Title-based Extractive Summarization via MRC Framework

Hongjin Kim[1], **Jai-Eun Kim**[2], **Harksoo Kim**[1,⋆]
[1]Konkuk University, Seoul, Republic of Korea
[2]Saltlux, Seoul, Republic of Korea
{jin3430, nlpdrkim}@konkuk.ac.kr, jaieun.kim@saltlux.com

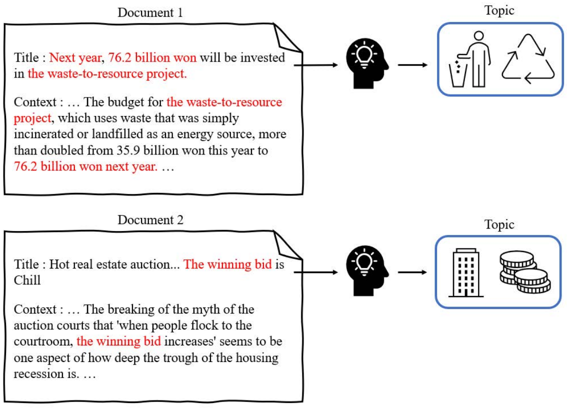
Abstract

Existing studies on extractive summarization have primarily focused on scoring and selecting summary sentences independently. However, these models are limited to sentence-level extraction and tend to select highly generalized sentences while overlooking the overall content of a document. To effectively consider the semantics of a document, in this study, we introduce a novel machine reading comprehension (MRC) framework for extractive summarization (MRCSUM) by setting a query as the title. Our framework enables MRCSUM to consider the semantic coherence and relevance of summary sentences in relation to the overall content. In particular, when a title is not available, we generate a TITLE-LIKE QUERY, which is expected to achieve the same effect as a title. Our TITLE-LIKE QUERY consists of the topic and keywords to serve as information on the main topic or theme of the document. We conduct experiments in both Korean and English languages, evaluating the performance of MRCSUM on datasets comprising both long and short summaries. Our results demonstrate the effectiveness of MRCSUM in extractive summarization, showcasing its ability to generate concise and informative summaries with or without explicit titles. Furthermore, our MRCSUM outperforms existing models by capturing the essence of the document content and producing more coherent summaries.

Keywords: Automatic Text Summarization, Extractive Summarization, Machine Reading Comprehension Framework, Title-based Summarization

1. Introduction

Extractive summarization is the task of generating a concise summary of a given document. The goal of extractive summarization is to extract and condense the core content from the document, while preserving the original meaning and context. Unlike abstractive summarization (Li et al., 2018), which generates a summary by understanding the content and rephrasing it in a new way, extractive summarization involves selecting core content directly from the given document to form the summary. It is widely used because extractive summarization is usually free from semantic, grammatical, and factual inconsistency problems (Zhong et al., 2020). Existing studies on extractive summarization have adopted two main approaches. The first method is that whereby extractive summarization is considered as a sequence labeling problem, and the model determines whether each sentence is selected (Cheng and Lapata, 2016; Nallapati et al., 2017). The second is the autoregressive method that was proposed to integrate sequence labeling into the autoregressive method (Zhou et al., 2018a). It selects summary sentences based on the relative importance between sentences of a document. Pretrained language models (PLM) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have exhibited surprisingly advanced performance in various NLP tasks. Liu and Lapata (2019) (BERTEXT) used BERT for extractive summarization. However, the above models are limited to sentence-level extraction, which leads to the selection of highly generalized sentences while the overall content of a document is overlooked (Zhong et al., 2020). Zhong et al. (2020) (MATCHSUM) proposed a summary-level framework for extractive summarization that implements a semantic text matching scheme for a candidate summary and a document, in which it is assumed that a good summary is semantically similar to the document. We adopted this assumption and revised it as follows: a good summary is semantically similar to the title. Unlike previous studies, we used the title as opposed to the entire document to match the candidate summary semantically.



Title : Next year, 76.2 billion won will be invested in the waste-to-resource project.

Context : … The budget for the waste-to-resource project, which uses waste that was simply incinerated or landfilled as an energy source, more than doubled from 35.9 billion won this year to 76.2 billion won next year. …

Title : Hot real estate auction... The winning bid is Chill

Context : … The breaking of the myth of the auction courts that 'when people flock to the courtroom, the winning bid increases' seems to be one aspect of how deep the trough of the housing recession is. …

Figure 1: Examples of title and context of the document. These examples are from the Modu-news dataset and have been translated from Korean into English. Further descriptions of the Modu-news dataset are provided in Section 4.1.

Moreover, we assume that extractive summariza-

[⋆] Corresponding author

LREC-COLING 2024, pages 16175–16186
20-25 May, 2024.

tion would benefit from the document's title since it extracts the core content of the document. The first role of the title of a document is to provide a compact summary to the reader, and the second is to attract the reader to read the document (Senda and Shinohara, 2002). In the case of the first, a title can be regarded as compressing a document into one sentence. Therefore, it can be helpful to consider the title when summarizing a document (Narayan et al., 2017). Unfortunately, titles are not always available, and existing work (Narayan et al., 2017) using the title did not take into account situations where the title is not available. To overcome this problem, we generate an alternative to the title using the topic and keywords of a document. We assume that the topic and keywords are inherent in the title. For example, in the title of the document in Figure 1, it can be inferred that the topics of the two documents are environment and economy. Moreover, in the context of the document in Figure 1, it can be observed that several words in the title overlap (red words). Based on these observations, it is expected that topics and keywords can be used simply but effectively, as opposed to the use of a title generator, to achieve the same effect as titles. Therefore, to alternate the title, we utilize the topic and keywords of a document. We use latent Dirichlet allocation (LDA) for the topic assignment (Blei et al., 2003). Furthermore, we use KeyBERT (Grootendorst, 2020) and TextRank (Mihalcea and Tarau, 2004) for keyword extraction.

Several approaches for formulating other NLP tasks (e.g., relation extraction and named entity recognition) as machine reading comprehension (MRC) have been studied in recent years (Li et al., 2019, 2020). These studies set a question to encode the primary information regarding their specific task to train models to consider a question. We formulate extractive summarization as the MRC framework, which extracts the answer in a document to queries (which we name MRCSUM). First, given the title, we set it as a query (which is referred to as the TITLE QUERY). Otherwise, we generate a TITLE-LIKE QUERY using the topic and keywords. Subsequently, MRCSUM extracts the candidate summary spans for the TITLE QUERY or TITLE-LIKE QUERY. Note that we acknowledge our work within the query-based summarization field, our approach uses a title-like 'query' but differs from traditional query-based summarization. The usual query-based method tailors summaries to specific external questions. Our method, instead, uses the document's title to guide the summary, ensuring it represents the document's overall theme and content. Our main contributions are as follows:

- We propose extractive summarization as the MRC framework that extracts candidate summary spans while considering the title. Unlike previous works in which sentences were scored and selected independently, MRCSUM considers the semantics of a compact summary because it trains the selection of summary sentences for the title.

- When the title is available, we utilize it as a query. In cases where the title is not available, we generate a TITLE-LIKE QUERY, aiming to achieve a similar effect as the actual title. Additionally, through our experiments, we provide evidence supporting our assumption that our MRC framework using the title and the TITLE-LIKE QUERY is effective for extractive summarization.

- We conducted experiments in two languages, English and Korean. Additionally, we evaluated our models on datasets containing both long and short summaries. Finally, we assessed the performance of our proposed MRCSUM system through automated evaluation using the ROUGE score and manual evaluation conducted by human annotators. Our MRCSUM has demonstrated superior performance in both automatic evaluation and human evaluation when compared to existing extractive summarization baselines.

2. Related Work

2.1. Extractive Summarization

In recent years, successful extractive summarization has been achieved using neural networks (Cheng and Lapata, 2016; Liu and Lapata, 2019; Zhong et al., 2020). The extractive summarization model mainly adopts an encoder-decoder structure that generates a vector representation of each sentence. The modeling of cross-sentence relations is one of the most effective methods for extracting appropriate sentences from a document, and it is generally achieved using recurrent neural networks (RNNs) (Cheng and Lapata, 2016; Nallapati et al., 2017). However, models based on RNNs are limited to sentence-level long dependency, whereby a long document or multiple documents result in significant performance degradation. Another method for extracting the cross-sentence relations is to design a graph structure. Early traditional methods, such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004), computed the cosine similarity. Liu and Lapata (2019); Wang et al. (2020) established the encoder based on the Transformer, which trains the interaction between sentence pairs.

2.2. Query-based Summarization

To generate a summary, query-based summarization (QBS) aims to structure sentences relating to the context of queries, and extractive techniques are common methods for conducting QBS. Otterbacher et al. (2009) suggested the form of question-answering extractive summarization based on Biased LexRank. Furthermore, Wang et al. (2013) presented the use of sparse trees and sentence compressions. Hermann et al. (2015) applied a neural network for QBS. However, extractive methods suffer from the problems of low coherence and less fluent summaries. To overcome this, in subsequent research on QBS, deep learning has been suggested as an approach to creating a query-based abstractive summary (Baumel et al., 2018; Hasselqvist et al., 2017). Nema et al. (2017) addressed the issue of repeated phrases by using encoder-decoder-based models while attempting to generate query-specific abstractive summaries. Xie et al. (2020) developed conditional self-attention to capture the conditional dependencies between given queries and input sequence pairs.

2.3. Title & Keywords based Summarization

Narayan et al. (2017, 2018a); Li et al. (2018) used the side information to summarize the text. Narayan et al. (2017) proposed a neural network for extractive summarization with side information such as title and image captions. However, this work did not consider the situation when the title is unavailable. Narayan et al. (2018a) used keywords from the input text to guide the process of summarization. These two works demonstrated that using the title and keywords of the document is helpful for summarization. Based on the findings from previous works (Narayan et al., 2017, 2018a), we incorporated the document's title, topic, and keywords to account for the semantic aspects of a concise summary. Moreover, unlike existing work (Nallapati et al., 2017), we consider situations where the title is unavailable and generate a TITLE-LIKE QUERY by utilizing the topic and keywords of a document to achieve a similar level of effectiveness as the title.

2.4. Machine Reading Comprehension

Machine reading comprehension (MRC) is a question answering task that extracts appropriate answers to users' queries in a given document. Recent studies of MRC are based on PLM, such as BERT, RoBERTa, and SpanBERT (Joshi et al., 2020). MRC systems frequently adopt span prediction that the learning of the start and end positions of the ground truth span within a given document. Then, the systems extract the answer span by summing the score of the start and end for all token positions. Even though this span prediction method is simple and effective, it struggles with extracting multiple answer spans. To extract multiple spans and a single span effectively, Jang et al. (2023) proposed the span matrix that extracts noncontinuous multiple spans and a training strategy that alleviates the performance drop in a single span. In the context of extractive summarization, in practice, the number of sentences in a summary can be one or more than two. In other words, we need to extract multiple spans and a single span effectively. Therefore, we adopt a span matrix inspired by (Jang et al., 2023) since it extracts both multiple spans and a single span better.

3. MRCSum

3.1. Task Definition

Given a document $D = \{s_1, s_2, ..., s_n\}$ consisting of n sentences, we must extract summary sentences in D by assigning a label $L \in [0, 1]$ to s_i. The label L indicates whether a sentence should be included in the summary.

3.2. Dataset Construction

First, we transform the tagging style of the summarization dataset into a set of (QUESTION, ANSWER, CONTEXT) triples. Let $s_i = \{x_1, x_2, ..., x_{n_i}\}$, where n_i denotes the number of tokens in the i-th sentence; then, we can redefine a document $D = \{x_1, x_2, ...x_N\}$, where N denotes the number of tokens in a document ($N = \sum_i n_i$). A summary sentence $x_{start,end} = \{x_{start}, ..., x_{end}\}$ is a substring of D satisfying start $\leq$ end and "$_{start,end}$" denotes the continuous tokens from the index 'start' to 'end'. Subsequently, we can obtain the triple $(q_y, x_{start,end}, D)$, where q_y is the TITLE QUERY or TITLE-LIKE QUERY.

3.3. Query Generation

The question generation method is vital because queries encode a compact summary of a document and significantly influence sentence selection.

TITLE QUERY Given the title, we set it as a query, as follows:

$$\begin{aligned} q_y &= Title(D) \\ Title(D) &= [w_1^t, w_2^t, ..., w_m^t], \end{aligned} \tag{1}$$

where m denotes the number of tokens in the title and w_i^t denotes the i-th token in the title.

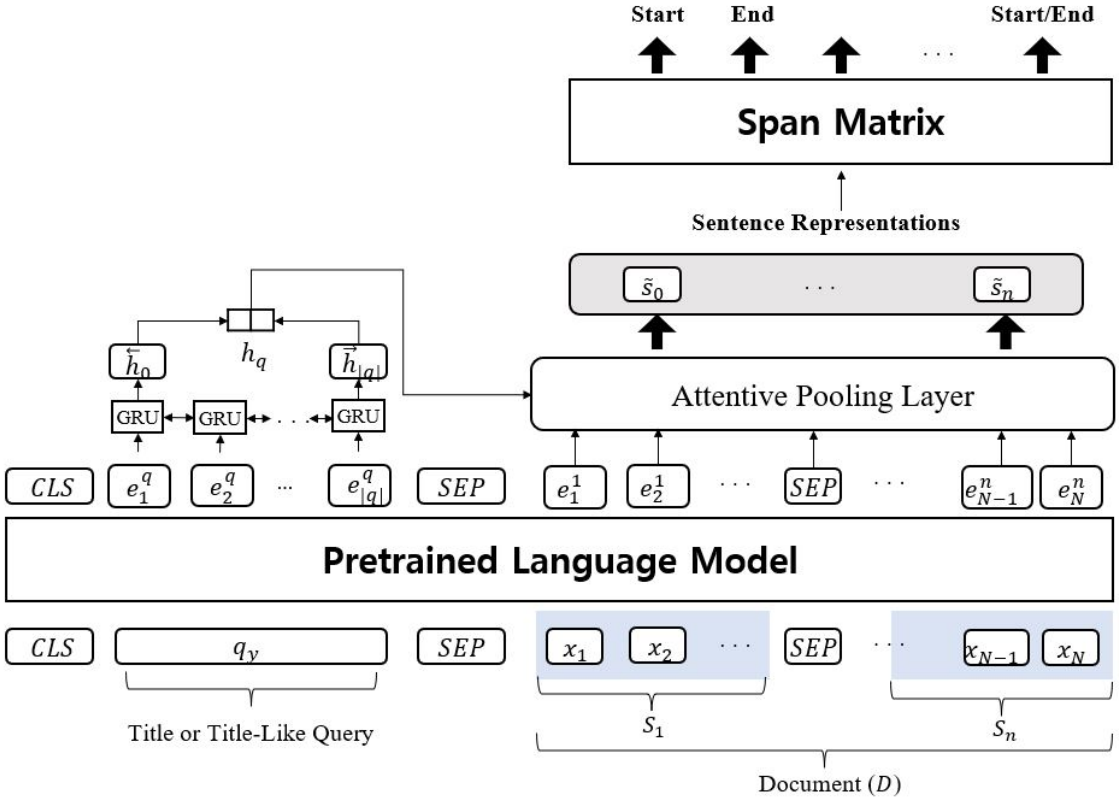


Figure 2: Overall architecture of MRCSUM.

TITLE-LIKE QUERY When the title is not available, we construct the TITLE-LIKE QUERY using the topic and keywords of the document. We use LDA to assign the topic to the document. In addition, we use KeyBERT and TextRank to extract the keywords from the document (Further details are in 4.3). First, we allocate topics to the entire dataset using LDA. Thereafter, a topic embedding table is initialized for all topics, and the topic embedding corresponding to each document is assigned as follows:

$$Topic(D) = W^{topic}[topic_D], \tag{2}$$

where $W^{topic} \in R^{T*H}$ is the embedding matrix, and $topic_D$ is the index of a topic. T is the number of the topics and H is the hidden size. Second, we extract the top five keywords from a single document D, which is used as a query along with the topic embedding. The TITLE-LIKE QUERY consisting of the topic and keywords is expressed as follows:

$$q_y = [topic_D, k_1^D, ..., k_5^D], \tag{3}$$

where k_i^D denotes the i-th keyword from a single document D. Finally, q_y can be redefined as follows:

$$q_y = \begin{cases} [W_1^t, ..., W_m^t], & \text{if given the title} \\ [topic_D, k_1^D, ..., k_5^D], & \text{Otherwise,} \end{cases} \tag{4}$$

3.4. Model Details

Figure 2 is the overall architecture of our proposed model. MRCSUM extracts the span $x_{start,end}$, which is a summary from D, given the query q_y. Here, we define span as a sentence unit. We use a PLM (Devlin et al., 2019; Liu et al., 2019) as the encoder to encode a query q_y and a document $D = \{x_1, x_2, ...x_N\}$. The input sequence is the concatenation of qeury q_y and document D, $\{[CLS], q_y, [SEP], x_1, ..., x_N\}$. The sentence separate token $[SEP]$ is inserted in between each sentence. The PLM outputs contextualized representations $E \in R^{\{C\} \times d}$, where C and d refer to the input sequence length and the hidden size, respectively.

Sentence Representations To guarantee that MRCSUM extracts the actual sentence rather than the sub-sentence, we define the span as a sentence unit. MRCSUM obtains sentence representations from the contextualized token representations E. To obtain the sentence representation considering the semantics of a query, we used attentive pooling (Yang et al., 2016) between tokens in a sentence S_i and a query vector h_q. To encode the query and get query vector h_q, we utilize the gated recurrent unit (GRU) (Chung et al., 2014) as follows:

$$\overrightarrow{h_j} = GRU(\overrightarrow{h}_{j-1}, e^q_j)$$
$$\overleftarrow{h_j} = GRU(\overleftarrow{h}_{j+1}, e^q_j) \quad (5)$$
$$h_q = [\overrightarrow{h}_{|q|}; \overleftarrow{h}_0],$$

Where e^q_j denotes the j-th token vector in a query q_y. The last forward hidden state $\overrightarrow{h}_{|q|}$ and the last backward hidden state $\overleftarrow{h}_0$ are concatenated to obtain query vector h_q. Then, the attentive pooling is calculated to apply the semantics of the query to the sentence representation as follows:

$$score_{u,v} = h_q e^u_v$$
$$a_{u,l} = exp(score_{u,l}) / \sum_{v=0}^{n_u} exp(score_{u,v}) \quad (6)$$
$$\tilde{s}_u = \sum_{v=0}^{n_u} a_{u,v} e^u_v,$$

Where e^u_v denotes the v-th token vector in the u-th sentence. In section 3.2, we define the number of tokens in the u-th sentence as n_u. The attention score, $score^u_v$, is the value reflecting the attention between the query vector h_q and the token vector e^u_v. The attention weight a^u_v means how much token vector e^u_v is associated with the query vector h_q. Finally, the sentence representation (i.e., sentence vector) is calculated by the weighted sum of an attention score, $score^u_v$, and token vector, e^u_v.

Summary Sentence Selection Given the sentence representations $S = \{\tilde{s}_1, \tilde{s}_2, ..., \tilde{s}_n\}$, MRCSum predicts the probability of each sentence is a start and an end of the summary span, as follows:

$$P_{start} = S \cdot T_{start}$$
$$P_{end} = S \cdot T_{end}, \quad (7)$$

where T_{start}, T_{end} are the weights to learn. Each row of P_{start}, P_{end} refers to the possibilities, which indicates whether each sentence is the start and end of a summary.

Span Matrix Figure 3 is the span matrix of our proposed model. Multiple summary sentences may exist within a single document. Therefore, MRCSum must predict multiple start and end indices. In addition, it must match a predicted start index with its corresponding end index. To obtain the start and end indices (i.e., $\hat{I}_{start}$ and $\hat{I}_{end}$), softmax and argmax are applied to each row of P_{start} and P_{end}:

$$\hat{I}_{start} = \{i | argmax(softmax(P^{(i)}_{start})) = 1\},$$
$$(i = 1, ..., n)$$
$$\hat{I}_{end} = \{i | argmax(softmax(P^{(i)}_{end})) = 1\}, \quad (8)$$
$$(i = 1, ..., n),$$

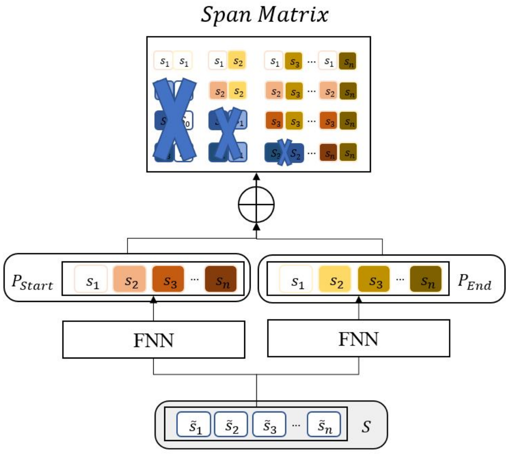


Figure 3: Span matrix of MRCSum

where $^{(i)}$ refers to the i-th row of a matrix. To pair the start index $i_{start} \in \hat{I}_{start}$ with its corresponding end index $i_{end} \in \hat{I}_{end}$, MRCSum predicts the pairing score using the span matrix in Figure 3, as follows:

$$P_{i_{start};j_{end}} = sigmoid(p \cdot concat(P_{i_{start}}; P_{j_{end}})), \quad (9)$$

where $p \in R^{1 \times \frac{d}{2}}$ denotes the weights to learn. As shown in Figure 3, we mask the span that the start index is higher than the end index (e.g., the start sentence of a summary is s_1, and the end sentence is s_0). When the sentences of the gold summary are s_1, s_2, and s_n, MRCSum trains to improve the pairing score of $[s_1; s_2]$ and $[s_n; s_n]$ in the span matrix.

3.5. Training Method

Two label sequences, $Y_{start} = \{y^s_1, y^s_2, \ldots, y^s_n\}$ and $Y_{end} = \{y^e_1, y^e_2, \ldots, y^e_n\}$, need to be predicted by MRCSum during training. Therefore, two losses are calculated for the start and end index predictions, as follows:

$$L_{start} = CE(P_{start}, Y_{start})$$
$$L_{end} = CE(P_{end}, Y_{end}), \quad (10)$$

where CE denotes the cross-entropy. Another label sequence, $Y_{start,end}$, indicates whether each start index should be paired with each end index. Therefore, we obtain the following start–end index pairing loss:

$$L_{span} = CE(P_{start,end}, Y_{start,end}), \quad (11)$$

Finally, these losses are minimized as follows:

$$L = L_{start} + L_{end} + L_{span}, \quad (12)$$

These losses are jointly trained in an end-to-end manner.

Datasets	# Pairs		# Tokens		# Ext
	Train	Test	Doc.	Sum.	
CNN/DM	287,084	11,489	766.1	58.2	3
Modu-news	3,950	439	642.9	121.8	3
Reddit	41,675	645	482.2	28.0	2
XSum	203,028	11,332	430.2	23.3	2

Table 1: Statistics of CNN/DM, Modu-news, Reddit, and XSum datasets. The value in Doc. and Sum. indicates the average length of the document and summary in the test set, respectively. # Ext is the number of sentences that should be extracted.

4. Experiments

4.1. Experimental Setup

We evaluated our model using both English and Korean datasets. For the English dataset, we used CNN/DailyMail (Hermann et al., 2015), which is a widely used news summarization dataset modified by Nallapati et al. (2017). We followed the same data-labeling method as that in Liu and Lapata (2019). We also used Reddit (Kim et al., 2019) and XSum (Narayan et al., 2018a) datasets and followed the same data labeling in MatchSum (Zhong et al., 2020). For the Korean dataset, we used Modu-corpus[1], which is a collection of various task datasets collected by the National Institute of Korean Language (NIKL) (Kim et al., 2021). We used a news summarization dataset from Modu-corpus that we named Modu-news. This dataset contains the title of a document. Statistics of all datasets are provided in Table 1. We evaluated our MRCSum automatically using the ROUGE score and manually through human evaluation.

4.2. Implementation Details

We implemented our model using the open-source PyTorch (Paszke et al., 2019) deep learning library. We adopted three types of PLMs for the shared encoder: BERT-base, RoBERTa-base, and RoBERTa-large. We set the batch sizes to 32 and 12 for the base and large models, respectively. We set the initial learning rate to 5e-5 for BERT-base and RoBERTa-base, and 5e-6 for RoBERTa-large. We conducted all experiments with an RTX 8000 GPU[2].

4.3. Details for Topic and Keywords

Topic Assign using LDA In order to utilize LDA, it is necessary to determine the number of topics in advance. In this study, we established 10, 65, 20, and 55 topics for the Modu-news, CNN/DM, Reddit, and Xsum datasets, respectively. Subsequently, LDA generated a topic distribution for each document, and we selected the topic with the highest probability score and assigned it to the respective document. It is important to note that we conducted topic modeling using only the training documents. Specifically, for the CNN/DM dataset, we utilized the training documents exclusively from that dataset. The same approach was applied to other datasets, and no additional dataset was utilized for topic modeling.

# of Topic	# of Keywords	R-1	R-2	R-L
1	3	56.82	40.51	43.00
	5	**56.91**	**40.58**	**43.07**
	7	56.63	40.39	42.88
3	3	56.84	40.52	43.00
	5	56.85	40.52	43.01
	7	56.53	40.31	42.86
5	3	56.60	40.37	42.87
	5	56.55	40.29	42.84
	7	56.48	40.14	42.65

Table 2: ROUGE F1 results with the various topics and keywords on the Modu-news dataset.

Keywords Extracting In this study, we utilize TextRank to extract keywords from each document using the TF-IDF method. This process generates a keyword distribution, from which we select the top 5 keywords. Additionally, when applying KeyBERT, we employ two BERT models to encode the document and different word combinations. These models produce representations for the document and combinations of words, enabling us to calculate the cosine similarity between them. We specify the number of word combinations as 5, and based on the KeyBERT model's score, we choose the best combination as the keywords for Title-like Query.

Lastly, we performed an empirical investigation to determine the optimal number of topics (ranging from 1 to 5) and keywords (ranging from 1 to 10). Table 2 shows the results obtained from the Modu-news dataset, while the findings from other datasets exhibit a similar trend. Through our analysis, we identified one topic and five keywords as the optimal choices.

4.4. Human Evaluation Details

For human evaluation, we sampled 50 articles from the Modu-news and CNN/DM datasets. Three people evaluated summaries. We ask evaluators to score between 0 and 2 on how the summary is informative. A score of 0 indicated that the sum-

[1] https://corpus.korean.go.kr/

[2] Our code will be available in the camera-ready if the paper is accepted

Model	R-1	R-2	R-L
LEAD-3	55.84	38.83	40.59
ORACLE	75.57	63.50	66.76
BertExt†	56.18	37.91	41.24
MatchSum†	56.65	38.47	41.67
SideNet† Title	56.52	38.04	41.50
MRCSum TextRank	56.91	40.58	43.07
MRCSum KeyBert	56.78	40.44	42.92
MRCSum Title	**57.82**	**41.83**	**44.03**

Table 3: ROUGE F1 results on Modu-news test set. The average results of five runs with random initialization are displayed. p-value < 0.05. The models with † were re-implemented in Korean.

Model	R-1	R-2	R-L
LEAD-3	40.43	17.62	36.67
ORACLE	52.59	31.23	48.87
SuumaRunner	39.60	16.20	35.30
Refresh	40.00	18.20	36.60
Latent	41.05	18.77	37.54
BanditSum	41.50	18.70	37.60
NeuSum	41.59	19.01	37.98
HiBert	42.37	19.95	38.83
PnBert	42.39	19.51	38.69
BertExt	42.57	19.96	39.04
BertExt + Tri-Blocking	43.23	20.22	39.60
MatchSum (BERT-base)	44.22	20.62	40.38
MatchSum (RoBERTa-base)	44.41	20.86	40.55
SideNet‡ (BERT-base)	43.99	20.42	39.97
SideNet‡ (RoBERTa-base)	44.01	20.47	40.04
MRCSum (BERT-base)	44.77	21.01	40.63
MRCSum (RoBERTa-base)	**44.81**	**21.07**	**40.66**

Table 4: ROUGE F1 results on CNN/DM test set. All results except for those of MRCSum and SideNet are cited from MatchSum. The average results of five runs with random initialization are reported. p-value < 0.05. The models with ‡ were re-implemented.

mary did not support important information from the original article. A score of 1 indicated that the summary partially supported important information from the original article. A score of 2 indicated that the summary fully supported important information from the original article. Both Korean and English evaluators are native speakers of their respective languages.

4.5. Experimental Results on Datasets with Long Summaries

Our MRCSum model utilizes a span matrix to effectively extract multiple sentences for summarizing documents. In other words, the span matrix enhances the summarization process when dealing with relatively long summaries. We conducted experiments on the Modu-news and CNN/DM datasets to examine the effectiveness of this approach. Table 3 presents the results for the Modu-news dataset. LEAD-3 refers to the extraction of the first three sentences of a document. ORACLE is the ground truth used in model training. In Table 3, MatchSum and MRCSum used RoBERTa-large, which was trained with a large-scale Korean corpus (Park et al., 2021). We have re-implemented all the baselines mentioned in Table 3, including BertExt, MatchSum, and SideNet, in Korean. MRCSum TextRank and MRCSum KeyBert used the title-like query, and extracted the keywords using TextRank and KeyBERT, respectively. MRCSum Title used the title query. Furthermore, MRCSum TextRank, MRCSum KeyBert, and MRCSum Title exhibited higher ROUGE scores compared to MatchSum. This suggests that MRCSum is highly effective for extractive summarization with and without a title being available. Next, SideNet(Narayan et al., 2017) is considered as the representative baseline model since it utilizes side information such as the title and image captions. SideNet employs a sentence extractor based on an attention mechanism to independently attend to side information while labeling sentences. To ensure a fair comparison, we implemented SideNet using the same PLM as the other models listed in Table 3. For experiments on Modu-news, we used only the title as side information of SideNet. Our MRCSum outperformed SideNet significantly, indicating the highly effective nature of formulating extractive summarization within the MRC framework.

We found that the superior performance of TextRank for keyword extraction can be attributed to its ability to better identify important keywords compared to KeyBERT. The token length limitation of 512 in KeyBERT prevents it from handling the entire token set of a document and extracting keywords from the truncated portions. This demonstrates that TextRank holds an advantage over KeyBERT in keyword extraction when considering the overall meaning of a document.

Table 4 displays the results on the CNN/DM dataset. We used BERT and RoBERTa, to compare the performance across different PLMs. Tri-Blocking is a simple but effective heuristic method for removing redundancies. MRCSum in Table 4 used the title-like query, which extracted keywords with TextRank. These keywords are also utilized by SideNet as side information. While SideNet did not consider situations where the title is not available, we implemented SideNet with title-like query to compare the effectiveness of using keywords with our MRCSum. We can see that our models outperformed all baseline models, with our MRCSum achieving higher ROUGE scores compared to SideNet. Notably, MRCSum shows higher ROUGE scores compared to MatchSum, which is the previous state-of-the-art Model. In contrast to our simple approach that focuses on the concise meaning of the entire document, MatchSum leverages semantic matching between the entire document and all candidate summaries. This entails calculating the representation of the entire document and all potential sentence combinations, resulting in high time complexity. The superior performance of our MRCSum compared to MatchSum highlights the efficiency of our proposed model. By utilizing the title and title-like query, we effectively consider the semantics of a document within an MRC framework for extractive summarization. This finding suggests our MRCSum effectively leverages the title and keywords. Finally, these superior ROUGE scores demonstrate the effectiveness of our method, which applies the MRC framework to extractive summarization tasks in both Korean and English datasets.

4.6. Experimental Results on Datasets with Short Summaries

We utilize a span matrix to extract multiple sentences effectively, but our MRCSum still can extract a single sentence well. In other words, our MRCSum are effective both in summarizing a document into single and multiple sentences. Many studies of abstractive summarization have evaluated the Reddit and XSum datasets since these datasets have short summaries (Zhong et al., 2020). Here, to investigate how our MRCSum is effective when dealing with short summaries, we conducted experiments on the Reddit and XSum datasets. Note that MRCSum in Table 5 used the title-like query, which extracted keywords with TextRank. SideNet in Table 5 used same keywords as side information. Although SideNet did not consider situations where the title is not available, we implemented SideNet with title-like query to compare the effectiveness of using keywords with our MRCSum. In Table 5, *Num* indicates the number of sentences models (i.e., BertExt, SideNet) extract; *Sel* indicates the number of sentences models (i.e., MatchSum, MRCSum) choose. First, our MRCSum offers the flexibility to choose the number of summary sentences, whereas most other methods, such as BertExt and SideNet, are limited to extracting a fixed number of summary sentences. Second, our MRCSum significantly outperformed SideNet, suggesting our MRC framework utilizes keywords more effectively for extractive summarization. Finally, MRCSum achieved superior performance compared to all baseline models, indicating our model extracts the summary effectively both on long and relatively short documents.

Model	R-1	R-2	R-L
Reddit			
BertExt (Num=1)	21.99	4.21	16.99
BertExt (Num=2)	23.86	5.85	19.11
SideNet* (Num=1)	22.28	4.81	17.18
SideNet* (Num=2)	24.05	5.88	19.76
MatchSum (Sel=1)	22.87	5.15	17.40
MatchSum (Sel=2)	24.90	5.91	20.03
MatchSum (Sel=1,2)	25.09	6.17	20.13
MRCSum (Sel=1)	23.66	5.72	19.02
MRCSum (Sel=2)	24.84	5.85	19.96
MRCSum (Sel=1,2)	**25.25**	**6.29**	**20.30**
XSum			
BertExt (Num=1)	22.53	4.36	16.23
BertExt (Num=2)	22.86	4.48	17.16
SideNet* (Num=1)	23.08	4.40	16.33
SideNet* (Num=2)	24.11	4.52	18.12
MatchSum (Sel=1)	23.35	4.46	16.71
MatchSum (Sel=2)	24.48	4.58	18.31
MatchSum (Sel=1,2)	24.86	4.66	18.41
MRCSum (Sel=1)	23.98	4.89	17.44
MRCSum (Sel=2)	24.69	4.97	18.47
MRCSum (Sel=1,2)	**25.08**	**5.16**	**18.66**

Table 5: ROUGE F1 results on the Reddit and XSum test set. The average results of five runs with random initialization are displayed. p-value < 0.05. The models with * were re-implemented.

Model	Informativeness	
	Modu-news	CNN/DM
LEAD-3	0.81	0.76
ORACLE	1.43	1.59
BertExt	1.01	0.97
MatchSum	1.15	1.12
SideNet	1.09	1.03
MRCSum Title-Like Query	1.20	1.16
MRCSum Title Query	**1.33**	**1.22**

Table 6: Human evaluation results of summaries for 50 randomly sampled articles in Modu-news and CNN/DM test set. The kappa ratio between evaluator scores was 0.42 for Modu-news and 0.46 for CNN/DM.

4.7. Human Evaluation

We conducted a human evaluation regarding the informativeness of extractive summaries. We randomly sampled 50 articles from the Modu-news and CNN/DM. We note that all titles in sampled articles functioned not merely as simple teasers but as effective summaries. According to Shapira et al. (2018), even humans have difficulty comparing two summaries that are not of similar length. Therefore, to alleviate this difficulty, all models extracted the number of sentences that are the same as the reference summary. Each article-summary pair was subsequently evaluated by three people. Each evaluator assigned a score in each category between 0 and 2 (Details are in 4.4). In Table 6, the informativeness score represents how effectively the summary covers the essential information from the article. As shown in Table 6, for the Modu-news dataset, our MRCSum significantly outperformed the baseline models and achieved an informativeness score comparable to that of the ORACLE. This suggests that MRCSum is more useful for humans than the previous model. For the CNN/DM dataset, our MRCSum outperformed all baseline models. It shows that our MRCSum is a more useful summarization model than the previous model for Korean and English.

4.8. Ablation Test

Model	ROUGE-L	
	Modu-news	CNN/DM
MRCSum	44.03	40.66
w/o QG	41.55	39.70
w/o SM	40.96	39.21
w/o QG & SM	40.29	39.16

Table 7: Results of the ablation test in Modu-news and CNN/DM test set. QG and SM denote query generation and span-matrix, respectively.

To investigate the effectiveness of our key components, we conducted ablation experiments on the Modu-news and CNN/DM datasets. Initially, instead of using TITLE QUERY or TITLE-LIKE QUERY, we employed a simple query of "What is the summary?" to assess their effectiveness. As shown in Table 7, MRCSum without TITLE QUERY or TITLE-LIKE QUERY (i.e., w/o QG) exhibited a significant decrease in ROUGE-L scores. This indicates that using a simple query fails to capture the specific topic or theme of a document, resulting in the extraction of sentence-level summaries. Next, to study the effectiveness of a span matrix, we conducted experiments without utilizing a span matrix and span loss in Eq.12.MRCSUM without a span matrix experienced a significant decline in ROUGE-L scores. This suggests that the absence of a span matrix limits the model's ability to extract multiple sentences effectively, leading to the generation of less comprehensive and coherent summaries. Finally, we performed experiments without employing query generation and a span matrix, as indicated in Table 7, which resulted in the lowest ROUGE-L score. Based on these findings, we can conclude that our TITLE QUERY, TITLE-LIKE QUERY, and span matrix are crucial in extracting more comprehensive and coherent summaries.

4.9. Analysis of Title-Like Query

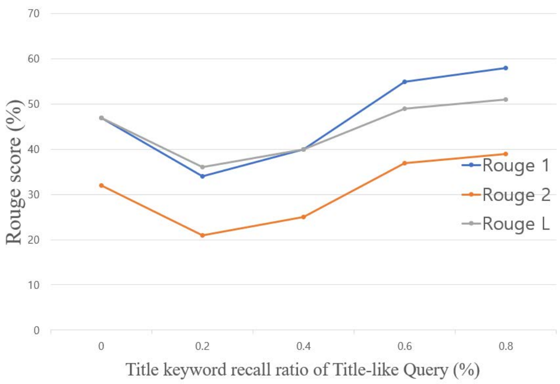


Figure 4: Effect of Title Keyword Recall Ratio on ROUGE Score in TITLE-LIKE QUERY. Results from the Modu-news Dataset.

Figure 4 illustrates the change in ROUGE scores based on the keyword recall ratio of the title in the TITLE-LIKE QUERY. It is evident that when the recall ratios were 0, the ROUGE scores were significantly higher than 0.2 (Note that cases with a low overlap, 0.2, are quite rare). This can be attributed to authors occasionally using abstract titles to captivate readers. Furthermore, a higher keyword recall ratio correlates with higher ROUGE scores, highlighting the importance of extracting words that convey the document's semantics, including keywords from the title.

5. Conclusions

We have proposed MRCSum, which considers the semantics of a compact summary because it trains the selection of summary sentences for the title. Moreover, when the title is not available, the TITLE-LIKE QUERY can be used. Our experimental results and human evaluations have demonstrated the effectiveness of our model for extractive summarization. In future work, we will explore distinguishing whether a news title is a summary or a teaser that attracts the reader. In addition, we will expand our work toward extracting effective summary sentences for both query-based and general purposes.

6. Limitations

Note that our MRCSum primarily works when the title roles a compact summary of documents since we use the title of documents as a query to match with the summary semantically. Although we use the title-like query when the title is unavailable, it is less likely to be effective when the title is used to entice a potential reader. However, from a different point of view, it may be practical to use the title-like query when the title of a document is used for enticement.

Although we did not compare MRCSum directly with the Large Language Model (LLM) which is the trend and recently popular works, it's important to point out that LLMs mostly use generation techniques, not extraction, and they often have a large model size. Additionally, LLMs typically utilize much larger training datasets than our foundation model (i.e., BERT and RoBERTa), which further increases the disparity. From a resource complexity perspective, we believe MRCSum has a clear advantage over LLMs. Furthermore, drawing direct comparisons between MRCSum and LLMs (e.g., through prompting or instructing for extractive summarization) may not present an equitable assessment due to the inherent differences in their operational paradigms and data scale advantages.

7. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) ("No. 2022-0-00369, (Part 4) Development of AI Technology to support Expert Decision-making that can Explain the Reasons/Grounds for Judgment Results based on Expert Knowledge" and "RS-2023-00216011, Development of artificial complex intelligence for conceptually understanding and inferring like human"). And this work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the metaverse support program to nurture the best talents ("IITP-2023-RS-2023-00256615") grant funded by the Korea government (MSIT).

8. Bibliographical References

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

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 (Volume 1: Long Papers)*, pages 484–494.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Johan Hasselqvist, Niklas Helmertz, and Mikael Kågebäck. 2017. Query-based abstractive summarization using neural networks. *arXiv preprint arXiv:1712.06100*.

KM Hermann, T Kočiskỳ, E Grefenstette, L Espeholt, W Kay, M Suleyman, and P Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 28.

Youngjin Jang, Hyeon-gu Lee, and Harksoo Kim. 2023. Long multispan prediction model for machine reading comprehension in healthcare domain. *Expert Systems with Applications*, 215:119300.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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)*, pages 55–60.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350.

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)*, pages 3730–3740.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018b. 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.

Shashi Narayan, Nikos Papasarantopoulos, Shay B Cohen, and Mirella Lapata. 2017. Neural extractive summarization with side information. *arXiv preprint arXiv:1704.04530*.

Preksha Nema, Mitesh M Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072.

Jahna Otterbacher, Gunes Erkan, and Dragomir R Radev. 2009. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing & Management*, 45(1):42–54.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, et al. 2021. Klue: Korean language understanding evaluation. *arXiv preprint arXiv:2105.09680*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080.

Yasuko Senda and Yaushi Shinohara. 2002. Analysis of titles and readers for title generation centered on the readers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Ori Shapira, David Gabay, Hadar Ronen, Judit Bar-Ilan, Yael Amsterdamer, Ani Nenkova, and Ido Dagan. 2018. Evaluating multiple system summary lengths: A case study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 774–778.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 6209–6219.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1384–1394.

Yujia Xie, Tianyi Zhou, Yi Mao, and Weizhu Chen. 2020. Conditional self-attention for query-based summarization. *arXiv preprint arXiv:2002.07338*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784.

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.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018a. 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)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018b. 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)*, pages 654–663.

9. Language Resource References

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, et al. 2021. What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of NAACL-HLT*, pages 2519–2531.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.