  scores with rouge-score python package.<sup>3</sup>

**Baselines.** We compare our system with three types of strong baselines, including extractive, abstractive, and mixed summarization methods. *LEAD-3* (See et al., 2017) is a common extractive summarization baseline that extracts the first three sentences as the document summary. *BertSum-Ext* (Liu, 2019) fine-tunes BERT for extractive summarization. *MatchSum* (Zhong et al., 2020) learns the semantic matching between candidate summary and source document using contrastive learning. *HAHSum* (Jia et al., 2020) incorporates a hierarchical attentive heterogeneous graph network in Albert (Lan et al., 2019) to perform redundancy-aware sentence embedding for extractive summarization. *Point-Generator* (See et al., 2017) first introduces the copy mechanism with coverage regularization for attentive seq2seq models in abstractive summarization. *BART* (Lewis et al., 2020) is an encoder-decoder Transformer model with denoising seq2seq pre-training. *PEGASUS* (Zhang et al., 2020) uses gap-sentence generation for pre-training an encoder-decoder Transformer on abstractive summarization tasks. *CIT+SE* (Saito et al., 2020) uses a key word extractor and selective encoding mechanism to guide abstractive summarization. *GSum* (Dou et al., 2021) uses the extracted summary from MatchSum as guidance to supervise BART for abstractive summarization.

**Implementation details.** We fine-tune BART-large on CNNDM and Newsroom datasets. For training data, we prepend a special token in front of each sentence for calculating its sentence representation. Each input sequence is truncated to 1024 tokens (including special tokens), to fit the maximum input length for BART. According to the summary length distribution of CNNDM and Newsroom datasets, we truncated the reference summary to be 128 and 256 tokens respectively to ensure more than 99%

<sup>3</sup><https://pypi.org/project/rouge-score/>

System	R-1	R-2	R-L
<i>CNNDM</i>			
LEAD-3	40.34	17.70	36.57
MatchSum	44.41	20.86	40.55
HAHSum	44.68	21.30	40.75
Point-Generator	39.53	17.28	36.38
BART	44.16	21.28	40.90
PEGASUS	44.17	21.47	41.11
CIT + SE	45.80	22.53	42.48
GSum	45.94	22.32	42.48
BART*	44.21	21.23	41.17
SEASON	<b><u>46.27</u></b>	<b><u>22.64</u></b>	<b><u>43.08</u></b>
<i>Newsroom</i>			
LEAD-3	30.49	21.27	28.42
Point-Generator	26.02	13.25	22.43
PEGASUS	45.15	33.51	41.33
BART*	45.50	33.05	41.69
SEASON	<b><u>46.00</u></b>	<b><u>33.37</u></b>	<b><u>42.03</u></b>

Table 1: Results on CNNDM and Newsroom test sets. Best scores are in bold. Scores significantly better than the best baseline model are underlined ( $p < 0.001$ ). Results with \* are reproduced by us. Other numbers are from prior papers.

of the reference summaries are fully preserved. For inference, we use the predicted soft estimation for allocation of expected salience. The predicted probability of salience degree is sharpened with a temperature  $\tau = 0.5$ . We use beam search with beam size of 5, length penalty of 1.5 and 3-gram blocking. According to their ROUGE-L  $F_1$  scores against the ground-truth summary, we split sentences into  $L = 3$  categories of salience degrees: 1) The most important top 15% sentences, 2) the bottom 50% least important ones, and 3) everything in between. More discussions regarding this setup is in §4.3).

## 4.2 Main Results

Tab. 1 shows the results on the two summarization datasets. For baselines on CNNDM, joint extractive and abstractive summarization methods (i.e. CIT+SE and GSum) perform better than independent extractive (i.e. LEAD-3, MatchSum and HAHSum) and abstractive summarization methods (i.e. Point-Generator, BART and PEGASUS) when using the same backbone models. Among the joint summarization baselines, using extractive summary as guidance (i.e. GSum) performs better than using key words as guidance (i.e. CIT+SE), which agrees with the observation by Dou et al. (2021). Our method promisingly improves the original BART by 2.06/1.41/1.91 points in terms of ROUGE-1/2/L  $F_1$  scores, indicating the multi-degree salience ex-

#degree	T-1	T-2	T-3	R-1	R-2	R-L
2	85%	-	-	45.94	22.52	42.74
3	50%	85%	-	<b>46.38</b>	<b>22.83</b>	<b>43.18</b>
4	30%	50%	85%	46.37	22.73	43.15

Table 2: Number of salience degrees, their best percentile thresholds (T-x), and achieved results on CNNDM dev set.

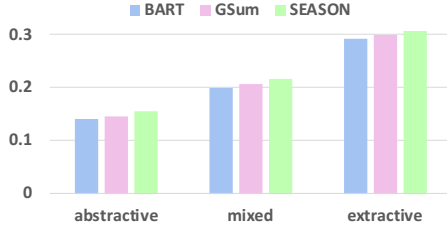


Figure 3: R-2 scores by density on CNNDM test set.

peptation allocation effectively guides the model to generate better-quality summaries. In comparison with GSum, our method achieves improvements of 0.33/0.32/0.60 points in terms of ROUGE-1/2/L F1 scores, w/o the help of SOTA extractive summarization system and with less additional parameters, indicating the proposed guidance is more effective than extractive summaries. Results on Newsroom dataset further verify that our method can achieve consistent improvements on different datasets. In comparison with the vanilla BART model, our method achieves 0.50/0.32/0.34 points improvements in terms of ROUGE-1/2/L F1 scores.

### 4.3 The Fifteen-Fifty Phenomenon

The number of salience degrees and thresholds to delimit them are the important hyper-parameters to discretize the proposed guidance. We apply a greedy search algorithm to find the best thresholds. First, we compute salience scores of all sentences in the corpus. In this work, we use ROUGE-L F1 between each document sentence and corresponding reference summary to represent salience, and find the best threshold for two salience degrees. Then we gradually add one more salience degree and search the additional threshold. The results on CNNDM is shown in Tab. 2. Splitting all sentences into three salience degrees by top 15% and bottom 50% salience scores leads to the best ROUGE-L F1. As the number of salience degrees  $L$  increases, the model performance first increases and then decreases. We attribute this phenomenon to the trade-off between informativeness and prediction accuracy of the guidance. Although a more fine-grained salience guidance is more informative,

	Informativeness	Faithfulness	Fluency
BART	87.21	76.77	85.86
GSum	78.45	<b>79.46</b>	26.94*
SEASON	<b>88.89</b>	78.11	<b>87.88</b>
Ground-Truth	77.78	75.76	72.39

Table 3: Percentage of positive votes ('Yes') on informativeness, faithfulness and fluency of summaries. \*GSum predictions provided by the authors are lower-cased and lemmatized, which hinders the fluency.

	1st	2nd	3rd	4th	avg.
BART	34.68	30.64	21.21	13.47	2.13
GSum	11.11	15.49	24.92	48.48	3.11
SEASON	<b>35.02</b>	29.63	24.24	<b>11.11</b>	<b>2.11</b>
Ground-Truth	19.19	24.24	29.63	26.94	2.64

Table 4: Percentage of ranking and the average rank by human evaluation.

our model generates summaries based on predicted guidance during inference, where error propagation exists. Increasing the number of salience degrees to predict also increases the risk of misclassification. Furthermore, we find the best number and thresholds of salience degrees for summarization is consistent on Newsroom, indicating the salience split by top fifteen and bottom fifty percentile is a nature property of news articles. This phenomenon may provide a useful insight for composing news articles by journalists.

## 5 Analysis

To gain further insights on the proposed method, we perform additional analyses on CNNDM to comprehensively investigate performance by abstractiveness (§5.1), summary length (§5.2), human evaluation (§5.3), and the impact of different model components (§5.4). A case study is also presented in §5.5.

### 5.1 Performance by Abstractiveness

To understand how adaptive our method is on documents with different abstractiveness, we split all documents into three subsets of equal sizes based on their density scores following Grusky et al. (2018). Results are shown in Fig. 3. SEASON performs better than baselines on all subsets, indicating that our method is adaptive to documents with different abstractiveness. The improvements on *abstractive* and *mixed* subsets are slightly higher than that on the *extractive* subset, indicating abstract documents benefit more than extractive ones from a flexible salience guidance.

SACA	MTL	R-1	R-2	R-L
✗	✗	44.21	21.23	41.17
✗	✓	44.57	21.55	41.49
✓(pred)	✓	46.27	22.64	43.08
✓(gold)	✓	54.85	31.36	52.14

Table 5: Results of models w/ or w/o salience-aware cross-attention (SACA) and multi-task learning (MTL) on CNNDM test set. For SACA, we provide results with both predicted and gold salience information.

$\alpha$	R-1	R-2	R-L
0.5	46.49	22.77	43.47
1.0	46.50	22.74	43.46
1.5	46.48	22.81	43.44

Table 6: Results on CNNDM dev set with different loss weights  $\alpha$  for salience prediction loss.

## 5.2 Summary Length

As SEASON achieves better performance under different abtractiveness with a more flexible salience guidance, a followup research question is: *Does the flexible salience guidance help predict summary length more accurately?* To answer this question, we compute the average lengths of Ground-Truth summaries and summaries generated by SEASON and baseline systems. The average summary length of Ground-Truth, SEASON, BART, and GSum are respectively 54.8, 59.0, 60.7, and 72.0. Among these methods, SEASON gave the closest average summary length to Ground-Truth. Moreover, while both of SEASON and GSum introduce sentence-level salience guidance to BART, they change the summary length in opposite directions.

## 5.3 Human Evaluation

We further evaluate the system outputs of SEASON, BART, Gsum and the Ground-Truth with human subjective evaluation. We randomly pick 100 instances from CNNDM test set. For each raw document in those instances, we provide the summary generated by each system and the ground-truth summary. We hire human evaluators on Amazon Mechanical Turk to answer three Yes/No questions for the four summaries and rank them. Each instance is assigned to 3 different human evaluators to answer the following three questions (Song et al., 2021). a) *Informativeness*: Does the summary include the major information from the news? b) *Faithfulness*: Does the summary give any additional information not covered by the news? c) *Fluency*: Is the summary grammatical and well-formed?

Tab. 3 reports the average percentage for each

Strategy	$\beta$	R-1	R-2	R-L
-	-	46.38	22.83	43.18
all	0.1	46.47	22.71	43.42
	0.2	46.40	22.68	43.38
adjacent	0.1	46.44	22.79	43.38
	0.2	<b>46.48</b>	22.81	<b>43.44</b>

Table 7: Results on CNNDM dev set with different label smoothing strategies.

Strategy	$\tau$	R-1	R-2	R-L
hard	-	46.48	22.81	43.44
soft	0.1	46.76	23.08	43.57
	0.2	46.94	<b>23.24</b>	43.75
	0.5	<b>46.98</b>	23.20	<b>43.78</b>
	1.0	46.59	22.75	43.44

Table 8: Results on CNNDM dev set with different salience estimation methods.  $\tau$  is the sharpening coefficient in softmax.

method to get a positive answer on the corresponding question. Among the three systems, SEASON performs the best on informativeness and fluency, while GSum performs the best on faithfulness. This indicates that our flexible guidance helps the model to identify salient content accurately and rephrase them properly. Not surprisingly, systems with guidance (i.e. SEASON and Gsum) are more faithful to original content than a system without any guidance (i.e. BART). Tab. 4 shows the ranking results. SEASON has the highest percentage of the highest-ranked summaries and the lowest percentage of the lowest-ranked summaries. It also has the best average rank. These results further demonstrate that summaries generated by SEASON are of high quality. Interestingly, we find that the ground-truth is not always the best choice in human evaluation. This observation aligns with the findings in prior studies (Maynez et al., 2020; Song et al., 2020; Fabri et al., 2021). It could happen since both human composition of summaries and human justification on their qualities could be subjective. Thus, ground-truth news summaries written by editors may not always be the first choice of readers. It also indicates that human evaluators could not distinguish between the real human writer and our automatic summarizer, and actually prefer our system outputs more than the ground-truth summaries.

## 5.4 Ablation Study

For all the experiments in this section, we use the default setting introduced in §4.1, unless discussed otherwise with different hyper-parameter values.

<b>BART</b>	New York-based writer Danielle Page set out to ask every cabbie she came across to dispense their best piece of relationship advice. The drivers, many of whom are married themselves, revealed their personal tips, life lessons and cultural anecdotes all in the name of love.
<b>GSum</b>	New York-based writer Danielle Page set out to ask every cabbie she came across to dispense their best piece of relationship advice in hopes of unlocking the key to a successful union. The drivers, many of whom are married themselves, revealed their personal tips, life lessons and cultural anecdotes all in the name of love. A 60-year-old named Michael revealed that his trick to marital bliss is putting his wife's happiness above his own – but insists that what really makes a relationship work is finding a partner who will do the same for you.
<b>SEASON</b>	New York-based writer Danielle Page set out to ask every cabbie she came across to dispense their best piece of relationship advice.
<b>Gold</b>	New York-based writer Danielle Page set out to ask every cabbie she came across to share their tips on finding - and keeping - a partner.
<b>BART</b>	Researchers from Texas A&M School of Public Health found that hospitalizations from car crashes dropped 7 percent between 2003 and 2010 in the 45 states with texting bans. Arizona, Texas, Montana, Missouri, and Oklahoma are the only five states in America that do not have texting at the wheel bans for all drivers.
<b>GSum</b>	Researchers from Texas A&M School of Public Health found that hospitalizations from car crashes dropped 7 percent between 2003 and 2010 in the 45 states with texting bans when compared to states with no restrictions. Drivers between the ages of 25 and 40 are the most likely group of people to get in an accident related to texting and driving.
<b>SEASON</b>	Researchers from Texas A&M School of Public Health found that hospitalizations from car crashes dropped 7 percent between 2003 and 2010 in the 45 states with texting bans. Arizona, Texas, Montana, Missouri, and Oklahoma are the only five states in America that do not have texting at the wheel bans for all drivers. The study found that older drivers were more likely to make a texting and driving mistake than a younger driver.
<b>Gold</b>	Study found that hospitalizations from car crashes dropped 7 percent between 2003 and 2010 in the 45 states with texting bans. Arizona, Texas, Montana, Missouri, and Oklahoma are the only five states in America that do not have texting at the wheel bans for all drivers. The study also found that older drivers were more likely to make a texting and driving mistake than a younger driver.

Table 9: Case Study. Continuous word spans overlapped with the gold summary of more than 3 words are in blue. Continuous word spans in the gold summary not covered by any prediction are in red. Baselines may suffer from extra details or information loss due to no or imperfect salience guidance.

**Multi-Task Learning.** We first investigate the effectiveness of salience prediction as an auxiliary task by removing salience-aware cross-attention. In this setting, the model jointly predicts the salience allocation and the abstractive summary, but does not feed the gold or predicted salience allocation to the decoder. That means the salience allocation is only used as supervision signals but not (intermediate) input features. As shown in Tab. 5, with MTL solely, the model can achieve 0.36/0.32/0.32 points improvements in terms of R-1/2/L. This indicates that the salience prediction task can not only provide effective guidance for abstractive summarization, but also act as supervision for learning more robust representations.

**Salience-Aware Cross-Attention.** We examine the effectiveness of the proposed salience-aware cross-attention module from two perspectives in Tab. 5. First, we provide the gold salience labels instead of predicted ones to explore its upper bound. The performance increases by 8.58/8.72/9.06 points with a perfect salience predictor. This result indicates that a better estimation of the salience can be helpful for further improving the abstractive summarization performance. Second, we compare it with the original cross-attention module while keeping the auxiliary task and observe a performance drop by 1.70/1.09/1.59 points in terms of R-1/2/L. This indicates that salience-aware cross-attention is essential for selecting important content accurately.

**Coefficient of Multi-Task Learning.** We further examine the influence of the coefficient  $\alpha$  of multi-task learning in Tab. 6. We test three different  $\alpha$  values and observe that the largest difference of

R1/2/L are within 0.02/0.07/0.03. According to the results, SEASON is not sensitive to  $\alpha$ , indicating our model architecture is robust.

**Adjacent Label Smoothing.** We compare different label smoothing strategies in Tab. 7. In general, label smoothing improves model generalization and calibration (Müller et al., 2019), therefore benefits the overall performance. Given the same smoothing probability  $\beta$ , adding label smoothing to adjacent salience degrees performs better than adding label smoothing on all other salient degrees.

**Salience Estimation.** We compare the effectiveness of using soft (Eq. 6) and hard (Eq. 5) strategies for salience estimation in Tab. 8. Computing the expectation with raw probabilities (i.e.,  $\tau = 1.0$ ) brings 0.11 points improvements on ROUGE-1. By adjusting the sharpness of probability distribution with the sharpening coefficient  $\tau$ , ROUGE-1/2/L improvements become 0.50/0.39/0.34 points, respectively. As defined in Eq. 3,  $\tau$  represents the confidence on predictions, and a lower  $\tau$  leads to sharper probability distribution. In our experiments,  $\tau = 0.5$  performs the best.

## 5.5 Case Study

We present a case study in Tab. 9 with two representative examples to illustrate the advantage of SEASON. In the first case, BART tends to generate extra details without the help of proper guidance when only one sentence is enough to summarize the document. GSum is guided by an extractive summary consisting of three sentences, so not surprisingly it provides even more details. In the second case, BART infers without any salience guidance



and ignores an important finding of the research. GSum selects exactly three sentences as guidance, thus it misses key information when multiple sentences are similarly important but some of them are not included in the guidance. SEASON performs well for both cases, indicating it is adaptive to documents of different properties.

## 6 Conclusion

In this paper, we propose SEASON, an abstractive summarization approach guided with salience allocation expectation. In SEASON, the salience guidance is adaptive to documents with different abstractiveness, and the salience-aware cross-attention module is flexible to decide how much signal to accept from the salience guidance. Automatic and human evaluation further demonstrate the effectiveness and reliability of our proposed method. Comparing to the strong baseline model (i.e. BART), our method achieves 2.06/1.41/1.91 ROUGE-1/2/L performance gain on CNNDM, and 0.33/0.32/0.60 performance gain on Newsroom. Finally, the empirical results on more than one million news articles demonstrate a natural fifteen-fifty salience split for news article sentences providing a useful insight for composing news articles.

## Acknowledgement

We thank all reviewers for their valuable suggestions. We also thank Zi-Yi Dou and Pengfei Liu for providing summaries generated by GSum and stored at DataLab (Xiao et al., 2022)<sup>4</sup>. This work was done when Fei Wang was doing an internship at Tencent AI Lab Seattle. Fei Wang is partially supported by Annerberg Fellowship. Muhao Chen is supported by the National Science Foundation of United States Grant IIS 2105329.

## Limitation

In this study we have experimented with using ROUGE-L  $F_1$  as the salience measurement. However, other metrics for text summarization can serve as alternatives, such as BLEU (Papineni et al., 2002) and BARTScore (Yuan et al., 2021). Choosing among more metrics for salience measurement can be explored in future work. The number of salience degrees and thresholds to delimit them could be language-dependent. The fifty-fifteen phenomenon is observed on two of the most represen-

tative English news summarization datasets. Future work on other languages may need to search the best salience degrees and thresholds of each language. Despite we use BART as our base model and maximum likelihood estimation (MLE) as the learning objective in this study, the proposed method can also be applied to other backbones and learning objectives (Zhang et al., 2020; Liu and Liu, 2021; Liu et al., 2022). While we have limited the proposed technique to abstractive summarization on news articles, future research can extend SEASON to other domains, such as scientific publications (Cohan et al., 2018) and podcasts (Song et al., 2022). In terms of evaluation, we focus on the supervised and in-domain setting. Future work may also consider to extend our method to zero-shot, few-shot, cross-domain, or cross-dataset settings. In addition to abstractive summarization, future research can also extend SEASON to other NLP tasks requiring salience-awareness, such as fact verification (Wang et al., 2021), information retrieval (Xiong et al., 2018) and distantly supervised relation extraction (Lin et al., 2016).

## Ethical Consideration

A general issue of automatic text summarization is intellectual property problem caused by copying content from the raw document to the generated summary. This work seeks to improve abstractive summarization models with salience allocation as guidance. As the proposed guidance is more flexible than extractive summaries, it is likely to reduce copying content. Although we create salience guidance based on ground-truth summaries, the documents and ground-truth summaries remains the same as it is in the original dataset, ensuring no further social bias is introduced.

## References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.
- Gong Cheng, Danyun Xu, and Yuzhong Qu. 2015. Summarizing entity descriptions for effective and efficient human-centered entity linking. In *Proceedings of the 24th International Conference on World Wide Web*, pages 184–194.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In

<sup>4</sup><https://datalab.nlpedia.ai/>

- Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621.
- Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4738–4747.
- Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842.
- Alexander Richard Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719.
- Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141.
- Naoya Inoue, Harsh Trivedi, Steven Sinha, Niranjan Balasubramanian, and Kentaro Inui. 2021. [Summarize-then-answer: Generating concise explanations for multi-hop reading comprehension](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6064–6080, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631.
- Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Semsum: Semantic dependency guided neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8026–8033.
- William A Johnston and Veronica J Dark. 1986. Selective attention. *Annual review of psychology*.
- Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Haoran Li, Arash Einolghozati, Srinivasan Iyer, Bhargavi Paranjape, Yashar Mehdad, Sonal Gupta, and Marjan Ghazvininejad. 2021. Ease: Extractive-abstractive summarization end-to-end using the information bottleneck principle. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 85–95.
- Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020. Keywords-guided abstractive sentence summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8196–8203.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.
- Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 261–264.
- Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.
- Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072.
- Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual*

- Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022. Summarization as indirect supervision for relation extraction. *arXiv preprint arXiv:2205.09837*.
- Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.
- Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319.
- Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.
- Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *arXiv preprint arXiv:2003.13028*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Kaiqiang Song, Chen Li, Xiaoyang Wang, Dong Yu, and Fei Liu. 2022. Towards abstractive grounded summarization of podcast transcripts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4407–4418.
- Kaiqiang Song, Bingqing Wang, Zhe Feng, and Fei Liu. 2021. [A new approach to overgenerating and scoring abstractive summaries](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1392–1404, Online. Association for Computational Linguistics.
- Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8902–8909.
- Sainbayar Sukhbaatar, Édouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive attention span in transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 331–335.
- Fei Wang, Kexuan Sun, Jay Pujara, Pedro Szekely, and Muhao Chen. 2021. Table-based fact verification with salience-aware learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4025–4036.
- Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. *arXiv preprint arXiv:2205.03972*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

- Yang Xiao, Jinlan Fu, Weizhe Yuan, Vijay Viswanathan, Zhoumianze Liu, Yixin Liu, Graham Neubig, and Pengfei Liu. 2022. [DataLab: A platform for data analysis and intervention](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 182–195, Dublin, Ireland. Association for Computational Linguistics.
- Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 575–584.
- Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. [DocNLI: A large-scale dataset for document-level natural language inference](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.
- Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. [Searching for effective neural extractive summarization: What works and what’s next](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.



## A Implementation Details

Our Implementation is based on Huggingface Transformers (Wolf et al., 2020), Pytorch-lightning<sup>5</sup> and Lightning-Transformers<sup>6</sup>. We further use DeepSpeed (Rasley et al., 2020) Stage2 and half-precision to speed up the training process. We applied the BART-large model consisting of 400M parameters and fine-tuned them on CNNDM<sup>7</sup> and Newsroom<sup>8</sup> dataset with  $8 \times V100$  GPU(32GB) for 10 epochs. It takes about 5 hours for training on CNNDM and 32 hours on NEWSROOM. We set our batch size to be 96 to maximize the utilization of the GPU memory. For Optimization, we use AdamW (Loshchilov and Hutter, 2018) with learning rates of  $3e - 5$ . The momentum parameters are 0.9 and 0.99. Our weight decay is set to be 0.01 for parameters other than bias term and LayerNorm layers. We uses a linear warmup strategy, the number of warmup steps are 1,500 and 5,000 on CNNDM and Newsroom respectively. The dropout rate is 10%. We follow the BART (Lewis et al., 2020) fine-tuning and use gradient clipping of 0.1. The coefficient of multi-task learning  $\alpha$  is 1.5. We uses adjacent label smoothing and the smoothing probability  $\beta$  is 20%. For inference, we use the predicted soft estimation for allocation of expected salience. The predicted probability of salience degree is sharpened with a temperature  $\tau = 0.5$ . We use beam search with beam size of 5 with length penalty of 1.5 and 3-gram blocking. The minimum and maximum decoding length is 20 and 256. For human evaluation, we choose master workers on Amazon Mechanical Turk<sup>9</sup> with more than 90% approve rates and more than 100 approved HIT.

---

<sup>5</sup><https://www.pytorchlightning.ai>

<sup>6</sup><https://github.com/Lightning-AI/lightning-transformers>

<sup>7</sup>[https://huggingface.co/datasets/ccdv/cnn\\_dailymail](https://huggingface.co/datasets/ccdv/cnn_dailymail)

<sup>8</sup><https://lil.nlp.cornell.edu/newsroom/download/index.html>

<sup>9</sup><https://www.mturk.com>