Exploring Explainable Selection to Control Abstract Generation

Wang Haonan\(^1\), Gao Yang\(^{1,3*}\), Bai Yu\(^1\), Mirella Lapata\(^2\) and Huang Heyan\(^{1,3}\)

\(^1\) School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, 100081
\(^2\) Institute for Language, Cognition and Computation, School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB
\(^3\) Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, gyang@bit.edu.cn, mlap@inf.ed.ac.uk

Abstract
It is a big challenge to model long-range input for document summarization. In this paper, we target using a select and generate paradigm to enhance the capability of selecting explainable contents (i.e., interpret the selection given its semantics, novelty, relevance) and then guiding to control the abstract generation. Specifically, a new designed pair-wise extractor is proposed to capture the sentence pair interactions and its centrality. Furthermore, the generator is hybrid with the selected content and is jointly integrated with a pointer distribution that is derived from a sentence deployment’s attention. The abstract generation can be controlled by an explainable mask matrix that determines to what extent the content can be included in the summary. Encoders are adaptable with both Transformer-based and BERT-based configurations. Overall, both results based on ROUGE metrics and human evaluation gain outperformance over several state-of-the-art models on two benchmark CNN/DailyMail and NYT datasets.

1 Introduction

Recent neural networks have shown overwhelming success in both extractive and abstractive summarization. Extractive paradigm (Nallapati et al., 2017; Zheng and Lapata, 2019) tends to generate readable and relevant summaries as its selective manner from the original input. Abstractive paradigm (See et al., 2017; Celikyilmaz et al., 2018; Wang et al., 2019) successfully generates new abstract sequences by encoder-decoder models that are derived from machine translation. Though their success, it remains a challenging task for modeling long-range context for document summarization.

Currently, two state-of-the-art approaches are raised to solve this problem. Recent work on pre-trained language models, such as ELMo (Peters et al., 2018), OpenAI GPT (Radford et al., 2018) and BERT (Devlin et al., 2018), have shown their remarkable performances on long-range contextual learning of representing texts and profited various NLP tasks, such as QA (Xu et al., 2019) and summarization (Liu and Lapata, 2019; Zhang et al., 2019). The other idea to digest the long document is using a select and generate framework, which selects salient sentences by an extractor then followed up with an abstractor to generate an abstract summary. The recent progress on the hybrid paradigm is designed by a two-stage pipeline (Chen and Bansal, 2018; Sharma et al., 2019) or an end-to-end learning approach (Hsu et al., 2018; Shen et al., 2019; Gehrmann et al., 2018). The most appealing advantage is able to explicitly obtain desirable content of the sources, such as entity-aware selection (Sharma et al., 2019) and word selection through latent switch variables (Gehrmann et al., 2018; Shen et al., 2019) and so on.

The hybrid summarization further refines the effect of informative and relevant content extraction (extractor) as well as aggregates into summary in line with linguistic expression (abstractor). While training a sentence-level extractor, much work encodes sentences as hidden representations to predict whether they are labeled ground-truth. However, the current extractor can not well explain the rationale of classifying the sentences as extracted summaries. An in-depth investigation of the rich inter-sentence relations is needed, i.e., identifying the semantic meaning, inferring the relations between two pieces of sentences, identifying whether the sentence is relevant to the document and whether it is novel to contribute to the summaries. Regarding sequence generation, pointer-generator is widely adopted by the abstractor (See et al., 2017; Wang et al., 2019; Chen and Bansal, 2018; Celikyilmaz et al., 2018). However, due to scattered spans of information diversity when
Gold Summary:
• Federal education minister Smriti Irani visited a Fabindia store in Goa, saw cameras.
• Authorities discovered the cameras could capture photos from the store’s changing room.
• The four store workers arrested could spend 3 years each in prison if convicted.

Abstractive Summary:
• Four employees of the store have been arrested, but its manager – herself a woman.
• State authorities ties launched their investigation right after Irani levied her accusation.
• The arrested staff have been charged with voyeurism and breach of privacy.

Table 1: The summary with red highlight is not relevant to any of the gold summaries. The generation drift occurs after the word “authorities” maybe because this span is in the source document but not in summary.

summarizing the long document, it is inevitable to include diffuse content drift with some unnecessary spans of generated summaries. For example, in Table 1, the abstractive generation can be thoroughly irrelevant to the gold summary, which is called diverse (or drift) generation. More importantly, the diversity is hard to control.

To fulfill the aforementioned demands, in this paper, we propose an Explainable Selection module to Control the generation of Abstractive summary, named as ESCA. Specifically, we construct an interaction matrix that explicitly points out the decision at each sentence depends on the content richness of the sentence and paired similarity, its relevance concerning the document and its novelty regards to accumulated summary representation. Then a new pair-wise ranking extractor is established to favor the complex relations within each sentence pair and its influence as the summary. To avoid generation diversity under the pointer-generator framework, a proposed sentence deployment mechanism encourages the sentence-level extractive-preferred content delivering to the word sequence generation. As such, the extractor and abstractor are seamlessly hybrid with the deploying based pointer as an end-to-end manner. Furthermore, the content selection can be controlled under different explainable masked matrices and, consequently, influence the coverage of abstract generation. Our contributions include: 1) proposing an explainable content selection module for summarization; 2) controlling the summary generation based on the desirable selection; 3) achieving the state-of-the-art performance on document summarization in a controllable training strategy.

2 Related Work

Abstractive summarization supposedly digests and understands the source content and generates new order of words and shorter sentences. Headline generation is typically a subtask of abstractive summarization and the seq2seq models with attention have substantially succeeded in achieving it (Nallapati et al., 2016; Zhou et al., 2017; Shen et al., 2019; Wang et al., 2019). But, it remains a big challenge to deal with the long-range context of document summarization in the abstractive manner.

The reasons are quite complex. One is derived from semantic representations which determine whether the model is able to accurately understand the document. This has been intensively studied in the community of pre-trained models (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018; Liu and Lapata, 2019; Zhang et al., 2019), which is not the focus of this paper. Another reason is the current language model-based generation is often “black-box” network and can not explicitly select the explainable content from the long documents.

Therefore, selecting desirable and meaningful content has been an urging task. Pointer-generator (Vaswani et al., 2017; See et al., 2017) was developed to use attention as a pointer to jointly determine the probability of selecting a word (Wang et al., 2019; Çelikyilmaz et al., 2018) or selecting a sentence (Chen and Bansal, 2018; Sharma et al., 2019), conditioned on contextual information. At word-based selection level, Zhou et al. (2017) used soft gating on source document. Gehrmann et al. (2018) pre-trained a sequential word selector to constrain the attention from the source. Hsu et al. (2018) updated the word attention by considering the sentence level of importance. To select salient sentences, a graph-based attention mechanism was used to enhance the salience discovery in abstractive model (Tan et al., 2017). Similarly, Li et al. (2018) achieved this by an information selection layer consisting components of global filtering and sentence selection. You et al. (2019) further modeled the salience attention by introducing a Gaussian focal bias to enhance the informative selection.

Regarding generation process, text can be summarised in diverse target sequences with different
semantics (Cho et al., 2019). To tackle this issue, Shen et al. (2019) utilised a decoupling content selection to allow fine-grained control over the generation. In our model, we inherit the benefits of pointer-generator for selection and generation, and further investigate explainable selection to control the desirable summary generation.

3 Background: Transformer-based Encoder-Decoder Framework

Encoder-decoder framework consists of an encoder and an attention-equipped decoder. Suppose that the sequential input \( X = \{x_1, \ldots, x_j, \ldots, x_n\} \) is a sequence of \( n \) number of words, and \( j \) is the index of the input. The shorter (i.e., summarized) sequence of output is denoted as \( Y = \{y_1, \ldots, y_t, \ldots, y_m\} \) with number of \( m \) words, and \( t \) indicates a position.

Encoder The basic structure is based on Transformer. It is composed of a stack of \( N \) identical layers, and each layer has two sub-layers:

\[
\begin{align*}
    h_1^l &= \text{LAYERNORM}(h_2^{l-1} + \text{MULH}_{\text{att}}(h_2^{l-1})) \\
    h_2^l &= \text{LAYERNORM}(h_1^l + \text{FFN}(h_1^l))
\end{align*}
\]

where the first is the self-attention sub-layer \( h_1^l \) and the second is the feed-forward sub-layer \( h_2^l \) of layer depth \( l \). LAYERNORM is the layer normalization, and the multi-head operation is

\[
\text{MULH}_{\text{att}}(h_2^{l-1}) = \text{CONCAT}(H_1, \ldots, H_h)W_l
\]

where \( H_i \) is self-attention operation of layer \( l \) at \( i \)th head. Finally, let \( Z_e \) be the sequence output of the encoder. We also employ pre-trained BERT encoder (Devlin et al., 2018) and the implementing details are included in Section 5.

Decoder For both Transformer-based and BERT-based configurations, we employ a similar decoder equipped with attentions. The decoder has a similar stacked of \( N \) identical layers. In addition to the two sub-layers that are similar with the encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Then, we calculate an attention between decoder position vector \( s_t \) and encoder sequence output \( Z_e \) at each source position. Then, the attention on source input at decoder position \( t \) is calculated as

\[
\alpha_t = \text{softmax}\left(\frac{QK^T}{\sqrt{d_m}}\right)
\]

where \( Q \) is \( s_t \) and \( K \) is \( Z_e \). Then the context vector at decoding position \( t \) is \( h^*_t = \alpha_t Z_e \). The decoder generates a target summary from a vocabulary distribution \( P_{\text{vocab}}(w) \) through the following process:

\[
P_{\text{vocab}}(w) = P(y_t|y_{<t}, x; \theta) = \text{softmax}(W_2(W_1[s_t, h^*_t] + b_1) + b_2)
\]

4 The Proposed Model

Our model is an end-to-end hybrid summarization and the overall framework is demonstrated in Figure 1. It comprises of (1) a pair-wise extractor that is leveraged by sentence interaction matrix and its centrality (§4.1); (2) the abstract generation is guided by the sentence deployed attention as a pointer to hybrid with a Pointer Generator (PG) abstracter (§4.2). (3) Finally, we introduce how the explainable content selection is controlled (§4.3) and impact the generation training (§4.4).

4.1 The Extractor

The encoder of the extractor is achieved by both Transformer and BERT, respectively, to showcase
the flexibility. Specifically, 1) the Transformer-based sentence and document are represented by hierarchical Transformer architecture (Vaswani et al., 2017). 2) Similar to (Liu and Lapata, 2019), several inter-sentence Transformer layers are stacked on top of BERT output.

4.1.1 Sentence Interaction Matrix

Complex relations are existed in each sentence pair that have mentioned in (Nallapati et al., 2017), such as content richness, novelty and relevance to document. We construct a sentence interaction matrix, \(Q^s\) (\(s\) is the number of sentences), leveraged by the above properties and additional with sentence pair similarity. Noticed that the “interaction” has direction, since the contribution induced by two sentences’ relations to their respective importance as a summary can be unequal, which is suggested by (Zheng and Lapata, 2019). The mutual influence of the sentence pair is differently directional, which is also grounded in theories of discourse structure (Mann and Thompson, 1988). The directional influence from sentence \(j\) to sentence \(i\) is denoted as \(q_{ij}\) in the interaction matrix \(Q^s\),

\[
q_{ij}(h_i, s_i, h_j, d) = \sigma(W_c h_i + h_j^T W_s h_j - h_i^T W_n \tanh(nov_i) + h_i^T W_r d + b_{\text{matrix}})
\]

where \(\sigma\) is a sigmoid function, the parameters \(W_c, W_s, W_n, W_r\) are trainable and \(b_{\text{matrix}}\) is the bias. \(h_i\) is the representation of sentence \(i\) and \(\text{nov}_i\) is the accumulated summary representation with respect to the current sentence \(i\) that \(\text{nov}_i = \frac{1}{s} \sum_{t=1}^{i-1} \sum_{k=1}^{s} h_t \times q_{tk}\).

4.1.2 Sentence Centrality Calculation

The interaction matrix provides the mutual influence concerning each sentence pair that assists in implying the sentence importance overall. There are several approaches to compute the sentence centrality, such as graph-based TextRank (Mihalcea and Tarau, 2004) and LexRank summarization models. Tan et al. (2017) have also drawn the similar idea to determine the sentence salience via the graph-based attention. With our end-to-end settings, the interaction matrix \(Q^s\) is directly transformed into the sentence distribution and is converted into a centrality vector:

\[
c = Q^s W_q
\]

where \(c = [c_1, \cdots, c_s] \in \mathcal{R}^s\) denotes the sentence centrality. \(W_q \in \mathcal{R}^{s \times 1}\). In the experiment, we truncate the number of sentences \(s\) in a document with the maximum of 50.

4.1.3 Pair-wise Learning Extractor

The extractor can be framed as a classification problem. Nallapati et al. (2017); Hsu et al. (2018); Liu and Lapata (2019) all used a point-wise ranking approach in which sentences are encoded as hidden representations. Then a binary classifier was taken over these representations to predict whether they are the summaries or not. However, the point-wise learning is not yet accurate enough since it can not reflect the interaction between sentences. Instead, we introduce a new pair-wise loss function that facilitates the extractor to decide summary classification, which is supported by inter-sentence labels. First, each sentence is labeled as described in Appendix A. Then, the inter-sentence label for each sentence pair \(\hat{P}_{ij}\) is marked with \(\{0, 1\}\). It is defined as 1 if sentence \(i\) is labeled as summary but sentence \(j\) is not, 0 if \(j\) is summary while \(i\) is not. To adapt our supervised system, we calculate the predicted co-occurrence probability \(r_{ij}\) of sentence \(i\) and sentence \(j\) as \(\sigma(c_i - c_j)\). Then, the loss function can be defined as

\[
L_{\text{ext}} = -\sum_{i=1}^{m} (\hat{P}_{ij} \log r_{ij} + (1 - \hat{P}_{ij}) \log(1 - r_{ij}))
\]

4.2 The Abstractor

In our proposed framework, the abstractor is achieved by a pointer generator (PG) network. The basic PG network contains two sub-modules, one is the pointer network and the other is the generation network. These two sub-modules jointly determine the probabilities of the words in the final generated summary. The generation employs the classic encoder-decoder network as described in Section 3. Our proposed model essentially leverages this configuration that integrates a new sentence deployed pointer, introducing the selected content flow into the generation network in the hybrid framework.

4.2.1 Sentence Deployed Pointer Generator

The pointer network uses attention as a pointer to select segments of the input as outputs (Vinyals
As such, a pointer network is a suitable mechanism for extracting salient information, while remaining flexible enough to interface with a language model for generating an abstractive summarization (See et al., 2017).

For our pointer network, the selected segments of input can be updated by the extractor with respect to their extractive-oriented centrality of each sentence. To influence the sequence generation, sentence importance is needed deploying to the word level. The deployment should determine how much information flow delivers to the word-level generation, at the same time considering the importance of the derived sentence. Then, the pointer can seamlessly connect the extractor and the abstractor in the hybrid approach.

**Hybrid Connector** The pointer is taken by the attention distribution that will be updated by our proposed hybrid connector. The hybrid is achieved by the sentence deployment attention mechanism which controls the generation focusing on what the selected contents explicitly convey,

\[
\hat{c}_n = \frac{\alpha_n^t(1 + p_{sen}c_{mn})}{\sum \alpha_n^t(1 + p_{sen}c_{mn})} 
\]

\[p_{sen} = \sigma (W_{sel}E_{sel}^t + b_{sen})\]

where \(c_{mn}\) denotes the score of sentence \(m\) which the word \(n\) belongs to. \(E_{sel}^t\) is the representation of the selected sentence \(m\) at decoding step \(t\). \(p_{sen}\) decides how much degree of a sentence influence can be introduced. \(W_{sel}\) is a trainable parameter. Additionally, the generation probability \(p_{gen}\) is modified as

\[
p_{gen} = \sigma (W_h^t h_s^t + W_s s_t + b_{gen})
\]

The pointer is taken based on the updated attention distribution \(\hat{c}_t\) over the source text, and the final output distribution is combined as

\[
P_{final}(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{j:w_j=w} \alpha_{t,j}
\]

\[4.3\] Controllability

Since the inter-sentence relations are captured in the interaction matrices introduced in Section 4.1.1, the overall centrality is capable to reflect different aspects of explainable summaries. To explore the explainability, we manipulate the selection through several mask matrices \(M\) that are calculated by a set of controllable thresholds (i.e., novelty \(\epsilon_n\), relevance \(\epsilon_r\)) with respect to the specific value of each criteria. The interaction matrix is controlled by the mask matrix as

\[
\hat{Q}^s = Q^s \odot M, \quad \text{where} \quad M_{ij} = \begin{cases} 1, & \text{val} \geq \epsilon \\ 0, & \text{val} < \epsilon \end{cases}
\]

where \(\odot\) is element-wise multiplication, and \(\text{val}\) can be \(\sigma(\text{novelty})\) or \(\sigma(\text{relevance})\) that are from Eq. (5). We use different mask matrices \(M\) to control what explainable content it focuses. As the document graph is then reshaped along with different mask matrices and, subsequently, the summary selection is changed because of the revised centrality. Therefore, the generation training is then fine-tuned because the controllable effect also involves the sentence deployed attention distribution of the abstractor.

\[4.4\] Training and Content Coverage

Finally, the basic generator objective is derived by maximizing the likelihood training for the PG generation, given a reference summary \(y^* = \{y_1^*, y_2^*, \cdots, y_m^*\}\) for document \(x\). The training objective is to minimize the negative log-likelihood of the target word sequence:

\[
\mathcal{L}_{abs} = - \sum_{t=1}^{m'} \log P_{final}(y_t^* | y_1^*, \cdots, y_{t-1}^*, x)
\]

\[\text{Coverage for generation}\]

Sequential generation models often suffer local optimization, generating the same words or phrases. Coverage mechanism Hsu et al. (2018) and See et al. (2017) prevents this repetition by calculating the coverage vector \(\text{cov}_t = \sum_{i=1}^{t-1} \alpha_i\) in each decoder step \(t\), indicating so far how much degree of coverage has been received. In our model, as we have constrained the coverage in selected content, the coverage loss \(\mathcal{L}_{cov}\) is further constructed as

\[
\mathcal{L}_{cov} = \sum_{t=1}^{m'} \min(\alpha_t, \beta, \text{cov}_t), \quad \text{where} \quad \beta \text{ is a one-hot mask vector that indicates which words are selected from the controlled extractor. This loss directly penalizes the repetition particularly within the selected range. Overall, the abstractor learning objective is combined as} \quad \mathcal{L} = \mathcal{L}_{ext} + \mathcal{L}_{abs} + \mathcal{L}_{cov}.
\]

\[5\] Experiments

In this section we describe the summarization datasets, present our experimental setup, implementation details, evaluation method, and analyze
Datasets

We evaluated our models and baselines on two benchmark datasets, namely the CNN/DailyMail news (Hermann et al., 2015), the New York Annotated Corpus (NYT) (Sandhaus, 2008).

The CNN/DailyMail dataset contained news articles and associated with highlights as summaries. We followed the standard splits for training, validation and testing by 90,266/1220/1093 of CNN dataset and 196,961/12,148/10,397 of DailyMail dataset. We didn’t anonymize entities. The datasets were pre-processed following See et al. (2017). The NYT dataset contained 110,540 articles with abstractive summaries which were divided into 100,834 training and 9,706 testing set, following Durrett et al. (2016). We also filtered the raw dataset by eliminating the documents with summaries that are shorter than 50 words. The filtered testing set, named as NTY50, includes 3,421 examples. As mentioned in (Liu and Lapata, 2019), the NYT test set contained longer and more elaborate summaries than the CNN/DailyMail whose summaries were largely extractive biased and concentrated at the beginning of documents. All above sentences were split with the Stanford CoreNLP toolkit (Manning et al., 2014).

Evaluation Metrics: We used ROUGE (Lin, 2004) as the evaluation metric, which measures the quality of a summary by computing the overlapping lexical elements between the candidate summary and a reference summary. Following previous practice, we assessed R-1 (unigram), R-2 (bigram) and R-L (longest common subsequence - LCS).

Comparative Models All the following state-of-the-art models followed the “select and generate” style (BERTSUMabs excluded), as Section 2 introduced, which were worth to compare with. Concretely, PG was BiGRU-based seq2seq model integrated with pointer network, and PG+Coverage has additional coverage mechanism (See et al., 2017). Graph-attention (Tan et al., 2017) employed a graph ranking-based attention to identify salient sentences. Select-Reinforce (Chen and Bansal, 2018) reinforced to extract important sentences by a reward function from the summary rewrite evaluation metric. Inconsistency-Loss (Hsu et al., 2018) was a loss function to use sentence-level attention to modulate the word-level attention for generation. Bottom-Up (Gehrmann et al., 2018) used extractive encoder as content selector to constrain word attention for the abstractive summarization. ExplicitSelection (Li et al., 2018) extended the vanilla seq2seq model with a soft information selection layer to control information flow. SENeca (Sharma et al., 2019) selected entity-aware sentences and then connects with reinforcement learning-based abstract generation. BERTSUMabs (Liu and Lapata, 2019) was BERT-based abstractive summarization.

5.1 Details of Our Models

5.1.1 Transformer-based ESCA

followed the settings from Liu and Lapata (2019). Specifically, we inserted $[CLS]$ tokens at the start of each sentence, and also used interval segment embeddings $[E_A]$ or $[E_B]$ to distinguish multiple sentences in a document. Then, the sentence embedding was learned by the $[CLS]$. Position embeddings in the BERT model had 512 length limit. We used a standard ‘bert-base-uncased’ version of BERT. Both source and target tokens were tokenized with the BERT’s subwords. The transformer layers hidden size was 768 and all the feed forward layers had 2048 hidden units. For extractor, we used one transformer layer to acquire the sentence representation with 8 head and dropout of 0.1. We used trigram block trick (Paulus et al., 2017) to prevent duplicates instead of coverage mechanism. We trained the abstractor of 15k iterations for NYT.

5.1.2 BERT-based ESCA

was trained with a 6-layer transformer. The hidden size was set as 512 while the feed forward dimension was 1024. 8 heads were set for multi-head attention. We used dropout with probability 0.2 before linear layers. The learning rate for PG was 0.15 with encoder’s batch size as 32 and decoder’s beam size was 4. The abstractor processed the input by truncating source documents to 400 tokens for CNN/DailyMail and 800 for NYT, and the target summaries to 100 tokens in training and validation sets. Both the learning rate of extractor and abstractor was 0.15. At the testing phrase, we limited the length of the summary to 120. We trained the model with early stopping and length penalty on the validation set.

1Our code is available at https://github.com/WoodenWhite/Centrality_Pre_Summ

2https://github.com/google-research/bert
| Models          | R-1    | R-2    | R-L    |
|-----------------|--------|--------|--------|
| PG              | 36.44  | 15.66  | 33.42  |
| PG+Coverage     | 39.53  | 17.28  | 36.38  |
| Graph-attention †| 38.1   | 13.9   | 34.0   |
| Select-Reinforce †| 40.88  | 17.80  | 35.84  |
| Inconsistency-Loss †| 40.68  | 17.97  | 37.13  |
| Bottom-Up †     | 41.22  | 18.68  | 38.34  |
| Explicit-Select †| 41.54  | 18.18  | 36.47  |
| SENICA †        | 41.52  | 18.36  | 38.09  |
| BERTSUMabs †    | 41.72  | 19.39  | 38.76  |
| Ours            | 41.65  | 18.89  | 37.94  |
| ESCA-Transformer| 42.12  | 19.51  | 38.70  |
| ESCA-BERT       |        |        |        |

Table 2: ROUGE F1 scores of abstractive summarization on the CNN/DailyMail dataset. The results with † mark are taken from the corresponding papers.

| Models          | R-1    | R-2    | R-L    |
|-----------------|--------|--------|--------|
| PG              | 42.47  | 25.61  | 36.10  |
| PG+Coverage     | 43.71  | 26.40  | 37.79  |
| Bottom-Up †     | 47.38  | 31.23  | 41.81  |
| SENICA †        | 47.94  | 31.77  | 44.34  |
| BERTSUMabs †    | 48.92  | 30.84  | 45.41  |
| Ours            | 45.17  | 27.61  | 41.52  |
| ESCA-Transformer| 49.1   | 34.5   | 46.1   |

Table 3: ROUGE recall (Durrett et al., 2016) scores of abstractive summarization on the NYT dataset. The results with † mark are taken from the corresponding papers.

The overall results were presented in Table 2 and Table 3. We observed that our ESCA-BERT outperformed all the strong state-of-the-art models on both datasets in all metrics except for R-L on the CNN/DailyMail dataset. Interestingly, the ESCA-Transformer performed weakly on the NYT dataset that is more abstractive than the CNN/DailyMail (Liu and Lapata, 2019). It further enunciates the transformer structure may not be suitable for sequence generation unless the representation is augmented by the pre-trained language model.

**Pair-wise vs. Point-wise Extractor** To investigate the effectiveness of pair-wise ranking strategy, we compared it with the counterpart point-wise ranking. The probability of a sentence was calculated by $\sigma(c_i)$ where $c_i$ is from Eq. (6). Then, the point-wise ranking loss was computed by the cross entropy of predicted score with gold label. In Table 4, we solely compared the extraction performance on the CNN/DailyMail dataset. Apparently, our model with pair-wise ranking is largely superior to point-wise extractor over ROUGE scores, with the relative improvement ranging from 11.0% to 11.55%. Since our study more cared about global sentence distributions over the documents, we selected top 6 sentences for the comparisons. This indicates the overall distribution of the pair-wise extractor is more likely close to the gold summary. It verifies that the pair-wise ranking has a noticeable effect on the summary extractor considering the pair-wise interaction of sentences.

**Interaction Matrix vs. Self-attention Layer** According to Vaswani et al. (2017), the inter-sentence relations can be also captured by multiple stacked self-attention layers. Therefore, we replaced it with a 2-layer self-attention to build a counterpart variant of the ESCA’s extractor, namely Extractor_{self-attention}. The comparing results are reported in Table 5. Based on the ROUGE scores, these two models have not any significant difference except that self-attention may not be able to explicitly explain the selection as the way the interaction matrix does.
5.3 Controllability

To explore the explainable selection with respect to relevance and novelty that mentioned in Eq.(5), we manually set thresholds to construct masking matrices that in Eq.(11). Performance was evaluated as shown in Table 6. We evaluated the controlled performance by ROUGE recall because it shows whether additional relevant summaries can be included under the deterministic controls. Additionally, there is always a trade-off between controllability and summary performance since the gold summary was automatically constructed with bias and not based on these controllability properties.

To verify the effectiveness of controlling novelty and relevance, we further conducted the following criteria ranking human evaluation (§5.4). Samples of the abstract summaries were also demonstrated in Appendix C for further inspection.

5.4 Human Evaluation

QA evaluation In addition to ROUGE metrics automatic evaluation, we conducted two separate human evaluations. The first one is to assess the degree to which our model retains key information from the document following a question answering (QA) paradigm which has been previously used to evaluate summary quality and document compression (Clarke and Lapata, 2010; Narayan et al., 2018). Following the similar setting, a set of questions based on the gold summary under the assumption that it highlighted the most important document content. The QA example is listed in Appendix B. Then, we examined whether participants can correctly answer these questions by reading only summaries without access to source articles. The more questions can be correctly answered, the better summarizing capability of the model. Example documents were selected from CNN/DailyMail and associated with corresponding questions (i.e., three questions for each document) (Narayan et al., 2018). We adopted the same scoring mechanism used in Clarke and Lapata (2010). Let the evaluation score be marked with 1 if the answer was correct, 0.5 if it was partially correct, and 0 otherwise. Each summary was elicited by 3 responses.

Criteria ranking We conducted another evaluation study that assessed the quality of summaries produced by different systems, in terms of the Best-Worst Scaling approach. Concretely, we presented the summary outcomes from comparative systems and the original documents (the documents are similar with the ones used in the QA evaluation) to the participants and asked them to select the best and worst summaries according to the specific criteria, such as Informative, Novelty and Relevance. The score of each system was computed as the percentage of times it was chosen as the best minus the times it was selected as the worst. Thus, the scores are ranging from -1 (worst) to 1 (best). We conducted two groups of ranking evaluations to testify the overall performance of the ESCA and the effectiveness of its controllability, respectively.

Based on QA evaluation in Table 7, it is observed used the summary produced by the ESCA-BERT represented a significant advance. In the first block of criteria ranking, 5 systems were simultaneously ranked. The gold summary set the upper bound except for the relevance. Unsurprisingly, since the gold summaries of CNN/DailyMail are mostly top sentences in the articles, their relevance can not be guaranteed. We also found the ESCA-BERT produced the most popular summaries. In the second block, the ESCA with novelty and relevance control were also evaluated together with the bottom-up and the original ESCA. The ranking difference slightly varies, but it obviously proves that the
ESCA under novelty or relevance control gained the corresponding highest rank.

6 Conclusion
This paper presents a novel hybrid framework for long document summarization. The proposed ESCA model can explicitly explain the selection and control the abstract generation. It is equipped with the proposed pair-wise ranking extractor and seamlessly connect with the abstractor with the sentence deployed pointer. Therefore, the generation can be affected through the updated deploying attention. Both empirical and subjective experiments show that our model makes a statistically significant improvement over state-of-the-art baselines.

References
Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1662–1675. Association for Computational Linguistics.

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, Volume 1 (Long Papers), pages 1662–1675.

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

Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. 2019. Mixture content selection for diverse sequence generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3112–3122.

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. Computational Linguistics, 36(3):411–441.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints.

In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1998–2008.

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

Karl Moritz Hermann, Tomas Kocisky, 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, pages 1693–1701.

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.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving neural abstractive document summarization with explicit information selection modeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1787–1796.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.

William C Mann and Sandra A Thompson. 1988. Towards a functional theory of text organization.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 404–411.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Thirty-First AAAI Conference on Artificial Intelligence.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira 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, CoNLL, pages 280–290.
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, Volume 1 (Long Papers), pages 1747–1759.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Evan Sandhaus. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, 6(12):e26752.

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, Volume 1: Long Papers, pages 1073–1083.

Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. 2019. An entity-driven framework for abstractive summarization. arXiv preprint arXiv:1909.02059.

Xiaoyu Shen, Jun Suzuki, Kentaro Inui, Hui Su, Dietrich Klakow, and Satoshi Sekine. 2019. Select and attend: Towards controllable content selection in text generation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 579–590.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826.

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 (Volume 1: Long Papers), pages 1171–1181.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Advances in Neural Information Processing Systems, pages 2692–2700.

Wenbo Wang, Yang Gao, He-Yan Huang, and Yuxiang Zhou. 2019. Concept pointer network for abstractive summarization. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3067–3076.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2324–2335. Association for Computational Linguistics.

Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. 2019. Improving abstractive document summarization with salient information modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2132–2141.

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 Annual Meeting of the Association for Computational Linguistics, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. arXiv preprint arXiv:1906.03508.

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