Keyword-aware Abstractive Summarization by Extracting Set-level Intermediate Summaries

Yizhu Liu
Shanghai Jiao Tong University
Shanghai, China
liuyizhu@sjtu.edu.cn

Qi Jia
Shanghai Jiao Tong University
Shanghai, China
Jia_qi@sjtu.edu.cn

Kenny Q. Zhu*
Shanghai Jiao Tong University
Shanghai, China
kzhu@cs.sjtu.edu.cn

ABSTRACT

Abstractive summarization is useful in providing a summary or a digest of news or other web texts and enhancing users reading experience, especially when they are reading on small displays such as mobile phones. However, existing encoder-decoder summarization models have difficulty learning the latent *alignment* between source documents and summaries because of their vast disparity in length. In this paper, we propose a extractor-abstractor framework in which the keyword-based extractor selects a few sets of salient sentences from the input document and then the abstractor paraphrases these sets of sentences in parallel, which are more aligned to the summary, to generate the final summary. The new extractor and abstractor are pretrained from a set of "pseudo summaries" extracted by specially designed heuristics, and then further trained together in a reinforcement learning framework. The results show that the proposed model generates high-quality summaries with faster training speed and less training memory footprint, and outperforms the state-of-the-art models on CNN/Daily Mail, Webis-TLDR-17, Webis-Snippet-20, WikiHow and DUC-2002 datasets.

CCS CONCEPTS

- **Information systems → Summarization**.

KEYWORDS

Abstractive Summarization, Alignment, Set-level Pseudo-summaries, Reinforcement Learning

ACM Reference Format:
Yizhu Liu, Qi Jia, and Kenny Q. Zhu. 2021. Keyword-aware Abstractive Summarization by Extracting Set-level Intermediate Summaries. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449906

1 INTRODUCTION

Abstractive summarization is the task of creating a short, accurate, and informative summary from a long text document without using the exact sentences from the source. This is useful in generating a snippet or digest for a searched web page or other web text. The essence of summarization is to compress information from the input document and retain only most important information in the output. This process can be seen as *aligning* the salient information from the source to the output. Recently, encoder-decoder (enc-dec) models with attention mechanism [3, 4, 8, 9, 21, 26] have made great progress on abstractive summarization. The attention mechanism is a way of capturing the *alignment* between input sequences of the encoder and decoder, trying to tell which parts of the source document are relevant to which parts in the summary. However, since lots of non-essential parts in the source document are omitted in the summary, the alignment using only attention mechanism is unsatisfactory. Table 1 shows that two state-of-the-art enc-dec models, i.e., PointGen [26] and BART [13], both frequently make incorrect alignments by either missing some salient parts or including redundancies.

Source document
new delhi, india police have arrested four employees. federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** pointed at the store 's **changing room**. four employees of the store have been arrested, but its manager was still at large saturday . state **authorities** found an overhead camera that the minister had spotted and determined that it was indeed able to take **photos** of customers. authorities sealed off the store and summoned six top officials from fabindia. the **arrested** staff have been charged with voyeurism and breach of privacy. if **convicted**, they could spend up to three years in jail .
Reference summary
federal **education minister** smriti irani visited a **fabindia** store in goa , saw **cameras**. **authoroities** discovered the **cameras** could capture **photos** from the store 's **changing room**. the four store workers **arrested** could spend **three years** each in prison if **convicted** .
PointGen [26]
four employees of a popular indian ethnic chain have been arrested, but its manager was still at large . authorities sealed off the store and summoned six top officials from fabindia.
BART [13]
federal education minister smriti irani was visiting a fabindia outlet in the tourist resort state of goa . she discovered a surveillance camera pointed at the changing room . four employees of the store have been arrested , but the manager is still at large . the arrested staff have been charged with voyeurism and breach of privacy

Table 1: Summarization results by SOTA models (PointGen and BART). The bold words or phrases are salient information. The underlined parts are redundant information in the output. Some salient information is also missing from the output.

[*] Corresponding author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449906

An alternate view [1, 5] of the summarization process is to paraphrase the *salient parts* in the source document, i.e., summary sentences are aligned to the *salient parts* of source document (see Table 1). This gives rise to a two-stage, extractor-abstractor (ext-abs) framework, which first selects salient sentences from the source (extractor) and then paraphrases the selected ones to generate a summary (abstractor). The ext-abs framework has two advantages: i) the input and output of the abstractor can be better aligned; ii) reduced size of the input to the abstractor reduces both training and inference time.

To train an ext-abs framework, one has first to generate the intermediate results, i.e., the salient sentences in the input document for all training samples. Since the real salient sentences are not known in practice, we call the intermediate result obtained algorithmically the *pseudo summary*. Pseudo summaries are used for training both the extractor and the abstractor in the ext-abs framework. As the intermediate result, the low-quality pseudo summaries can bring noises to the model. Better pseudo summaries can reduce the noise and enhance the alignment between encoder and decoder of abstractor. Previously, there are two types of heuristics to create pseudo summaries: sentence-level [5] and summary-level methods [20, 27].

Sentence-level methods assume that there is one unique salient sentence in the source that matches each sentence in the reference summary. To this end, they extract the sentence with the highest ROUGE score [15] for each reference sentence. This simple assumption gives rise to the design of parallel abstractors (one for each reference sentence) to achieve speed-up. However, the very nature of summarization dictates that a sentence in the summary may be condensed from multiple sentences in the source and not just one. For example, in Table 2, the first sentence in sentence-level pseudo summary is pertinent to both 1^{st} and 2^{nd} reference sentences in Table 1, while the second pseudo sentence misses out some information ("changing room") of 2^{nd} reference sentence. In response to this deficiency, summary-level methods were proposed to select the best combination of a subset of input sentences that maximizes ROUGE score with reference summary as a whole. Nevertheless, they lose the advantage of parallelism in the sentence-level approach. Worse still, when mixing all the sentences together, they treat every token equally in computing the ROUGE, resulting in pseudo summaries that are similar to the reference only by unimportant words. For example, The summary-level pseudo summary in Table 2 doesn't match the information about "authorities" and "arrested", which are more important in the story.

In this paper, we present a novel set-level matching heuristics that divides the reference summary into a few disjoint clusters of sentences, each of which represents a topic or an aspect, and matches a non-overlapping set of sentences in the source with each cluster of reference sentences. This new heuristics strives to trade off the pros and cons of the previous two approaches. Instead of assuming one-to-one or all-to-all alignment between the pseudo summary and the reference summary, we are assuming a many-to-many alignment, which allows for more flexible alignment while still achieving parallelism using multiple abstractors. When computing the similarity between the pseudo summary and the reference, on top of ordinary ROUGE scores, we emphasize keywords in the reference summary. This amounts to representing summaries not only as a sequence of words but also as a set of important keywords. Accordingly, we design a keyword-aware extractor which includes both an ordinary document encoder and a keyword encoder.

Sentence-level
1) federal education minister smriti irani was visiting a fabindia outlet in the tourist resort state of goa on friday when she discovered a surveillance camera pointed at the store's changing room. *2)* state authorities found an overhead camera that the minister had spotted and determined that it was indeed able to take photos of customers. *3)* if convicted, they could spend up to three years in jail.
Summary-level
federal education minister smriti irani was visiting a fabindia outlet in the tourist resort state of goa on friday when she discovered a surveillanc camera pointed at the store's changing room. if convicted, they could spend up to three years in jail.
Set-level based on Keywords
Set 1) federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** pointed at the **changing room**. state **authorities** found an overhead **camera** that the minister had spotted and determined that it was indeed able to take **photos** of customers. *Set 2)* four employees of the store have been **arrested**. if **convicted**, they could spend up to **three years** in jail.

Table 2: The pseudo summaries produced by different heuristics for the source and reference in Table 1.

One natural way to connect the extractor and abstractor into an end-to-end trainable model is to use reinforcement learning (RL). Previous ext-abs models use sentence-level [5] or summary-level [1] ROUGE scores as the reward. The sentence-level rewards can not properly reflect the quality of overall summary because of overlapping contents [1, 22], while summary-level rewards ignore the accuracy of the sentences extracted at each step. Therefore, we propose a comprehensive reward which is the weighted sum of sentence/set/summary-level ROUGE scores. This comprehensive reward can help the extractor select sentences that match abstractive reference summaries better.

In summary, our contributions are as follows:

(1) Our **set-level** matching heuristics extracts better pseudo summaries as the training data to pretrain both the extractor and the abstractor, and subsequently allows the abstractor to learn the alignments effectively. (See Section 2.1, Section 3.3.1)
(2) The use of **keywords** to represent salient concepts and entities in documents and summaries provides a significant boost in the ext-abs framework. (See Section 2.2, Section 2.3, Section 3.3)
(3) The integration of pretrained language models into a comprehensively rewarded RL gives a potent end-to-end summarization framework that outperforms the state-of-the-art (SOTA) methods on popular abstractive summarization datasets including CNN/Daily Mail, Webis-TLDR-17, Webis-Snippet-20, WikiHow and DUC-2002.(See Section 3.3)

2 APPROACH

Our new ext-abs framework is illustrated in Figure 1. There are three main components: a keyword-based extractor, an abstractor and a comprehensively rewarded reinforcement learning (RL). As a preprocessing step, we first obtain the set-level pseudo summary from the training data. We then pretrain the keyword-based extractor using the source document and the pseudo summary, and the parallel abstractor using the pseudo summary and the reference summary. Finally, we use RL to bridge the pretrained extractor and abstractor to further finetune the parameters in both models. The RL updates the extractor and abstractor by a comprehensive reward evaluating both the extracted intermediate summaries and abstractive summaries at sentence-level, set-level and summary-level.

In the rest of this paper, we use the following definitions.

- A source document D is a sequence of sentences $(d_0, ..., d_i, ...)$;
- A set-level *pseudo summary* P is a sequence of sentences organized in sets denoted as $(p^0_0, p^0_1, ..., p^l_i, ..., p^m_x)$ where p^l_i is the i^{th} sentence in the sequence that belongs to the l^{th} set;
- The output of extractor, i.e., *intermediate summary* Q, is a sequence of sentences denoted as $(q^0_0, q^0_1, ..., q^l_i, ..., q^M_y)$ similar to P;
- A *reference summary* R consists of sentences $(r_0, r_1, ..., r_i, ..., r_z)$;
- A *reorganized reference summary* $\hat{R}$ consists of sentences $(\hat{r}^0_0, \hat{r}^0_1, ..., \hat{r}^l_i, ..., \hat{r}^m_z)$, similar to P and Q;
- The generated abstractive summary A is a sequence of sentence $(a^0, a^1, ..., a^l, ..., a^M)$ and a denotes the set of sentences.
- t ranges over the time steps in both encoding and decoding.

Next, we describe the preprocessing of the training data to obtain pseudo summaries, and the key components in the framework. [1]

2.1 Data Pre-processing: Set-level Matching Heuristics

In order to enhance the alignment between pseudo summaries and generated summaries, we propose a set-level matching heuristics to obtain pseudo summaries based on a set of keywords.

We use TextRank algorithm [19] to extract the keywords from the reference summary and obtain the set-level pseudo summary by Algorithm 1. For instance, as shown in Figure 2 we extract the sentence sets covering the most reference keywords (bold) with the highest ROUGE-2 scores from the source document for each reference sentence. Then, if there is an overlap between two extracted sentence sets, the two sets will be merged into one and their reference sentences will also be merged into a longer sentence. In the end, each sentence set in pseudo summary has a corresponding sentence set in reference summary. As a result, in Figure 2 the 1st reference sentence matches source sentence 1), and the best matching for the 2nd reference sentence is the combination of source sentence 1) and 2). The pseudo summary set consisting of source sentence 1) and 2) is corresponding to the combination of 1st and 2nd reference sentences.

2.2 Keyword-based Extractor (KE)

In extractive summarization, we take document D as input and set-level pseudo summary P as output. Our extractor consists of a **dual encoder** and an **aligned pointer decoder**. The dual decoder has a **document encoder** and **keywords encoder**. The document encoder learns sentence representations using a language model and helps with natural language understanding. Keywords encoder learns keywords representations and guides the decoder to select more accurate sentences. The model is illustrated in Figure 3.

Algorithm 1: Extraction of Set-level Pseudo Summaries

```
Input: a document $D$, a reference summary $R$, a set of
       keywords $K$
Output: pseudo summary $P$ and reorganized reference
        summary $\hat{R}$
; // D and R are each a set of sentences
$len()$ computes the number of sentences in a text
$rec()$ and $f1()$ compute ROUGE-2 recall and F1 score
  between two texts
$o()$ computes the number of overlapping words between the
  two sequences
for $i = 0 \rightarrow len(R)$ do
    $d_i$ is the $i$-th sentence in $D$
    $r_i$ is the $i$-th sentence in $R$
    $k^i$ denotes the keywords of $r_i$
    Initialize $p \leftarrow init \in D$ with highest $rec(init, r_i)$
    $o_{max} \leftarrow o(init, k^i), f1_{max} \leftarrow f1(init, r_i)$
    $D' \leftarrow D - init, \hat{r} \leftarrow r_i$
    for $j = 0 \rightarrow len(D)$ do
        if $o(d'_j, k^i) > o_{max}$ or $(o(d'_j, k^i) = o_{max}$ and
         $f1(d'_j, r_i) > f1_{max})$ then
            $p \leftarrow p \cup \{d'_j\}$
            $o_{max} \leftarrow o(p, k^i)$
            $f1_{max} \leftarrow f1(p, r_i)$
    $d' \leftarrow d' - p$
    for $j = 0 \rightarrow len(d')$ do
        if $f1(d'_j, r_i) > f1_{max}$ then
            $p \leftarrow p \cup \{d'_j\}$
            $o_{max} \leftarrow o(p, k^i)$
            $f1_{max} \leftarrow f1(p, r_i)$
    Add $p$ into $P$
    Add $\hat{r}$ into $\hat{R}$
    while the last two sub-sets in $P$ have overlap do
        Merge last two sub-sets in $P$ Merge last two sub-sets
          in $\hat{R}$
return $P$, $\hat{R}$
```

Keywords Encoder. We use TextRank algorithm to receive a sequential list of keywords from the source document, ordered by their original positions in the source. We take convolutional neural network (CNN) model to embed extracted keywords as $(\mathbf{k_1}, \mathbf{k_2}, ..., \mathbf{k_{|k|}})$, where $|K|$ is the number of keywords. The combination of keywords representation and sentence representation enbodies the intuition that the keywords are more important carriers of the salient information and should be treated specially during sentence selection.

Document Encoder. We consider two options for the document encoder: training from scratch with *BiLSTM document encoder* and fine-tuning on pretrained model named *HIBERT document encoder*. The former is a standard document encoding model, and the latter is the state-of-the-art pretrained model for document encoding.

[1] Our framework is flexible with respect to the choice of document encoder and abstractor.

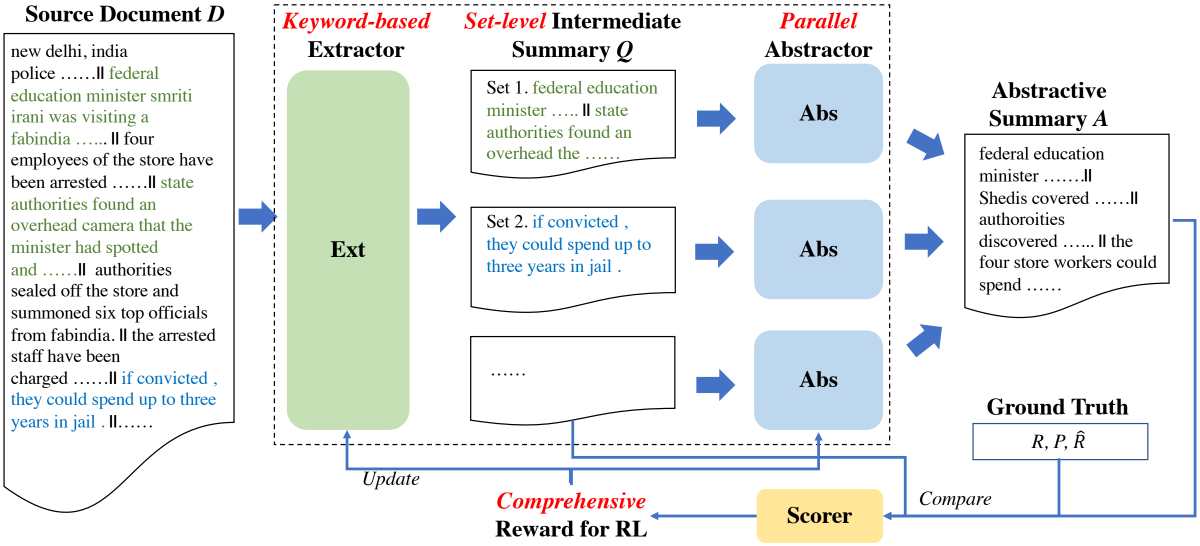
Source Document D
new delhi, india police ……‖ federal education minister smriti irani was visiting a fabindia …... ‖ four employees of the store have been arrested ……‖ state authorities found an overhead camera that the minister had spotted and ……‖ authorities sealed off the store and summoned six top officials from fabindia. ‖ the arrested staff have been charged ……‖ if convicted , they could spend up to three years in jail . ‖……
Keyword-based Extractor
Ext
Set-level Intermediate Summary Q
Set 1. federal education minister ….. ‖ state authorities found an overhead the ……
Set 2. if convicted , they could spend up to three years in jail .
……
Parallel Abstractor
Abs
Abs
Abs
Abstractive Summary A
federal education minister …….‖ Shedis covered ……‖ authoroities discovered …... ‖ the four store workers could spend ……
Ground Truth
R, P, R̂
Update
Comprehensive Reward for RL
Scorer
Compare

Figure 1: The overview of keyword-aware reinforced extractor-abstractor framework.

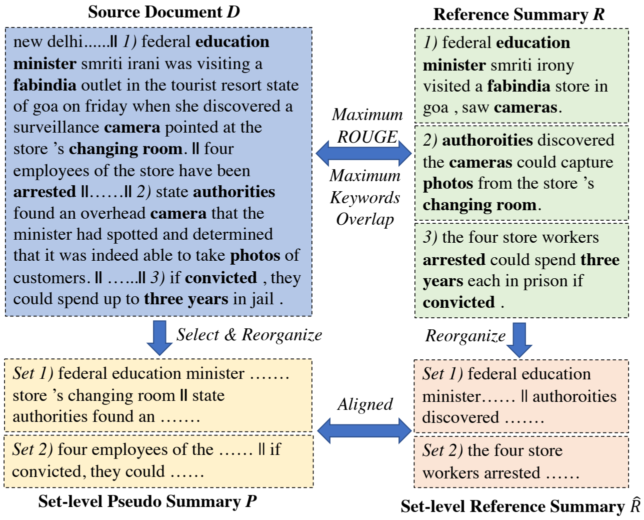
Source Document D
new delhi......‖ 1) federal education minister smriti irani was visiting a fabindia outlet in the tourist resort state of goa on friday when she discovered a surveillance camera pointed at the store 's changing room. ‖ four employees of the store have been arrested ‖......‖ 2) state authorities found an overhead camera that the minister had spotted and determined that it was indeed able to take photos of customers. ‖ ......‖ 3) if convicted , they could spend up to three years in jail .
Reference Summary R
1) federal education minister smriti irony visited a fabindia store in goa , saw cameras.
2) authoroities discovered the cameras could capture photos from the store 's changing room.
3) the four store workers arrested could spend three years each in prison if convicted .
Maximum ROUGE
Maximum Keywords Overlap
Select & Reorganize
Reorganize
Set 1) federal education minister ……. store 's changing room ‖ state authorities found an …….
Set 2) four employees of the …… ‖ if convicted, they could ……
Aligned
Set 1) federal education minister…… ‖ authoroities discovered …….
Set 2) the four store workers arrested ……
Set-level Pseudo Summary P
Set-level Reference Summary R̂

Figure 2: The process of creating Set-level pseudo summary and reorganized reference summary. The words or phrases in bold are keywords. ‖ denotes the sentence boundary.

The *BiLSTM document encoder* has two sub-encoders: a sentence encoder based on temporal CNN model and a document encoder based on bidirectional LSTM network. A document is represented as $(h_0, h_1, ..., h_n)$ where $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$ is the representation of i-th sentence.

The *HIBERT document encoder* is a pretrained encoder [34], which contains two Transformer-based sub-encoders. We combine the word embeddings and their corresponding position embeddings as the input and obtain the context sensitive sentence representations $(h'_0, h'_1, ..., h'_n)$ as the output.

Aligned Pointer Decoder. We extend Pointer Network [31] as the decoder. The pseudo summary consisting of multi-sentence sets is the input of the decoder. We extract a set of keywords for each multi-sentence set in the pseudo summaries and order them based on their positions in the input text. To distinguish the sentences and keywords in different sets, we set the representations for the placeholder <SEP> in pseudo summaries and pseudo keywords. The pseudo summary becomes $P = (p_0^0, p_1^0, ..., p_j^0, SEP, p_{j+1}^1, ..., P_x^m)$, and its keywords become $K = (k_0^0, k_1^0, ...k_j^0, SEP, k_{j+1}^1, ..., k_{|k|}^m)$. We randomly initialize the representation of h_{SEP} for pseudo summary and k_{SEP} for pseudo keywords.

At each time step t, we take the output of decoder attending to the encoder sentence representations as the predicted vector c_t^h, which is calculated by:

$$\begin{aligned} c_t^h &= \sum_i^n \alpha_{it}^h W^{a1} h_i \\ \alpha_t^h &= \text{softmax}(v^h \tanh(W^{g1} g_t + W^{h1} h_i)) \end{aligned} \tag{1}$$

where g_t is the decoder hidden state at step t. h_i is the sentence representation of i-th sentence based on document encoder (BiLSTM or HIBERT). a_t^h is the attention weights based on sentences. W and v in different labels are trainable parameters. Similarly, the keywords vector c_t^k can be computed.

We compute the current extraction probabilities using predicted sentence vector and keywords vector:

$$\begin{aligned} p(y_t | y_1, .., y_{t-1}, c^h, c^k) = \text{softmax}(v \tanh(W^g g_t + W^h c_t^h \\ + W^k c_t^k)) \end{aligned} \tag{2}$$

where y_t is the sentence with the highest probability at current step.

Combinatorial Loss. We propose a combinatorial loss to train extractor, including cross-entropy loss, keywords loss and set loss. The *cross-entropy loss* reflects the accuracy of one-to-one alignment between extracted sentences and pseudo summaries, which is

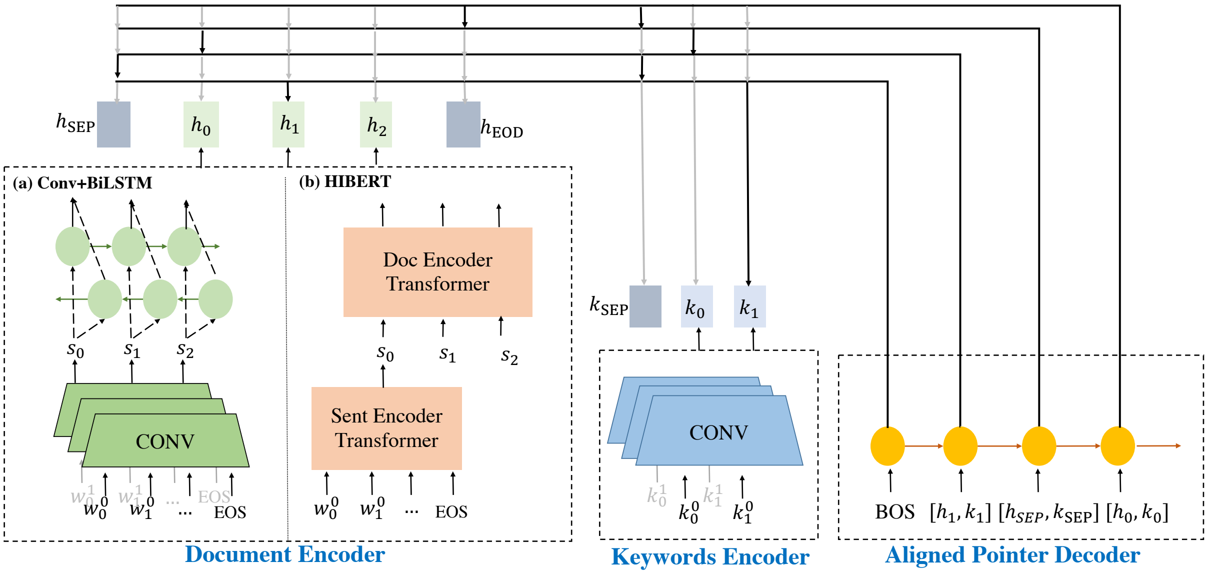

Figure 3: The architecture of baseline model and pretrained model for keyword-aware extractor. w represents words in corresponding sentences. s represents sentence embeddings. h_{SEP}, h_{EOD} and k_{SEP} are three random initialized vectors for special tokens, which are updated during training.

computed as:

$$L_{ce} = -\sum_{(P,D)\in T} \log(p(P|D)) \tag{3}$$

where T is training set with N samples. We use *keywords loss* to emphasize the importance of the related salient information. The probability of keywords extraction is computed as:

$$p(k_t|k_1, ..., k_{t-1}, c) = \text{softmax}(v \tanh(W'^g g_t + W'^k c_t^k)) \tag{4}$$

where k_t is the predicted keyword at t step. We compute the keywords loss based on the keywords ground truth K and source document D as:

$$L_{key} = -\sum_{(K,D)\in T} \log(p(K|D)) \tag{5}$$

As the output of extractor consists of multi-sentence sets, we need correctly predict <SEP> at the proper positions. For each training sample, we obtain intermediate summary Q extracted from source document D, yielded by greedily selecting sentence that maximizes the output probability at each time step. We align the <SEP> of P and Q in the same position by padding or truncation the sequence of sentence labels in the set of P. For example, given pseudo summary $P = (p_0, p_1, SEP, p_2, SEP)$ and extracted intermediate summary $Q = (q_0, SEP, q_1, q_2, SEP)$, we get aligned pseudo summary as $P' = (q_0, SEP, q_2, SEP, SEP)$. We define the *set loss* function as:

$$L_{set} = -\sum_{(P',D)\in T} \log(p(P'|D)) \tag{6}$$

We use the combinatorial loss as follows:

$$L_{cl} = \frac{1}{N}(\lambda_c L_{ce} + \lambda_k L_{key} + \lambda_s L_{set}) \tag{7}$$

2.3 Abstractor

The abstractor can paraphrase the inputs in parallel. We take set-level pseudo summaries and their reference summaries as the input and output of abstractor at training. The abstractor is an independent neural network without parameter sharing with the extractor.

In this work, we take two representative Enc-Dec model options as our abstractor: the standard Enc-Dec model PointerGen [26] with attention mechanism [18] and copy mechanism, and a pretrained language model BART [13] finetuned on our pseudo summaries and reference summaries.

Special loss. For training, we take pseudo summary P as input and reorganized reference summary $\hat{R}$ as output. Given a pseudo summary, our abstractor deal with the sets in pseudo summary in parallel, so we first compute the *cross-entropy loss* between i-th multi-sentence set of pseudo summary and reference summary

$$L'_{ce}(i) = -\log(p(\hat{r}_i|p_i)) \tag{8}$$

Then, we consider all of the sets in a complete summary. The loss of i-th set in pseudo summary is as follows:

$$\begin{aligned} L_{sp}(i) &= \frac{(1 + PoL(i))}{2} L'_{ce}(i) \\ PoL(i) &= \frac{L'_{ce}(i)}{\sum_{i=0}^{m} L'_{ce}(i)} \end{aligned} \tag{9}$$

where $PoL(i)$ is *propotion of the loss* of i-th sentence set over the complete summary. This can strengthen the penalty for worse predicted multi-sentence set in a complete summary.

2.4 Comprehensive Reinforcement Learning

We apply comprehensive reinforcement learning (CRL) to make ext-abs framework an end-to-end trainable model. We use policy gradient technique to optimize our model and take extractor as the RL agent.

During training, we first use extractor to obtain an intermediate summary Q, which is divided into several sentence sets by <SEP>.

Then, the abstractor paraphrases the sets in Q, and connects the rewritten sentences with <SEP> to generate an abstractive summary A. At each time step t, in order to compare the intermediate summary Q and pseudo summary P, we define a *sentence-level reward* using ROUGE-L (R-L) score between the sets of Q and P at the same position.

$$R_{sen}(t) = R\text{-}L_{F1}(q_t, p_t) \tag{10}$$

The sentence-level reward directly measures the accuracy of intermediate summary sentences. As the intermediate sentences and pseudo sentences are both extracted from source document, the R-L, calculating the longest common subsequence (LCS), is the best way to evaluate the intermediate sentences with the pseudo sentences.

To evaluate the alignment between intermediate summary Q and reorganized reference summary $\hat{R}$, we propose a *set-level reward*. We compute the *set-level reward* by ROUGE-2 (R-2) as:

$$R_{set}(t) = \begin{cases} R\text{-}2_{recall}(a^l, \hat{r}^l), & \text{if } t = b + |q^l| \\ R\text{-}2_{recall}(\text{concat}(q_b^l ... q_t^l), \hat{r}_l), & \text{otherwise} \end{cases} \tag{11}$$

where *concat* concatenates all the inputs. $\hat{r}^l$ is the l-th set of sentences in $\hat{R}$ and $|q^l|$ is the number of sentences in l-th set of Q. b is the index of the first sentence of l-th set in Q. $t = b + |q^l|$ means that the prediction of l-th set in intermediate summary is over. For *set-level reward*, at step t, we concatenate all of the extracted sentences in l-th set as a extracted set E_t^l and compute the R-2 score between E_t^l and its corresponding set $\hat{r}^l$ in reorganized reference summary. At the end of the prediction of this set, we compare the abstractive summary a_l generated from q^l with $\hat{r}^l$. Since reference summary is abstractive which has many variant, the R-2 matching bigram between summaries is more suitable. As the E_t^l is the part of the input of the abstractor, the ROUGE score reflects the alignment between the input and output of abstractor during test. The higher recall between E_t^l and $\hat{r}^l$ means that the E_t^l contains more information of $\hat{r}^l$ and can predict better abstractive summary.

Considering the quality of an overall generated summary, we compute *summary-level reward* as:

$$R_{sum}(t) = \begin{cases} R\text{-}2_{F1}(\text{concat}(a^0 ... a^l), \text{concat}(\hat{r}^0 ... \hat{r}^l)), \text{if } t = \sum_0^l |q^l| \\ R\text{-}2_{F1}(\text{concat}(q_0^0 ... q_t^l), R), \quad \text{otherwise} \end{cases} \tag{12}$$

where $t = \sum_0^l |q^l|$ means that prediction of l-th set in intermediate summary is over. For *summary-level reward*, we concatenate all of the extracted sentences as a extracted set E_t at each time step t. We use F1 score as reward, because the length should be considered during evaluating the whole generated summary, which can also reflects the alignment of abstrator. At the end of the prediction of each set, we compare the concatenated generated abstractive summary and corresponding reference summary. Especially, while the prediction of model is over, we compute the R-2 F1 score between generated abstractive summary A and reference summary R.

The total reward is the combination of above:

$$R_{overall} = \gamma_1 R_{sen} + \gamma_2 R_{set} + \gamma_3 R_{sum} \tag{13}$$

3 EVALUATION

In this section, we introduce the dataset and experimental setup. We compare our proposed framework, along with its variants, with existing summarizaiton models and demonstrate the advantages of our keyword aware models [2] trained on set-level pseudo summaries.

3.1 Datasets

In this experiment, we use 5 datasets which are either news, web pages or user generated QAs on the web for training and test.

CNN/Daily Mail [10] (CNNDM) is a popular summarization dataset, which contains 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs. We follow Nallapati [21] with the data preprocessing and use the non-anonymized version as See et al. [26].

Webis-TLDR-17 Corpus [32] (Web17), one of the first large-scale summarization datasets from social media domain, contains 3 million pairs of content and self-written summaries from Reddit.

Webis-Snippet-20 Corpus [4] (Web20) contains approximately 3.5 Million (webpage content, abstractive snippet) triples for the task of abstractive snippet generation of web pages. The corpus is compiled from the DMOZ Open Directory Project.

WikiHow Corpus [12] (Wiki) is a large-scale dataset using the online WikiHow knowledge base. Each article consists of multiple paragraphs and each paragraph starts with a sentence summarizing it. The dataset contains 200,000 long-sequence pairs.

DUC-2002 (DUC) is a test set of 567 document-summary pairs for single-document summarization. We use the models trained on CNNDM to do the test on DUC, which can evaluate the generalizability of the models.

3.2 Experimental Setup

3.2.1 Implementation details. We set batch size as 32 for all training processes. All models are optimized by Adam optimizer. In extractor, we take a single-layer CNN model with 100 dimensions as keywords encoder whose input are randomly initialized with 128-dimensional vectors. For pointer network decoder, we employ LSTM models with 256-dimensional hidden states. We implement our document encoders, BiLSTM encoder and HIBERT encoder, as described by Chen [5] and Zhang [34]. We fine-tune HIBERT encoder with *learning rate* (*lr*) $5e-5$ and warmup steps $4,000$. We set $\lambda_c = 1.0$, $\lambda_k = 0.5$, $\lambda_s = 0.5$ (Eq. 7). For abstractor, the *lr* of PG is $1e-03$. We follow Lewise [13] in fine-tuning BART with $lr = 3e-05$ and warmup = 500. For RL, the *lr* of RL as $1e-04$. We set $\gamma_{sen} = 0.5$, $\gamma_{set} = 1.0$, $\gamma_{sum} = 1.0$ (Eq. 13) with grid search on validation set.

3.2.2 Models under comparison. In this experiments, we evaluate different methods on above datasets. The brief description of these methods are shown in Table 3.

3.2.3 Evaluation Metrics. We evaluate the performance of our method by *automatic metrics* and *human evaluation*.

Automatic Metrics. ROUGE scores (F1) include ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L(R-L) [15].

[2] The data and source code are released on https://github.com/YizhuLiu/SetKE_ABS.

Abbrev.	Description
Extractive Summarization	
PN [5]	BiLSTM encoder with pointer decoder
PN$_{ad}$	BiLSTM encoder with aligned pointer decoder
KE	Keyword-based extractor with BiLSTM encoder
KE$_{cl}$	Keyword-based extractor with BiLSTM encoder and combinatorial loss
HIBERT [34]	Pretrained HIBERT model
HIBERT$_{ad}$	HIBERT encoder with aligned decoder
KE$_{HI}$	Keyword-based extractor with HIBERT encoder
KE$_{HIcl}$	Keyword-based extractor with HIBERT encoder and combinatorial loss
Abstractive Summarization	
PG [5, 26]	Pointer generator
PG$_{sl}$	PG in parallel with special loss
BART [13]	BART model
BART$_{sl}$	BART in parallel with special loss
KE$_{cl}$-PG$_{sl}$	2-stage KE$_{cl}$ and PG$_{sl}$
KE$_{HIcl}$-BART$_{sl}$	2-stage KE$_{HIcl}$ and BART$_{sl}$
FastAbs [5]	Ext-Abs framework training on sentence-level pseudo summaries
FastAbs$_{HB}$	Replace extractor and abstractor in FastAbs to HIBERT and BART
X-RL$_{sen}$	2-stage model *X* training on sentence-level reward
X-RL$_{sum}$	2-stage model *X* training on summary-level reward
X-CRL	2-stage model *X* training on CRL

Table 3: The abbreviation and description of different methods.

Human Evaluation. We randomly select 100 samples from each dataset and average the scores by three human annotators who are native or proficient English speakers. [3]

- **Manual Alignment Accuracy** (manAlign). We rank and score pseudo summaries with three-scale scores based on the informativeness and redundancy of pseudo summary with respect to reference, i.e., better (2.0), equal (1.0) and worse (0.0).
- **Keyword Coverage** reflects the accuracy of keywords in generated summary. Given a pair of generated summary and reference summary, we manually extract their keywords and sequence these keywords based on their locations in source. Keyword coverage is computed as the ROUGE-1 precision between generated and reference keywords sequences.
- **Readability**. We rank summaries generated by our best model and that of BART according to logical consistency with source document and informativeness. The summary should be labeled as better, equal or worse. includes the percentage of the number of summaries with different label to the total summaries.

3.3 Results

3.3.1 Pseudo Summary. In a two-stage framework, the pseudo summaries is critical to the training and testing of the model. Better intermediate summaries can enhance the alignment between inputs and outputs of the abstractor during training and generate more accurate abstractive summaries during testing. As shown in Table 4, our set-level keyword-based matching heuristics outperforms sentence-level and summary-level heuristics, achieving the best manAlign score. As shown in Table 2, the sentence-level pseudo summaries always ignore cross-sentence information. Summary-level pseudo summaries capture the information among sentences and get better ROUGE scores than sentence-level pseudo summaries. However, summary-level pseudo summaries cannot recognize important information in reference summary, which bring noise to the pseudo summaries. The proposed set-level heuristic extracts the most aligned multi-sentence set for one or more reference sentences, which can better align the reference sentences abstracted from multiple source sentences. As the set-level method is based on keywords, the pseudo summaries cover all keywords in reference summaries, significantly reducing salient information lose.

In order to examine the effects of different pseudo summaries on the model, we assume that the extractor is perfect and directly input three kinds of pseudo summaries to train and test the abstractors respectively. As shown in Table 4, the ROUGE scores between these generated summaries and references denote the upper bound of models on different pseudo summaries, which can reflect the alignment between pseudo summaries and reference summaries. The higher ROUGE scores, the more aligned dataset. The ROUGE scores of CNNDM dataset in Table 4 are much better than ROUGE scores of other dataset. Since some sentences in reference summaries of CNNDM dataset are extracted from the source documents, the quality of intermediate results has a greater impact on CNNDM dataset. The improvement of ROUGE scores reflects the enhancement of alignment. Compared with other datasets, the improvement of ROUGE score on Web20 is minimal. The reason is that the length of reference summaries of Web20 dataset is shorter than others, causing the similar pseudo summaries extracted through different heuristics.

The models trained on set-level pseudo summaries achieve the highest ROUGE scores on all of the datasets. This denotes that the abstractor models can benefit from training on set-level pseudo summaries. Thus, our proposed set-level matching heuristics can produce more aligned training pairs for generation and make the abstractor better.

3.3.2 Results for the framework. We compare our proposed models with existing models. Following previous work, we take reference summaries in datasets as the ground truth of extractive summaries and abstractive summaries.

Extractor. We train the extractor on (source document, pseudo summary) pairs. As shown in Table 5, the keyword-based extractor achieves higher ROUGE scores on various datasets. The basic models (PN$_{ad}$ and HIBERT$_{ad}$) with only one document encoder have been improved on ROUGE scores by adding keyword encoder (KE), which demonstrates that KE is useful to guide extractor to select more accurate sentences. After adding combinatorial loss (CL), the ROUGE scores become higher. The reason is that the composition of CL is consistent with the extraction of pseudo summaries and the structure of extractor. Besides, CL containing keywords loss can help extractor to select sentences with more keywords. The ROUGE scores of extracted summaries generated by HIBERT$_{ad}$ are higher since the HIBERT$_{ad}$ is fine-tuned on a pretrained model which can

[3] The Cohen's Kappa coefficient between annotators are 0.68 (manAlign), 0.72 (KC) and 0.64 (Read), indicating substantial agreement.

Data	Pseudo	R-1				R-2				R-L				manAlign
	summary	PG	PG$_{sl}$	BART	BART$_{sl}$	PG	PG$_{sl}$	BART	BART$_{sl}$	PG	PG$_{sl}$	BART	BART$_{sl}$	
CNNDM	sentence	48.75	49.70	50.55	50.64	26.16	26.63	27.34	27.60	45.98	46.94	47.02	47.59	0.65
	summary	48.98	-	51.20	-	26.85	-	28.32	-	46.41	-	48.43	-	0.70
	set	**49.31**	**50.2**	**52.02**	**52.53**	**27.12**	**27.64**	**28.66**	**28.83**	**49.34**	**49.88**	**48.72**	**49.12**	**1.65**
Web17	sentence	19.20	19.44	19.77	20.01	5.04	5.16	5.87	5.98	16.12	16.26	17.02	17.64	0.75
	summary	19.51	-	19.82	-	5.13	-	5.66	-	16.38	-	17.11	-	0.85
	set	**20.34**	**20.75**	**21.19**	**22.02**	**5.28**	**5.65**	**5.97**	**6.10**	**16.76**	**16.92**	**17.24**	**17.80**	**1.40**
Web20	sentence	19.26	19.24	19.55	19.61	5.07	5.50	6.12	6.14	17.56	17.96	18.21	18.37	0.94
	summary	19.28	-	20.70	-	5.03	-	6.21	-	17.27	-	18.26	-	0.97
	set	**19.30**	**19.46**	**21.22**	**21.43**	**5.32**	**5.67**	**6.34**	**6.54**	**17.58**	**18.02**	**18.27**	**18.45**	**1.09**
WiKi	sentence	27.01	28.17	28.74	29.02	10.40	11.06	11.98	11.75	20.79	21.76	21.22	22.78	0.78
	summary	32.28	-	33.47	-	11.27	-	12.32	-	25.25	-	26.12	-	0.86
	set	**34.07**	**34.76**	**35.06**	**35.45**	**11.76**	**12.16**	**12.37**	**12.94**	**26.22**	**27.61**	**27.33**	**28.02**	**1.36**

Table 4: The ROUGE scores of abstractors trained on pseudo summaries at different levels.

Models		CNN/DM			Web17			Web20			Wiki			DUC		
		R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Only Extractor																
PN$_{ad}$		37.02	16.62	33.78	16.17	3.13	10.55	7.81	1.40	7.02	18.65	3.99	14.88	35.43	15.20	31.72
KE		40.25	18.15	36.46	16.34	3.51	10.39	7.90	1.43	7.10	18.83	4.01	15.07	37.02	16.35	33.56
KE$_{cl}$		**40.38**	**18.95**	**36.94**	**17.00**	**3.83**	**10.76**	**8.04**	**1.50**	**7.33**	**19.50**	**5.35**	**15.62**	**37.79**	**17.73**	**33.95**
HIBERT$_{ad}$		41.71	19.35	38.44	18.25	3.95	14.20	7.93	1.55	7.72	20.78	5.79	16.27	38.63	18.04	36.27
KE$_{HI}$		42.61	19.50	38.57	18.32	4.11	14.34	9.01	1.97	8.62	21.22	5.84	16.44	39.17	18.45	36.20
KE$_{HIcl}$		**42.87**	**20.04**	**39.02**	**18.69**	**4.17**	**14.34**	**9.34**	**2.44**	**8.75**	**22.50**	**5.92**	**16.62**	**40.07**	**18.78**	**36.34**
Extractor-Abstractor w/o RL																
PN$_{ad}$	PG$_{sl}$	32.75	14.03	30.32	15.88	3.01	10.47	7.65	1.39	7.13	12.71	3.12	9.11	29.07	13.74	24.11
	BART$_{sl}$	40.12	17.71	32.35	16.04	3.48	10.95	8.15	1.50	8.28	19.11	4.92	16.80	36.20	16.38	29.67
KE	PG$_{sl}$	37.42	15.70	34.83	15.66	3.31	10.00	7.38	1.41	7.33	14.83	3.87	13.91	34.20	14.03	29.70
	BART$_{sl}$	40.65	18.29	35.32	16.14	3.72	12.31	7.65	1.44	7.13	19.66	5.12	16.82	37.20	16.76	33.46
KE$_{cl}$	PG$_{sl}$	38.09	16.61	35.64	16.55	3.75	10.77	7.25	1.44	7.36	18.85	4.23	16.52	34.88	15.23	31.00
	BART$_{sl}$	40.70	19.27	36.23	17.74	4.05	13.69	9.73	2.12	10.07	20.32	5.77	16.80	37.57	17.21	33.97
HIBERT$_{ad}$	PG$_{sl}$	38.45	16.03	33.85	16.78	3.21	10.98	7.36	1.40	7.52	18.83	4.75	16.27	32.16	15.74	33.11
	BART$_{sl}$	42.48	19.61	39.02	17.77	4.11	14.07	8.13	1.59	7.82	20.03	5.94	16.99	38.76	17.95	36.11
KE$_{HI}$	PG$_{sl}$	39.62	18.07	33.29	16.22	3.38	11.11	8.47	1.60	8.01	19.14	5.08	16.25	35.18	16.34	34.01
	BART$_{sl}$	42.19	19.82	38.57	18.53	4.16	14.27	12.16	2.53	**11.58**	22.17	6.82	18.24	39.73	18.94	36.38
KE$_{HIcl}$	PG$_{sl}$	40.63	18.11	36.34	18.59	3.66	12.52	8.37	1.67	7.90	20.47	5.66	16.27	35.66	17.12	34.09
	BART$_{sl}$	**42.84**	**20.13**	**39.08**	**18.75**	**4.20**	**14.66**	**12.71**	**2.89**	11.55	**25.70**	**7.52**	**20.08**	**40.24**	**19.01**	**36.49**

Table 5: The ROUGE scores of extractor and extractor-abstractor without RL.

enhance the language modeling ability. Compared with PN$_{ad}$, the HIBERT$_{ad}$ can capture more information about the relationship between inputs of the encoder and the decoder, including keyword information. Therefore, the improvements on different datasets of HIBERT document encoder (HIBERT$_{ad}$) are always less than BiLSTM document encoder (PN$_{ad}$). As shown in Table 6, compared with the extractor without keyword encoder, the sentences extracted from our keyword-based extractors can capture more keywords of reference summary. However, the extractor without CL always generates duplicate keywords. As shown in Table 8 and Table 6, KE$_{HIcl}$ performs better than KE$_{HI}$ as the sentences extracted by KE$_{HIcl}$ contain more keywords with less repetition. The reason is the loss function of KE$_{HIcl}$ considers the accuracy of the extracted keywords.

As the extractor is the first step of ext-abs framework, the output of the extractor is very important. As shown in Table 5, with the same abstractor, the ext-abs frameworks with keyword-based extractor get higher ROUGE scores. This shows that the keyword-based extractor can provide better input to abstractor.

Abstractor. We improve the abstractor by creating a new training set, pseudo summaries, which enhances the alignment between input of encoder and decoder. Table 4 shows that the models trained on set-level pseudo summaries generate more accurate summaries. The models with our designed special loss (PG$_{sl}$ and BART$_{sl}$) get higher ROUGE scores on sentence-level and set-level pseudo summaries, because the special loss considers the global information of the summary during parallel summarization. We do not apply special loss to the abstractor trained on summary-level pseudo summaries since the input of abstractor on summary-level pseudo summaries is complete and it cannot be processed in parallel. The summaries generated by pretrained models achieve higher ROUGE scores due to better document representations. As shown in Table 6, with the

Reorganized Reference Summary	
Set 1. federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** pointed at the **changing room**. state **authorities** found an overhead **camera** that the minister had spotted and determined that it was indeed able to take **photos** of customers. *Set 2.* four employees of the store have been **arrested**. if **convicted**, they could spend up to **three years** in jail.	
Extractive Summaries of Different Extractor	
HIBERT	new delhi , india -lrb- cnn -rrb- police have **arrested** four employees of a popular indian ethnic-wear chain after a minister spotted a security **camera** overlooking the **changing room** of one of its stores . federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** at the **changing room** , police said .
HIBERT$_{ad}$	*Set 1)* federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** pointed at the **changing room** , police said . *Set 2)* " *fabindia* is deeply concerned and shocked at this allegation , " the company said in a statement . " we are in the process of investigating this internally and will be cooperating fully with the police . "
KE$_{HI}$	*Set 1)* new delhi , india -lrb- cnn -rrb- police have **arrested** four employees of a popular indian ethnic-wear chain after a minister spotted a security **camera** overlooking the **changing room** of one of its stores . *Set 2)* federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** at the **changing room** , police said . *Set 3)* four employees of the store have been **arrested** , but its manager – a woman – was still at large saturday , said goa police superintendent kartik kashyap .
KE$_{HIcl}$	*Set 1)* federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa on friday when she discovered a surveillance **camera** pointed at the **changing room** . state **authorities** launched their investigation right after irani levied her accusation . *Set 2)* four employees of the store have been **arrested** . if **convicted**, they could spend up to **three years** in jail.
Abstractive Summaries of Different End-to-end Models	
BART	federal **education minister** smriti irani was visiting a **fabindia** outlet in the tourist resort state of goa . she discovered a surveillance camera pointed at the **changing room**. four employees of the store have been **arrested** , but the manager is still at large . the arrested staff have been charged with voyeurism and breach of privacy .
KE$_{HIcl}$-BART	*Set 1)* federal **education minister** smriti irani was visiting a **fabindia** outlet in goa . *Set 2)* **fabindia** is concerned and shocked at this allegation.
KE$_{HIcl}$-BART$_{sl}$	*Set 1)* police **arrested** four employees after a minister spotted a security camera . *Set 2)* federal **education minister** smriti irani was visiting a **fabindia** outlet in goa .
KE$_{HIcl}$-BART$_{sl}$-CRL	*Set 1)* federal **education minister** smriti irani was visiting a fabindia store the tourist resort state of goa. she discovered a **camera** at the **changing room**. **authoroities** discovered found it was able to take **photos** from the store 's **changing room**. *Set 2)* the four store workers could spend **three years** in jail if **convicted**.

Table 6: The extractive and abstractive summaries for the example in Table 1.

same extractor, the abstractor with special loss can generate more informative summaries with less redundancy.

Comprehensive Reinforcement Learning. As shown in Table 5, we combine our extractor and abstractor in different ways. Compared with BART, PG cannot abstract the extracted sentences effectively and achieves worse ROUGE scores than its connected extractors. Some ROUGE scores of summaries generated by keyword-based extractor with BART become lower because the less effective extractor brings more noise to the downstream abstractor. These results show that a good extractor is critical for ext-abs framework. KE_{HIcl}-$BART_{sl}$ has a lower R-L score on Web20 in Table 5 as the sentences in Web20 are very short. This causes that the overlapping of sentence-level longest common subsequence between reference and generated summary may be slightly lower when their R-1 and R-2 are higher.

We use RL to connect extractor and abstractor, which makes ext-abs framework an end-to-end trainable model. We observe the changes of different models after adding RL. As shown in Table 7, after adding sentence-level or summary-level reward, the ROUGE scores of the models on datasets become worse, which demonstrates that it is important to desige a suitable reward for ext-abs framework. The models trained on CRL achieve better ROUGE scores than that trained without RL, which denotes that our CRL can enhance extractor to select more accurate sentences. The ROUGE scores of extractor extended by RL are improved. The CRL bridges the backpropagation from abstractive summary to source document. So the ROUGE-based comprehensive rewards between generated summaries and reference summaries reflect the quality of extracted sentences and generated summaries which can guide the extractor to select correct sentences. The higher ROUGE scores of ext-abs with RL also show that the ext-abs model can benefit from a better extractor.

As shown in Table 8, our strongest model with CRL (KE$_{HIcl}$-BART$_{sl}$-CRL) outperforms the SOTA abstractive models on all datasets. As BART is the SOTA abstractive summarization model, the ROUGE scores of KE$_{HIcl}$-BART$_{sl}$-CRL are better than the BART but they are close. We take t-test to measure the difference of ROUGE scores between our model and BART. The p-values on ROUGE scores of the SOTA model BART and KE$_{HIcl}$-Abs$_{sl}$-CRL

Models	CNNDM			Web17			Web20			Wiki			DUC		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
KE$_{cl}$PG$_{sl}$	38.09	16.61	35.64	16.55	3.75	10.77	7.25	1.44	7.36	18.85	4.23	16.52	34.88	15.23	31.00
KE$_{cl}$PG$_{sl}$-RL$_{sen}$	38.12	15.60	34.45	16.37	3.63	10.32	7.09	1.42	7.45	12.19	3.02	11.21	35.43	16.37	32.12
KE$_{cl}$PG$_{sl}$-RL$_{sum}$	38.36	15.22	34.38	16.42	3.23	10.21	7.39	1.53	8.67	12.21	2.99	10.77	35.02	15.01	31.34
KE$_{cl}$PG$_{sl}$-CRL	39.66	19.69	36.61	18.01	4.12	13.91	12.01	2.54	11.54	21.73	6.46	19.67	38.07	17.64	34.22
KE$_{HIcl}$BART$_{sl}$	42.84	20.13	39.08	18.75	4.20	14.66	12.71	2.89	11.55	25.70	7.52	20.08	40.24	19.01	36.49
KE$_{HIcl}$BART$_{sl}$-RL$_{sen}$	42.17	19.50	33.12	18.46	4.01	14.29	12.02	2.66	11.32	25.75	7.64	21.48	35.22	18.01	32.10
KE$_{HIcl}$BART$_{sl}$-RL$_{sum}$	40.44	19.44	35.79	18.39	3.97	14.30	12.55	2.66	11.37	22.14	6.98	20.07	34.18	17.75	30.44
KE$_{HIcl}$BART$_{sl}$-CRL	**43.57**	**20.37**	**40.27**	**19.46**	**4.34**	**16.44**	**14.46**	**4.09**	**14.12**	**27.01**	**8.66**	**21.79**	**42.16**	**20.17**	**36.87**

Table 7: ROUGE scores of models with different RL.

Models	CNNDM			Web17			Web20			Wiki			DUC		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Extractive summarization															
lead-3 [26]	40.34	17.70	36.57	18.32	3.87	12.66	10.36	2.29	11.02	26.00	7.24	18.25	39.24	16.68	35.12
PN	37.04	16.57	33.81	16.21	2.03	10.22	8.79	1.44	8.23	18.71	4.03	15.11	35.70	15.38	33.12
HIBERT	42.37	19.95	38.83	19.32	4.10	14.89	13.84	3.03	11.93	26.34	7.53	19.68	40.31	18.43	36.18
KE$_{cl}$-CRL	41.37	19.11	38.74	18.54	4.03	13.27	11.20	1.90	10.07	20.45	5.63	16.51	38.26	18.11	34.27
KE$_{HIcl}$-CRL	42.92	20.10	39.68	19.38	4.22	14.73	14.30	3.16	12.04	26.02	7.79	19.18	40.55	18.45	36.35
Abstractive summarization															
PG	39.53	17.28	36.38	18.01	3.82	12.17	10.00	2.11	10.71	20.30	6.12	18.97	37.22	15.78	33.90
FastAbs	40.88	17.80	38.54	18.45	3.67	12.89	10.12	2.63	11.34	21.44	6.37	19.64	37.80	16.48	34.26
BART [4]	42.25	20.09	39.63	18.36	4.23	14.65	14.09	3.25	13.58	26.75	8.50	20.61	41.47	19.84	35.58
FastAbs$_{HB}$	42.71	20.08	39.69	19.27	4.20	14.25	14.23	3.74	13.26	26.21	8.47	20.90	41.54	19.15	35.60
KE$_{cl}$-PG$_{sl}$-CRL	39.66	19.69	36.61	18.01	4.12	13.91	12.01	2.54	11.54	21.73	6.46	19.67	38.07	17.64	34.22
KE$_{HIcl}$-BART$_{sl}$-CRL	$\underline{\textbf{43.57}}$	$\underline{\textbf{20.37}}$	$\underline{\textbf{40.27}}$	$\underline{\textbf{19.46}}$	$\underline{\textbf{4.34}}$	$\underline{\textbf{16.44}}$	**14.46**	**4.09**	$\underline{\textbf{14.12}}$	$\underline{\textbf{27.01}}$	**8.66**	$\underline{\textbf{21.79}}$	$\underline{\textbf{42.16}}$	$\underline{\textbf{20.17}}$	$\underline{\textbf{36.87}}$

Table 8: ROUGE scores of different end-to-end trainable models on datasets. The scores underlined are statistically significantly better than BART with $p < 0.05$ according to t-test.

of all datasets are less than 0.05, except for Web20. As the reference summary in Web20 is very short and abstract, it is difficult to extract pseudo summary aligned to the reference summary. The performance of models on Web20 is close and not good. This shows that the ext-abs framework with our approaches are effective. As test-only dataset, DUC testing on KE$_{HIcl}$-BART$_{sl}$-CRL gets highest ROUGE scores, which shows that our proposed model has a better generalization. The FastAbs$_{HB}$ in Table 8 takes HIBERT as extractor and BART as abstractor. The ROUGE scores of FastAbs$_{HB}$ are similar to BART due to its poor alignment of sentence-level training set and its extractor without keyword-based encoder. The best ROUGE scores of our models show that the abstractive models can be improved by locating the salience information. As shown in Table 6, the summary generated by our model with CRL contains more keywords and becomes more readable.

Speed and Memory. We take the speed and memory usage of CNNDM as example. As shown in Table 9, we evaluate our models on the speed and memory usage. Based on fune-tuning on the pretrained model or not, we compare our KE$_{cl}$-PG$_{sl}$-CRL with PG and KE$_{HIcl}$-BART$_{sl}$-CRL with BART, to be fair. KE$_{cl}$-PG-CRL is almost 7 times faster in total training time and occupies less memory than PG. We cannot fine-tune BART on GPU RTX-2080ti due to out-of-memory. We test BART based on the released pretrained summarization model. The KE$_{HIcl}$-BART$_{sl}$-CRL performs much better than BART on speed and memory usage. Both of our proposed models can decode summaries (word) in faster speed and occupy less memory.

Abstractive models have to encode long documents with attention model looking at all encoded tokens at each time step, which causes low speed and large memmory usage. As a pointer network, our extractor is faster than most abstractive models. Our models first extract sentence sets from a source and then input them to abstractor. These inputs can be decoded in parallel, which speed up the model. The average length of inputs is shortened from 780 to 100, which reduces the memory usage. The FastAbs and FastAbs$_{HB}$ are faster than our models because they train and test models on sentence-level pseudo summaries which are shorter than our set-level pseudo summaries. However, the difference is not significant. This is because that the different matching heuristics may extract different sentences for the same sentence in the reference summaries. Besides, as shown in Table 8, the ROUGE scores of KE$_{cl}$-PG$_{sl}$-CRL and KE$_{HIcl}$-BART$_{sl}$-CRL are better than than FastAbs and FastAbs$_{HB}$.

3.3.3 *Human Evaluation.*

We compare the readability and keyword coverage of our best model (KE$_{HIcl}$-BART$_{sl}$-CRL) and the SOTA model. As shown in Table 10, our model get the highest readability score and keyword coverage score, which means that our model can generate more informative summaries with more keywords. As shown in Table 6, our model generates more readable

Models	Training			Testing			
	T (h)	Epoch/h	M (G)	T (h)	summaries/s	tokens/s	M (G)
PG	40.24	0.29	6.72	16.13	0.60	23.74	2.02
FastAbs	6.71	1.74	3.26	1.34	2.18	76.3	0.91
KE$_{cl}$-PG-CRL	7.04	1.57	3.42	1.60	1.99	69.82	0.94
BART	-	-	OOM	8.30	0.38	35.31	3.67
FastAbs$_{HB}$	13.64	0.44	8.93	3.41	0.81	68.85	2.37
KE$_{HIcl}$-BART$_{sl}$-CRL	16.61	0.30	9.74	4.63	0.72	61.2	2.55

Table 9: Total time (T), speed and memory usage (M) of models during training and testing of CNNDM dataset on RTX-2080ti.

summaries. This means that our model improve BART by keyword-based extractor capturing salient and aligned information.

Models	CNNDM		Web17		Web20		Wiki		DUC	
	Read	KC	Read	KC	Read	KC	Read	KC	Read	KC
BART	0.74	0.36	0.80	0.31	0.80	0.20	0.79	0.25	0.73	0.33
Ours	**0.81**	**0.45**	**0.86**	**0.39**	**0.91**	**0.28**	**0.85**	**0.30**	**0.88**	**0.37**

Table 10: The Readability(Read) and Keyword Coverage(KC) of generated summaries.

4 RELATED WORK

Related work on extractor-abstractor framework and pretrained models for summarization are introduced as follows.

4.1 Extractor-Abstractor Framework

In this paper, we adopt the extractor-abstractor (ext-abs) framework, which has been a popular method for abstractive summarization recently. Unlike the end-to-end models [13, 17, 21, 24, 26] in abstractive summarization, the ext-abs framework trains two enc-dec models, extractor and abstractor. The extractor captures salient content (pseudo summary) of source document, where the pseudo summary can be either sentence-level [5, 11, 29] or summary-level [1, 27], and then abstractor paraphrases the salient content to generate a summary. In this paper, we present a set-level matching heuristics to construct the pseudo summaries, better aligned to reference summaries.

Extractive models adopt hierarchical neural network as encoder and pointer network as decoder [6, 20]. It is extended with variant models, such as reinforcement learning [22] and joint scoring [37]. As transformer preforms excellent on language model, Liu et al. [16] and Zhang et al. [34] apply pretrained transformers to extractive summarization. Zhou [38] and Li et al. [14] have shown that keywords play an important role in summarization. So we enhanced the extractive models with an additional keyword encoder to get better alignments between pseudo summaries and reference summaries.

Abstractive models are based on sequence-to-sequence learning [2, 28]. The pointer-generator networks [26] consisting of copy mechanism and coverage model are the most popular baseline in abstractive summarization. The pretrained transformer language models have success in natural language processing. Through fine-tuning the pretrained models on summarization task, the quality of generated summaries are improved [34, 36].

The reinforcement learning (RL) is always used to connect the extractor and abstractor together, which makes an end-to-end trainable model. Chen [5] and Bae et al. [1] encourage extractor to select sentences with high ROUGE scores by RL. Sharma et al. [27] propose an entity-driven encoder and utilize RL with coherent rewards to make abstractor generate readable summaries. Different from previous end-to-end evaluation rewards, we propose a comprehensive reward, taking the intermediate extracted pseudo summary and set-level abstactive summaries into consideration.

4.2 Pretrained Models for Summarization

The pretrained transformer language models have success in natural language understanding (NLU) and natural language generation (NLG). NLU models are pretrained on unidirectional and bidirectional prediction. GPT [25] employs a unidirectional transformer [30] to predict the sequence. ELMo [23] learns two unidirectional language models of forward and backward. BERT [7] uses a bidirectional transformer encoder to predict the masked words. NLG models pretrain on sequence-to-sequence (seq2seq) models. UniLM [8] is a multi-layer transformer network, including unidirctional, bidirectional and seq2seq language model. BART [13] takes combines bidirectional transformer encoder and auto-regressive transformer decoder. ProphetNet [24] trains on the transformer seq2seq model and takes future n-gram prediction as self-supervised. Through fine-tuning the pretrained models or representations on summarization task, the quality of generated summaries can be improved [34–36]. PEGASUS [33] is a new, pretrained model for text summarization, which uses self-supervised objective Gap Sentences Generation to train a transformer seq2seq model. Different from previous pretrained models, it masks sentences rather than smaller continuous text spans. We choose BART which has achieve the SOTA results on summarization tasks as a basic component. Since these pretrained model are encoder-decoder models and can be used in the same context, it has the potential to substitute BART in our ext-abs framework and enjoy similar boost in accuracy and speed.

5 CONCLUSION

To enhance the alignment between documents and summaries in ext-abs framework, we propose a set-level matching heuristics to extract pseudo summary as training set. We introduce a new ext-abs framework that use comprehensive RL to connect keyword-based extractor and abstractor with pretrained models together. The result shows that our model outperforms the SOTA methods on variant datasets, such as news and web text. Besides, our models are faster and occupy less memory than previous pretrained models during

[4] Test BART on released model *bart.larg.cnn* https://github.com/pytorch/fairseq/tree/master/examples/bart.

training and testing. In the future, we will improve the extractor and strengthen the relation between extractor and abstractor.

ACKNOWLEDGMENTS

This work was partly supported by the SJTU-CMBCC Joint Research Scheme and SJTU Medicine-Engineering Cross-disciplinary Research Scheme.

REFERENCES

[1] Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. 2019. Summary Level Training of Sentence Rewriting for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. 10–20.

[2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[3] Asli Çelikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep Communicating Agents for Abstractive Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*.

[4] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive Snippet Generation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. 1309–1319.

[5] Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

[6] Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. [n.d.]. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 4171–4186.

[8] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. 13042–13054.

[9] Min Gui, Zhengkun Zhang, Zhenglu Yang, Yanhui Gu, and Guandong Xu. 2018. An Effective Joint Framework for Document Summarization. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*. 121–122.

[10] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*.

[11] 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)*. 132–141.

[12] Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305* (2018).

[13] 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, ACL 2020*. 7871–7880.

[14] Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana, 55–60.

[15] Chin-Yew Lin. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out* (2004).

[16] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. 3728–3738.

[17] Yizhu Liu, Zhiyi Luo, and Kenny Q. Zhu. 2018. Controlling Length in Abstractive Summarization Using a Convolutional Neural Network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[18] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[19] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004*. 404–411.

[20] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 3075–3081.

[21] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar 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, CoNLL 2016*.

[22] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 1747–1759.

[23] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, Volume 1 (Long Papers)*. 2227–2237.

[24] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. (2020), 2401–2410.

[25] Alec Radford. 2018. Improving Language Understanding by Generative Pre-Training.

[26] 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, ACL 2017*.

[27] Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An Entity-Driven Framework for Abstractive Summarization. (2019), 3278–3289.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*.

[29] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive Document Summarization with a Graph-Based Attentional Neural Model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 5998–6008.

[31] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*. 2692–2700.

[32] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Copenhagen, Denmark, 59–63.

[33] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event (Proceedings of Machine Learning Research)*, Vol. 119. 11328–11339.

[34] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document Level Pre-training of Hierarchical Bidirectional Transformers for Document Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*. 5059–5069.

[35] 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*. 6197–6208.

[36] 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 Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*. 1049–1058.

[37] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 654–663.

[38] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective Encoding for Abstractive Sentence Summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*.