

Transforming Wikipedia Into Augmented Data for Query-Focused Summarization

Haichao Zhu , Li Dong, Furu Wei, Bing Qin , and Ting Liu

Abstract—The limited size of existing query-focused summarization datasets renders training data-driven summarization models challenging. Meanwhile, the manual construction of a query-focused summarization corpus is costly and time-consuming. In this paper, we use Wikipedia to automatically collect a large query-focused summarization dataset (named WIKIREF) of more than 280,000 examples, which can serve as a means of data augmentation. We also develop a BERT-based query-focused summarization model (Q-BERT) to extract sentences from the documents as summaries. To better adapt a huge model containing millions of parameters to tiny benchmarks, we identify and fine-tune only a sparse subnetwork, which corresponds to a small fraction of the whole model parameters. Experimental results on three DUC benchmarks show that the model pre-trained on WIKIREF has already achieved reasonable performance. After fine-tuning on the specific benchmark datasets, the model with data augmentation outperforms strong comparison systems. Moreover, both our proposed Q-BERT model and subnetwork fine-tuning further improve the model performance.

Index Terms—Query-focused summarization, natural language processing, data augmentation, neural networks.

I. INTRODUCTION

QUERY-FOCUSED summarization aims to create a brief, well-organized and informative summary for a document with specifications described in the query. Various unsupervised methods [1]–[7] and supervised methods [8]–[14] have been proposed for the purpose. DUC [15] 2005 - 2007 are high-quality query-focused summarization benchmarks constructed by humans. But the limited size renders training neural query-focused summarization models challenging, especially for the data-driven methods. Meanwhile, the manual construction of a large-scale query-focused summarization dataset is costly and time-consuming.

Manuscript received 19 July 2021; revised 4 March 2022; accepted 21 April 2022. Date of publication 3 May 2022; date of current version 28 July 2022. The work of Bing Qin and Ting Liu was supported in part by Science and Technology Innovation 2030 - New Generation Artificial Intelligence Major Project under Grant 2018AA0101901, in part by National Key Research and Development Project under Grant 2018YFB1005103, in part by the General Project of National Natural Science Foundation of China under Grant 61976073, and in part by Shenzhen Foundational Research Funding under Grant JCYJ20200109113441941. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Preslav Nakov. (Corresponding author: Bing Qin.)

Haichao Zhu, Bing Qin, and Ting Liu are with the Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China (e-mail: hczechu@ir.hit.edu.cn; qinb@ir.hit.edu.cn; tliu@ir.hit.edu.cn).

Li Dong and Furu Wei are with Microsoft Research Asia, Beijing 100080, China (e-mail: lidongl@microsoft.com; fuwei@microsoft.com).

The dataset is publicly available on-line at <https://aka.ms/wikiref>.

Digital Object Identifier 10.1109/TASLP.2022.3171963


Marina Beach

From Wikipedia, the free encyclopedia

Contents

- 1 History
- 2 Ecology
 - 2.1 Environment
 - 2.2 Flora and fauna
- 3 Dimensions and characteristics
- 4 Infrastructure and activities
- 5 Structures along the beach
- 6 Renovation
- 7 Safety measures and policing
- 8 Controversies
- 9 Incidents
- 10 Events
- 11 Transportation
- 12 Legacy



The Marina Beach after the tsunami 

With the assistance of the World Bank, the government built 2,000 temporary Marina beach shelters each measuring about 250 sq.ft. to house families affected by the tsunami at a cost of ₹ 172.3 million.^[83]

11,000 remain homeless even as shelters rot

Vivek Narayanan | TNN | Updated: Mar 6, 2011, 6:47 IST

But the shelters are not of much use for the fishermen either. The fisher folk sleep on the sand in the night. **They say that the 250-sq ft tsunami shelters built at a cost of Rs 17.23 crore are too small for families.**

The government built the Marina beach shelters with World Bank money to house families affected by the 2004 tsunami. More recently, it has earmarked these shelters for fisher folk who were forced to move out of the nearby Tamil Nadu Slum Clearance Board houses that are being pulled down. The fisher folk say the government wants to move their families to Kannagi Nagar.

Fig. 1. Example of automatic query-focused summarization dataset construction. Given a statement in Wikipedia article “Marina Beach,” we take the body text of citation as the document, use the article title along with section titles (i.e., “Marina Beach, Incidents”) to form a query, and the statement is the summary.

To advance neural query-focused summarization with limited data, we propose to transform Wikipedia into a large-scale query-focused summarization dataset (named WIKIREF) as a means of data augmentation. To automatically construct query-focused summarization examples using Wikipedia, we use the citations of the statements in Wikipedia articles as pivots to align the queries and documents. Fig. 1 shows an example that is constructed by the proposed method. We first take the highlighted statement as the summary. Its supporting citation is expected to provide an adequate context to derive the statement, thus can serve as the source document. On the other hand, the section titles give a hint about which aspect of the document is the summary’s focus. Therefore, we use the article title and the section titles

of the statement to form the query. Given that Wikipedia is the largest online encyclopedia, we can automatically construct massive query-focused summarization examples. At last, we have WIKIREF dataset of more than 280,000 examples.

Most extractive summarization models on the DUC benchmarks can be decomposed into two modules, i.e., sentence scoring and sentence selection. Sentence scoring aims to measure query relevance and sentence salience. It is well acknowledged that pre-trained language models [16]–[21], e.g., BERT [16], exhibit strong text understanding ability. In this paper, we develop a BERT-based model (Q-BERT) to score sentences. The model takes the concatenation of the query and the document as input. The token-level interactions between the query and the document and their internal interactions are all carried out through multi-head self-attention mechanism [22] in the BERT encoder. We then apply a query-focused pooling layer on top of the contextual token encodings to get the vector representations of the sentences. At last, we use a simple linear layer to get the score of each sentence extracted into the summary. Given sentence scores, we follow the common practice of previous works to select top-ranked sentences with minimal redundancy constraints as the final summary.

Given the proposed extractive model and the large-scale WIKIREF dataset as augmentation data, we first pre-train the model on the WIKIREF. Then we fine-tune the pre-trained model on the benchmark datasets. The tiny benchmarks only have no more than 100 examples that can be used to fine-tune the BERT-base model, which contains at least hundreds of millions parameters, e.g., size of BERT-base is 110 *M*. Even during fine-tuning, the mismatch between the tiny size of benchmarks and the massive number of parameters poses great challenges to the model optimization. Therefore, we only fine-tune a small fraction of the whole pre-trained model parameters, which correspond to a sparse subnetwork within the original model.

Experimental results on three DUC benchmarks show that the model achieves competitive performance by fine-tuning, and using WIKIREF as a means of data augmentation outperforms strong comparison extractive summarization systems. The proposed Q-BERT model with pooling layer and the sparse subnetwork fine-tuning (ST) strategy both further improve the model performance. More importantly, Q-BERT shows that the lacking of large-scale datasets hinders developing more effective data-driven models. The WIKIREF is shown to help reveal the effectiveness of these models and is also shown to be an eligible large-scale dataset to advance query-focused summarization research. Further analysis on augmentation data shows that the data quality is more important than the scale. Explorations on model structure find that the pooling methods are important to get effective sentence representations. And the choice of sparse subnetworks for fine-tuning is also critical to the performance besides using fewer parameters.

II. WIKIREF : TRANSFORMING WIKIPEDIA INTO QUERY-FOCUSED SUMMARIZATION DATASET

We automatically construct a query-focused summarization dataset (named as WIKIREF) using Wikipedia and corresponding

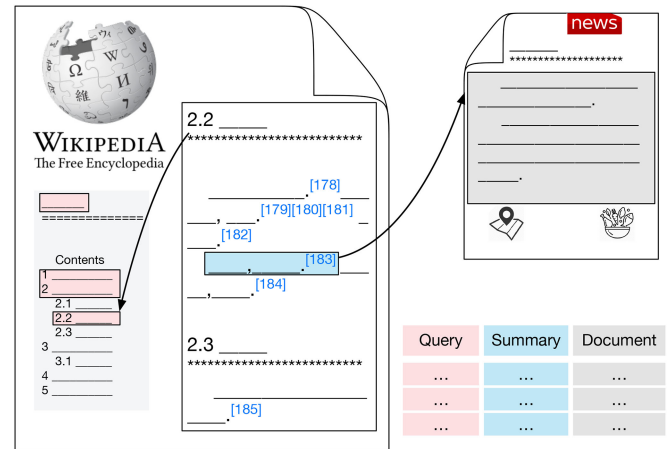


Fig. 2. Illustration of WIKIREF examples creation using Wikipedia and reference pages.

reference web pages. In the following sections, we will first elaborate on the creation process. Then we will analyze the queries, documents and summaries quantitatively and qualitatively.

A. Data Creation

We follow two steps to collect and process the data: (1) we crawl and parse English Wikipedia articles and their references to form raw examples; (2) we process and filter the raw examples through a set of fine-grained rules.

1) *Raw Data Collection*: In the first step, we parse the English Wikipedia database dump into plain text and save statements with citations. To maintain the highest standards possible, most statements in Wikipedia are attributed to reliable, published sources that can be accessed through hyperlinks. If a statement is attributed to multiple citations, only the first citation is used. We do not extend it to multi-document summarization. We limit the sources of the citations to four types, namely web pages, newspaper articles, press and press releases.¹

A query-focused summarization example consists of a summary, a document and a query. We find that the statement can be seen as a summary of the supporting citations from a certain aspect. Therefore, we can take the title and the body of the citation² as the document and treat the statement as the summary. We also find section titles are rough indicators of what aspects the statements focus on. So we stack the article title and the section titles to form the query. It is worth noticing that the queries are keywords, instead of natural language text as in other query-focused summarization datasets. Now we have all the constituents a query-focused summarization example needs.

We illustrate the raw data collection process in Fig. 2. The associated query, summary and the document are highlighted in colors in the diagram. Eventually, we have collected more than 2,000,000 examples through the raw data collection step.

¹ Citation types including book, journal, AV media, Wikidata, album notes, comic, conference, court, act, encyclopedia, episode, mailing list, map, news group, patent, thesis and video game are skipped.

² pyproject.org/project/newspaper is used for downloading and parsing.

TABLE I
STATISTICS OF TRAINING SET, DEVELOPMENT SET AND TEST SET OF THE WIKIREF DATASET

	Train	Dev	Test
Total Examples	256,724	12,000	12,000
Wiki Articles	160,223	11,457	11,476
Document Tokens	397.7	395.4	398.7
Document Sents	18.8	18.7	18.8
Summary Tokens	36.1	35.9	36.2
Summary Sents	1.4	1.4	1.4
Query Depth	2.5	2.5	2.5
Query Tokens	6.7	6.8	6.7

TABLE II
PERCENTILES FOR DIFFERENT ASPECTS OF THE WHOLE WIKIREF DATASET

	5	20	40	50	60	80	95
Document Tokens	208	267	346	387	431	530	618
Document Sents	9	12	16	18	20	25	33
Summary Tokens	14	20	27	31	36	50	75
Summary Sents	1	1	1	1	1	2	3
Query Depth	2	2	2	2	3	3	4
Query Tokens	3	4	5	6	7	9	13

2) *Data Curation*: To make sure the statement is a plausible summary of the cited document, we process and filter the examples through a set of fine-grained rules. The texts are tokenized and lemmatized using Spacy.³

First, we calculate the unigram recall of the summary with reference to the document, where only the non-stop words are considered. We throw out the example whose score is lower than the threshold. Here we set the threshold to 0.5 empirically, which means at least more than half of the summary tokens should be in the document. It controls the quality of the dataset rather than restricts the summaries to be strictly extractive.

Next, we filter the examples under multiple length and sentence number constraints. To set reasonable thresholds, we get the statistics of the examples survived in the previous step. The 5th and the 95th percentiles are used as low and high thresholds of each constraint.

Finally, to make sure generating the summary with the given document is feasible, we filter the examples by extractive oracle score. The extractive oracle is obtained through a greedy search over sentence combinations with no more than 5 sentences. ROUGE-2 recall is the scoring metric and only the examples with an oracle score higher than 0.2 are kept.

After running through the above curation steps, we have the WIKIREF dataset with 280,724 examples. We randomly split the data into training, development and test sets, and ensure no overlapping documents across splits.

B. Data Statistics

Tables I and II show statistics of the WIKIREF dataset. The development set and the test set contains 12,000 examples each. Statistics across splits are evenly distributed. The numerous Wikipedia articles cover a wide range of topics. The average

³spacy.io.

TABLE III
QUALITY RATING RESULTS OF HUMAN EVALUATION ON THE WIKIREF DATASET

ORACLE INTERVAL	RELATEDNESS	SALIENCE
20 ~ 30	2.87	2.33
30 ~ 50	2.80	2.40
50 ~ 70	2.87	2.53
70 ~ 100	2.93	2.60

“Relatedness” indicates the relatedness of the summary and the query. “Salience” indicates to what extent the summary conveys the salient document content. Two metrics are scored from 1 to 3, the higher the better.

depth of the query is 2.5 with article titles are considered. Since the queries are keywords in WIKIREF, it is relatively shorter than the natural language queries with an average length of 6.7 tokens. Most summaries are composed of one or two sentences. The document contains 18.8 sentences on average.

C. Human Evaluation

We also conduct a human evaluation on 60 WIKIREF samples to examine the quality of the automatically constructed data. We partition the examples into four bins according to the oracle scores and then sample 15 examples from each bin. Each example is scored by three volunteers in two criteria: (1) “Relatedness” examines to what extent the summary is a good response to the query and (2) “Salience” examines to what extent the summary conveys salient document content given the query. Three participants are asked to score each example from 1 to 3.

Table III shows the evaluation results. We can see that the summaries are good responses to the queries across bins. Since we take section titles as the query and the statement under the section as the summary, the high evaluation score can be attributed to high-quality Wikipedia pages. When the oracle scores are getting higher, the summaries continue to better convey the salient document content specified by the query. On the other hand, we notice that sometimes the summaries only contain a proportion of salient document content. But it is acceptable to use for data augmentation purposes.

III. METHODOLOGY

Fig. 3 gives an overview of our method based on data augmentation. We first pre-train the model on the automatically constructed augmentation data, and then fine-tune model parameters on the human annotated benchmarks. To train large models more efficiently on small benchmarks, we apply a subnetwork fine-tuning strategy (ST), which identifies sparse subnetworks in the model and only update the corresponding fraction of the parameters. In the following, we will first describe our BERT-based query-focused summarization model (Q-BERT). Then we introduce the sparse subnetwork fine-tuning strategy.

A. Input Representation

The query $\mathcal{Q} = (q_1, q_2, \dots, q_m)$ of m tokens sequence and the document $\mathcal{D} = (s_1, s_2, \dots, s_n)$ containing n sentences are flattened and packed as a token sequence as input. Following

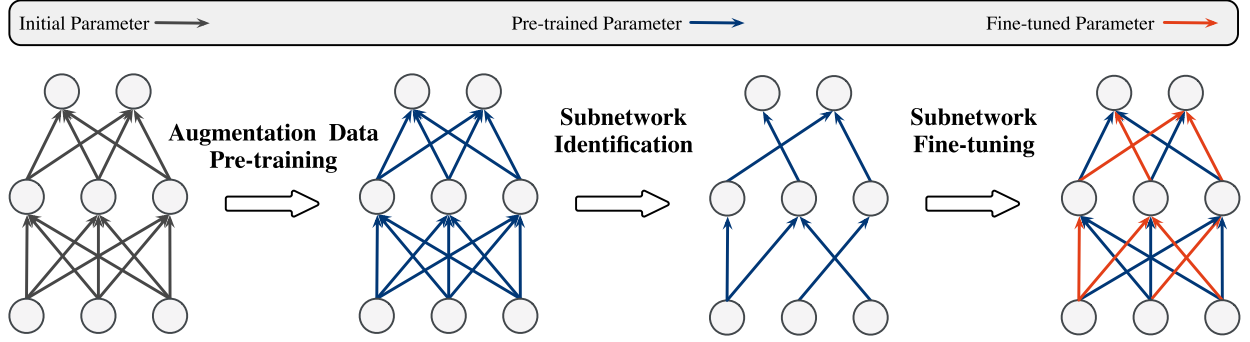


Fig. 3. The overview of our method based on data augmentation. We first pre-train on the augmentation data, i.e., WIKIREF. Then we identify a sparse subnetwork whose parameters are fine-tuned on the small benchmarks.

the standard practice of BERT, the input representation of each token is constructed by summing the corresponding token, segmentation and position embeddings. Token embeddings projects the one-hot input tokens into dense vector representations. Two segment embeddings \mathbf{E}_Q and \mathbf{E}_D are used to indicate query and document tokens respectively. Position embeddings indicate the absolute position of each token in the input sequence. To embody the hierarchical structure of the query in the sequential input, we insert a [L#] token before the #-th query token sequence. We also insert a [CLS] token at the beginning and a [SEP] token at the end.

B. BERT Encoding Layer

In the encoding layer, we use BERT [16], a deep Transformer [22] consisting of stacked self-attention layers, as the encoder to aggregate query, intra-sentence and inter-sentence information into token-level encodings. Given the packed input embeddings $\mathbf{H}^0 = [\mathbf{x}_1, \dots, \mathbf{x}_{|x|}]$, we apply an L -layer Transformer to encode the input:

$$\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1}) \quad (1)$$

where $l \in [1, L]$. The embedding of the i -th input token is \mathbf{h}_i^L .

C. Query-Focused Pooling Layer

To get the query-focused sentence-level representation of each document sentence, we pool the token-level BERT encodings. We first have the vector representation of the query via mean pooling:

$$\mathbf{h}_Q = \text{MeanPool}([\mathbf{h}_1^L, \dots, \mathbf{h}_m^L]) \quad (2)$$

Then for the token encodings $\mathbf{H}_j^L \in \mathbb{R}^{d \times |s_j|}$ of the j -th sentence, we apply the weighted mean pooling to get its vector representation:

$$\mathbf{v}_j = \text{softmax}(\mathbf{h}_Q \mathbf{W}_q \mathbf{H}_j^L) \mathbf{H}_j^L \quad (3)$$

where \mathbf{W}_q is a trainable parameter matrix. The weight of each token in the sentence is determined according to the query.

D. Output Layer

The output layer is used to score sentences for extractive query-focused summarization. Given $\mathbf{v}_j \in \mathbb{R}^d$ is the vector representation for the j -th sentence. When the model is supervised by sentence classification, the output layer is a linear layer followed by a sigmoid function σ :

$$P(s_i|Q, D) = \sigma(\mathbf{W}_c \mathbf{v}_j + \mathbf{b}_c) \quad (4)$$

where \mathbf{W}_c and \mathbf{b}_c are trainable parameters. The output is the probability of including the i -th sentence in the summary.

When it comes to sentence regression, a linear layer without activation function is used to estimate the score of a sentence:

$$\mathbf{r}(s_i|Q, D) = \mathbf{W}_r \mathbf{v}_j + \mathbf{b}_r \quad (5)$$

where \mathbf{W}_r and \mathbf{b}_r are trainable parameters.

E. Model Training and Inference

The training objective of sentence classification is to minimize the binary cross-entropy loss:

$$\mathcal{L} = - \sum_i^n y_i \log P(s_i|Q, D) + (1 - y_i) \log(1 - P(s_i|Q, D)) \quad (6)$$

where $y_i \in \{0, 1\}$ is the oracle label of the i -th sentence.

Training sentence regression model is to minimize the mean square error between the estimated score and the oracle score:

$$\mathcal{L} = \frac{1}{n} \sum_i^n (\mathbf{r}(s_i|Q, D) - g(s_i|\mathcal{S}_{ref}))^2 \quad (7)$$

where \mathcal{S}_{ref} is the reference summary and $g(s_i|\mathcal{S}_{ref})$ is the oracle score of the i -th sentence.

During inference, a query-specific subset of \mathcal{D} is extracted as the output summary $\hat{\mathcal{S}}$, subject to a length constraint l_c :

$$\begin{aligned} \hat{\mathcal{S}} &= \underset{\mathcal{S} \subseteq \mathcal{D}}{\text{argmax}} \sum_{s_i \in \mathcal{S}} \mathcal{M}(s_i|Q, D; \theta) \\ \text{s.t.} \quad &\sum_{s_i \in \mathcal{S}} |s_i| \leq l_c \end{aligned} \quad (8)$$

where \mathcal{M} is a sentence scoring model and θ are the parameters.

F. Sparse Subnetwork Fine-Tuning (ST)

Our Q-BERT model contains tremendous number of parameters, first pre-trained on the augmentation data, and then fine-tuned on the small benchmarks. The mismatch between the small data size and the massive number of parameters renders updating all parameters challenging. Thus, we exploit a small fraction of parameters that deliberately selected for query-focused summarization by identifying sparse subnetworks in the pre-trained model.

We adopt a simple method to identify sparse subnetworks by including the largest magnitude parameters of the BERT encoding layer. It is done in one step and does not require any annotated samples. We compare the magnitude of parameters within each parameter matrix independently. That is, the percentage of the remaining parameters in each parameter matrix is the same. It avoids pruning some parameter matrices completely, especially with high sparsity. The input embedding layer and output layer are always kept intact in subnetworks. Note that all parameters participate in the forward computation, but only the parameters of the subnetworks are updated.

IV. EXPERIMENTS ON WIKIREF

In this section, we elaborate on experimental environment, model training and evaluation metrics. We then present the results of benchmarking WIKIREF as a standard query-focused summarization dataset.

A. Implementation Details

We use the uncased version of BERT-base for experiments. The max sequence length is set to 512. We use Adam optimizer [23] with learning rate of $3e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay of 0.01, and linear decay of the learning rate. We split long documents into multiple windows with a stride of 100. Therefore, a sentence can appear in more than one window. To avoid making predictions on an incomplete sentence or with a suboptimal context, we score a sentence only when it is completely included and its context is maximally covered. We search for the best training epoch out of $\{3, 4\}$ and select batch size out of $\{24, 32\}$. We run all experiments with NVIDIA V100-32 GB GPU cards using the PyTorch framework [24] and the Hugging Face Transformer library [25].

B. Evaluation Metrics

We use ROUGE [26] as our automatic evaluation metric. ROUGE⁴ is the official metric of the DUC benchmarks and widely used for summarization evaluation. ROUGE-N measures the summary quality by counting overlapping N-grams with respect to the reference summary. ROUGE-L measures the longest common subsequence. ROUGE-SU considers skip-gram with unigrams concurrence.

C. Settings

Training extractive summarization models requires ground-truth labels for document sentences. However, we can not find the sentences that exactly match the reference summary for most examples. In order to solve the problem, we use a greedy algorithm similar to [27] to find an oracle summary with document sentences that maximizes the ROUGE-2 F1 score with respect to the reference summary. Given a document of n sentences, we greedily enumerate the combination of sentences. For documents that contain numerous sentences, searching for a global optimal combination of sentences is computationally expensive. Meanwhile, it is unnecessary since the reference summaries contain no more than four sentences. So we stop searching when no combination with i sentences scores higher than the best combination with $i-1$ sentences. When training models with sentence regression supervision, the oracle score is its ROUGE-2 F1 score.

During inference, we rank sentences according to their predicted scores. Then we append the sentence one by one to form the summary if it scores higher than a threshold and is not redundant. We skip the redundant sentences that contain overlapping trigrams concerning the current output summary as in [28]. The threshold is searched on the development set to obtain the highest ROUGE-2 F1 score.

D. Baselines

We compare the proposed model with the following baselines.

- 1) *ALL*: outputs all sentences of the document as a summary.
- 2) *LEAD*: selects the leading sentences. We take the first two sentences for that the ground-truth summary contains 1.4 sentences on average.
- 3) *BERT*: is a pre-trained language model based on Transformer. We append a [CLS] token to each sentence and use its token-level encoding to fine-tune BERT.
- 4) *TRANSFORMER*: uses the same structure as the BERT with randomly initialized weights.

E. Results

We report ROUGE-1, ROUGE-2 and ROUGE-L scores.⁵ As shown in Table IV, Q-BERT with query-focused pooling layer outperforms all baselines and the strong BERT model. The improvements are more pronounced when training with sentence classification supervision. On average, the output summary consists of 1.8 sentences. LEAD is a strong unsupervised baseline that achieves comparable results with the supervised neural baseline Transformer. Even though WIKIREF is a large-scale dataset, training models with parameters initialized from BERT still significantly outperform Transformer. The model trained using sentence regression performs worse than the one supervised by sentence classification. It is in accordance with oracle labels and scores as expected. We observe a performance drop when generating summaries without queries (see “w/o Query”). It proves that the summaries in WIKIREF are indeed query-focused.

⁴ROUGE-1.5.5.

⁵-n 2 -m -c 95 -r 1000.

TABLE IV
ROUGE SCORES OF BASELINES AND THE PROPOSED MODEL ON WIKIREF DATASET

Systems	Dev			Test		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ALL	14.05	7.84	12.97	14.09	7.88	13.01
LEAD	26.55	10.66	21.99	26.32	10.48	21.81
<i>Regression Supervision</i>						
BERT	34.52	17.96	29.29	34.42	17.88	29.17
Q-BERT	34.83	18.18	29.51	34.58	17.92	29.27
ORACLE	51.34	35.80	45.62	51.41	35.89	45.68
<i>Classification Supervision</i>						
TRANSFORMER	28.18	12.92	23.92	28.07	12.80	23.79
BERT	35.40	18.49	30.30	35.08	18.15	29.99
BERT w/o Query	32.91	16.05	27.97	32.52	15.83	27.65
Q-BERT	35.85	18.91	30.59	35.50	18.45	30.37
ORACLE	54.34	37.39	48.34	54.46	37.52	48.51

TABLE V
STATISTICS OF DUC BENCHMARKS

	2005	2006	2007
Clusters	50	50	45
Documents	1,593	1,250	1,125
Sentences	46,033	34,585	24,176

V. EXPERIMENTS ON DUC BENCHMARKS

A. Dataset

The documents of DUC are from the news domain and grouped into clusters according to their topics. The summary is required to be no longer than 250 tokens. Table V shows statistics of the DUC datasets. Each document cluster has several reference summaries generated by humans and a query that specifies the focused aspects and desired information. We show an example query from the DUC 2006 dataset below:

EgyptAir Flight 990?
What caused the crash of EgyptAir Flight 990?
Include evidence, theories and speculation.

The first narrative is usually a title and followed by several natural language questions or narratives.

B. Settings

We follow the standard practice to alternately train our model on two years of data and test on the third. The oracle scores used in model training are ROUGE-2 recall of sentences. In this paper, we score a sentence by only considering the query and its document. Then we rank sentences according to the estimated scores across documents within a cluster. For each cluster, we fetch the top-ranked sentences iteratively into the output summary with redundancy constraints met. A sentence is redundant if more than half of its bigrams appear in the current output summary. To be comparable with previous work on DUC datasets, we report the ROUGE-1 and ROUGE-2 recall computed with official parameters⁶ that limits the length to 250 words.

⁶-n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -l 250.

The WIKIREF dataset is used as augmentation data for DUC datasets in two steps. We first pre-train a summarization model on the WIKIREF dataset. Subsequently, we use the DUC datasets to further fine-tune parameters of the best pre-trained model. We use 10% parameters for sparse subnetwork fine-tuning due to its optimal performance.

C. Baselines

We compare our method with several previous query-focused summarization models:

- 1) *LEAD*: is a simple baseline that selects the leading sentences to form a summary.
- 2) *QUERY-SIM*: is an unsupervised method that ranks sentences according to its TF-IDF cosine similarity to the query.
- 3) *SVR* [9]: is a supervised baseline that extracts both query-dependent and query-independent features and then using Support Vector Regression to learn the weights of features.
- 4) *ATTSUM* [11]: is a neural attention summarization system that tackles query relevance ranking and sentence salience ranking jointly.
- 5) *CRSUM* [12]: is the contextual relation-based neural summarization system that improves sentence scoring by utilizing contextual relations among sentences.
- 6) *PQSUM* [13]: uses the BERTSUM [28] model pre-trained on the CNN/DailyMail dataset [29].
- 7) *QUERYSUM* [14]: is a coarse-to-fine framework with its query relevance estimator trained with external question answering datasets.

D. Results

Table VI shows the ROUGE scores of previous models and our proposed method. Fine-tuning BERT on DUC datasets alone obtains comparable results. Our data augmentation method advances the model to a higher performance on all DUC benchmarks. And the Q-BERT and the sparse subnetwork fine-tuning both further improve the results. We also notice that models pre-trained on the augmentation data achieve reasonable performance without further fine-tuning model parameters. It implies

TABLE VI
ROUGE SCORES ON THE DUC 2005, 2006 AND 2007 BENCHMARKS

Systems	DUC 2005		DUC 2006		DUC 2007	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
<i>Without Data Augmentation</i>						
LEAD [12]	29.71	4.69	32.61	5.71	36.14	8.12
QUERY-SIM [12]	32.95	5.91	35.52	7.10	36.32	7.94
SVR [9]	36.91	7.04	39.24	8.87	43.42	11.10
CRSUM [12]	36.96	7.01	39.51	9.19	41.20	11.17
ATTSUM [11]	37.01	6.99	40.90	9.40	43.92	11.55
QUERYSUM [14]	-	-	41.60	9.50	43.30	11.60
BERT	37.82	7.88	41.35	9.60	43.55	11.39
Q-BERT	37.47	7.58	40.88	9.54	42.84	11.24
<i>Classification Pre-training</i>						
PQSUM [13]	37.55	7.84	40.41	9.22	42.41	11.08
DA PRE-TRAINED	36.19	7.00	38.67	7.88	40.08	9.19
BERT + DA	38.77	8.31	41.65	10.04	44.31	11.85
<i>Regression Pre-training</i>						
DA PRE-TRAINED	36.52	7.02	38.81	8.37	41.09	10.29
BERT + DA	38.44	8.33	41.64	9.98	44.14	12.12
Q-BERT + DA	38.68	8.51	41.81	10.17	44.72	12.43
Q-BERT + DA + ST	39.21	8.62	41.96	10.29	45.06	12.55
ORACLE	43.71	13.77	48.02	17.22	49.80	19.19

“DA” is short for data augmentation using the WIKIREF dataset. “DA Pre-Trained” denotes applying the model Pre-Trained on augmentation data to DUC without Fine-Tuning. “ST” is short for subnetwork Fine-Tuning.

TABLE VII
HUMAN EVALUATION RESULTS ON THE DUC 2007 BENCHMARK. TWO METRICS ARE SCORED FROM 1 TO 3, THE HIGHER THE BETTER

	RELATEDNESS	REDUNDANCY
BERT	2.20	2.75
BERT + DA	2.48	2.78

the WIKIREF dataset reveals useful knowledge shared by the DUC dataset. We pre-train models on augmentation data under both sentence classification and sentence regression supervision. Since the training objectives of pre-training and fine-tuning are the same, the performance of pre-training with regression supervision is slightly better.

Without data augmentation, fine-tuning the more complex Q-BERT performs worse than BERT on the tiny DUC benchmarks. However, by pre-training on the augmentation data, the advantages of Q-BERT are revealed and better results are achieved. It shows that the small size of DUC benchmarks renders training data-driven neural models difficult and hinders the development of more effective architectures for query-focused summarization.

E. Human Evaluation

We conduct a human evaluation of the output summaries before and after applying data augmentation. We sample 30 examples from the DUC 2007 dataset for analysis. Three volunteers are asked to score the outputs on a 1-3 scale. The results are shown in Table VII. We can see that the model augmented by the WIKIREF dataset produces more query-related summaries. We attribute this to the improved coverage and query-focused extraction brought by the large-scale augmentation data. As to

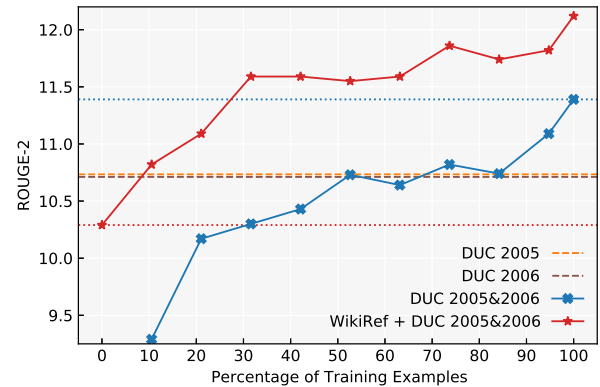


Fig. 4. ROUGE-2 score on the DUC 2007 with various number of training data. The x-axis shows the percentage of used training data. The horizontal lines indicate using DUC 2005 or DUC 2006 for training. The red line indicates using WIKIREF as augmentation data for pre-training and the blue line does not.

the redundancy, the WIKIREF dataset yields no significant effect to produce less redundant summaries.

VI. ANALYSES AND DISCUSSION

A. The Effectiveness of Data Augmentation

To further analysis the effectiveness of our data augmentation method, we vary the number of human annotated examples for fine-tuning BERT or model pre-trained on WIKIREF. Here we take DUC 2007 for evaluation, and the joint of DUC 2005 and 2006 for training. As shown in Fig. 4, we can see that our data augmentation method obtains consistent improvement over the BERT fine-tuning with various number of data. Using either DUC 2005 alone or DUC 2006 alone yields inferior performance than using both. In addition, the red horizontal dashed line shows that the pre-trained model performs as well as fine-tuning BERT

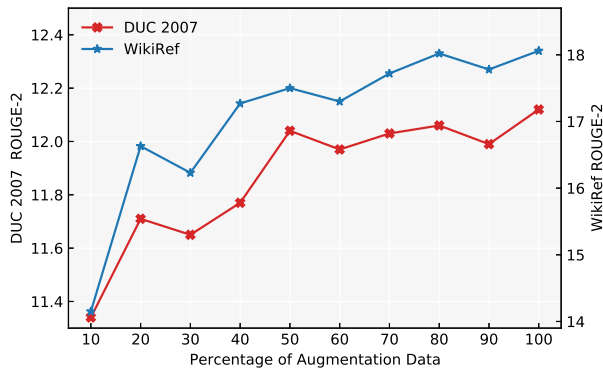


Fig. 5. Trends of ROUGE-2 scores with various number of WIKIREF training data, which used as augmentation data for DUC 2007.

using approximately 30% of the data. The blue horizontal dashed line shows that using the same amount of the data with our data augmentation outperforms full data BERT fine-tuning.

Three characteristics of the WIKIREF make it an effective augmentation data for the DUC benchmarks. At first, their documents share the same domain. The DUC documents are news articles. We also crawl newspaper webpages as one source of the WIKIREF documents. Secondly, queries in the WIKIREF dataset are in nature hierarchical that specify the focused aspects gradually. This intrinsic is in line with the DUC queries composed of several narratives to specify the desired information. It makes the model transfer to the DUC datasets more easily. At last, we construct the WIKIREF dataset to be a large-scale query-focused summarization dataset that contains more than 280,000 examples, which is in sharp contrast to the DUC datasets containing only 145 clusters with around 10,000 documents in total. The large size improves the coverage. Query relevance and sentence context can be better modeled using data-driven neural methods with WIKIREF. The above characteristics together contribute to the effective augmented data for query-focused summarization.

B. Impact of Augmentation Data Size and Quality

Firstly, we investigate the impact of data size on the effectiveness of data augmentation methods. Fig. 5 shows that increasing the data size improves the performance on WIKIREF. However, the improvements on DUC 2007 seem to be saturated when using more than half of the augmentation data. It could be partly due to that increasing the data size also introduces more noise.

Next, we analysis the impact of data quality. The oracle ROUGE-2 score is used as an proxy for sample quality. Specifically, we sort all the samples in ascending order and take half of the data with a step of 10%. In this way, we have five overlapped slices of data of increasing data quality, namely *Poor*, *Fair*, *Average*, *Good*, *Very Good*, *Excellent*. As shown in Fig. 6, higher data quality always yield better performance on both WIKIREF and DUC 2007. Furthermore, the highest quality slice with only half the data size outperforms the full augmentation data. It shows that data quality is more important than data size and reducing the noise in the augmentation data can lead to further improvement.

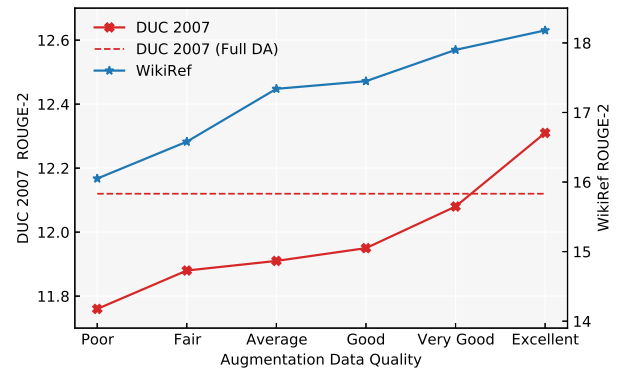


Fig. 6. Trends of ROUGE-2 scores with half of WIKIREF training data of different quality. The red horizontal dashed line indicates using the full WIKIREF training set as augmentation data for DUC 2007.

TABLE VIII
EFFECTIVENESS OF USING DIFFERENT POOLING METHODS TO GET SENTENCE-LEVEL REPRESENTATIONS

Model	WIKIREF		DUC 2007	
	R-2	R-L	R-2	R-SU4
BERT	18.15	29.99	12.12	17.58
+MAX POOLING	17.80	29.00	11.85	17.37
+MEAN POOLING	18.26	29.93	11.93	17.47
+ATTENTION POOLING	18.17	29.75	12.21	17.65
Q-BERT	18.45	30.37	12.43	17.75

C. Effect of Pooling Methods

In Section III-C, we apply query-focused pooling (2) to token-level encodings to get sentence vector. We have tried several other alternatives to the query-focused pooling. A special token is appended to each sentence and its token-level representation is instead used for prediction in BERT. Attention pooling [30] is similar to the proposed query-focused pooling, except it uses a trainable vector independent of the input query to assign the weight of sentence tokens in (3). As shown in Table VIII, max pooling encodes the least information and performs the worst. Mean pooling and attention pooling are comparable to BERT. At last, our model with query-focused pooling that weighs sentence tokens adaptively according to the input query works best.

D. Subnetwork Identification Methods

In Section III-F, we identify the subnetwork in one step by keeping the largest magnitude parameters of the model pre-trained on augmentation data. We have explored an iterative method [31] to identify subnetworks in models initialized by BERT or pre-trained on augmentation data. This iterative method gradually reduces the size of subnetwork during training with augmentation data. In Table IX, we show the results of fine-tuning the identified subnetworks on DUC 2007 benchmark. Fine-tuning subnetworks generated by the iterative method performs worse than that of the full model under the two initializations. The one-step method applied to the pre-trained model can yield subnetworks specific to the query-focused summarization task works best in our work. Note that we do

TABLE IX
PERFORMANCE OF DIFFERENT SUBNETWORK IDENTIFICATION METHODS

Method	Init	R-1	R-2	R-SU4
None	BERT	44.72	12.43	17.75
ITERATIVE	BERT	41.98	10.93	16.11
ITERATIVE	DA PRE-TRAINED	44.54	12.28	17.58
ONE-STEP	DA PRE-TRAINED	45.06	12.55	17.86

not apply one-step method to BERT because it generates the same general subnetwork for all tasks [32]–[34], which is not our focus.

VII. RELATED WORK

A wide range of unsupervised approaches has been proposed for extractive summarization. Surface features, such as n-gram overlapping, term frequency, document frequency, sentence positions [12], sentence length [11], and TF-IDF cosine similarity [4]. Maximum Marginal Relevance (MMR) [1] greedily selects sentences and considered the trade-off between saliency and redundancy. McDonald [3] treat sentence selection as an optimization problem and solve it using Integer Linear Programming (ILP). Lin and Bilmes [35] propose using submodular functions to maximize an objective function that considers the trade-off between coverage and redundancy terms.

Graph-based models make use of various inter-sentence and query-sentence relationships are also widely applied in the extractive summarization area. LexRank [2] scores sentences in a graph of sentence similarities. Wan and Xiao [4] apply manifold ranking to make use of the sentence-to-sentence and sentence-to-document relationships and the sentence-to-query relationships. We also model the above mentioned relationships, except for the cross-document relationships, like a graph at token level, which are aggregated into distributed representations of sentences.

Supervised methods with machine learning techniques [8]–[10] are also used to better estimate sentence importance. In recent years, few deep neural networks based approaches have been used for extractive document summarization. Cao *et al.* [11] propose an attention-base model that jointly handles sentence saliency ranking and query relevance ranking. It automatically generates distributed representations for sentences as well as the document. To leverage contextual relations for sentence modeling, Ren *et al.* [12] propose CRSum that learns sentence representations and context representations jointly with a two-level attention mechanism. Xu and Lapata [14] propose a coarse-to-fine framework which progressively estimates sentence relevance. Kulkarni *et al.* [36] propose SIBERT that extends HIBERT [37] to query-focused multi-document summarization by introducing a cross-document infusion layer and incorporating queries as additional contexts. The small data size is the main obstacle to develop neural models for query-focused summarization.

Wikipedia provides rich resources for exploring various natural language processing tasks. For text summarization, Liu *et al.* [38] build WikiSum, a multi-document

summarization dataset that uses both Wikipedia references and web search results to generate long Wikipedia article abstractively. Base on WikiSum, Hayashi *et al.* [39] propose WikiAsp for aspect-based summarization, a subset of section titles is selected as the set of aspects for each domain. Question answering datasets [40] are also explored to mine query-based multi-document summarization examples, the ComSum dataset [36], or train evidence estimator to improve the relatedness between the summary and the query [14]. We use Wikipedia to automatically construct a large dataset for single document query-focused summarization. It can be used as a standard dataset or as a means of data augmentation for small benchmarks.

VIII. CONCLUSION

In this paper, we propose to use Wikipedia articles and the corresponding references to automatically construct a large-scale query-focused summarization dataset named WIKIREF. The statements, supporting citations and article title along with section titles are used as summaries, documents and queries respectively. The WIKIREF dataset serves as a means of data augmentation for DUC benchmarks. It is also shown to be an eligible query-focused summarization benchmark. Moreover, we develop a BERT-based extractive query-focused summarization model and a sparse subnetwork fine-tuning method to improve the performance on tiny benchmarks. The results on DUC benchmarks show that our augmentation data facilitate query-focused summarization and helps to develop more efficient data-driven models. Quantitatively and qualitatively analysis shows that improving the augmentation data quality is more important than expanding the scale, which is promising in future work.

REFERENCES

- [1] J. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1998, pp. 335–336. [Online]. Available: <http://doi.acm.org/10.1145/290941.291025>
- [2] G. Erkan and D. R. Radev, “LexRank: Graph-based lexical centrality as salience in text summarization,” *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, 2004.
- [3] R. McDonald, “A study of global inference algorithms in multi-document summarization,” in *Advances in Information Retrieval*, G. Amati, C. Carpineto, and G. Romano, Eds., Berlin, Germany: Springer, 2007, pp. 557–564.
- [4] X. Wan and J. Xiao, “Graph-based multi-modality learning for topic-focused multi-document summarization,” in *Proc. 21st Int. Joint Conf. Artif. Intell.*, C. Boutilier, Ed., Pasadena, California, USA, 2009, pp. 1586–1591. [Online]. Available: <http://ijcai.org/Proceedings/09/Papers/265.pdf>
- [5] G. Feigenblat, H. Roitman, O. Boni, and D. Konopnicki, “Unsupervised query-focused multi-document summarization using the cross entropy method,” in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 961–964. [Online]. Available: <http://doi.acm.org/10.1145/3077136.3080690>
- [6] T. Baumel, M. Eyal, and M. Elhadad, “Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models,” *Comput. Res. Repository*, 2018, *arXiv:1801.07704*. [Online]. Available: <https://arxiv.org/abs/1801.07704>
- [7] H. Roitman, G. Feigenblat, D. Cohen, O. Boni, and D. Konopnicki, “Unsupervised dual-cascade learning with pseudo-feedback distillation for query-focused extractive summarization,” in *Proc. Int. World Wide Web Conf.*, 2020, pp. 2577–2584. [Online]. Available: <https://doi.org/10.1145/3366423.3380009>

- [8] M. Galley, "A skip-chain conditional random field for ranking meeting utterances by importance," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 364–372. [Online]. Available: <https://www.aclweb.org/anthology/W06-1643>
- [9] Y. Ouyang, W. Li, S. Li, and Q. Lu, "Applying regression models to query-focused multi-document summarization," *Inf. Process. Manage.*, vol. 47, no. 2, pp. 227–237, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306457310000257>
- [10] C. Li, X. Qian, and Y. Liu, "Using supervised bigram-based ILP for extractive summarization," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1004–1013.
- [11] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li, "AttSum: Joint learning of focusing and summarization with neural attention," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 547–556. [Online]. Available: <https://www.aclweb.org/anthology/C16-1053>
- [12] P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. de Rijke, "Leveraging contextual sentence relations for extractive summarization using a neural attention model," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 95–104. [Online]. Available: <http://doi.acm.org/10.1145/3077136.3080792>
- [13] M. T. R. Laskar, E. Hoque, and J. X. Huang, "WSL-DS: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 5647–5654. [Online]. Available: <https://aclanthology.org/2020.coling-main.495>
- [14] Y. Xu and M. Lapata, "Coarse-to-fine query focused multi-document summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 3632–3645. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.296>
- [15] H. T. Dang, "Overview of DUC 2005," in *Proc. Document Understanding Conf.*, 2005, pp. 1–12.
- [16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technologies*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, MN, USA, 2019, pp. 4171–4186. [Online]. Available: <https://doi.org/10.18653/v1/n19-1423>
- [17] Y. Liu *et al.*, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: <http://arxiv.org/abs/1907.11692>
- [18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2019, pp. 5753–5763. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9e67cc69-Abstract.html>
- [19] L. Dong *et al.*, "Unified language model pre-training for natural language understanding and generation," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2019, pp. 13 042–13 054. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html>
- [20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soiccut, "ALBERT: A lite bert for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=H1eA7AEtvS>
- [21] K. Clark, M.-T. Luong, Q. Le, and C. D. Manning, "Pre-training transformers as energy-based cloze models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 285–294. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.20>
- [22] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017, pp. 5998–6008. [Online]. Available: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [24] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [25] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45. [Online]. Available: <https://aclanthology.org/2020.emnlp-demos.6>
- [26] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, pp. 74–81. [Online]. Available: <https://www.aclweb.org/anthology/W04-1013>
- [27] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 654–663. [Online]. Available: <https://www.aclweb.org/anthology/P18-1061>
- [28] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3730–3740. [Online]. Available: <https://aclanthology.org/D19-1387>
- [29] K. M. Hermann *et al.*, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, 2015. [Online]. Available: <https://proceedings.neurips.cc/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf>
- [30] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 189–198. [Online]. Available: <https://aclanthology.org/P17-1018>
- [31] M. H. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," in *Proc. Int. Conf. Learn. Representations, Workshop Track*, 2018. [Online]. Available: <https://openreview.net/forum?id=S11N69AT>
- [32] S. Prasanna, A. Rogers, and A. Rumshisky, "When BERT plays the lottery, all tickets are winning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 3208–3229. [Online]. Available: <https://doi.org/10.18653/v1/2020.emnlp-main.259>
- [33] T. Chen *et al.*, "The lottery ticket hypothesis for pre-trained BERT networks," in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 15834–15846. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/b6af2c9703f203a2794be03d443af2e3-Paper.pdf>
- [34] C. Liang *et al.*, "Super tickets in pre-trained language models: From model compression to improving generalization," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 6524–6538. [Online]. Available: <https://aclanthology.org/2021.acl-long.510>
- [35] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 912–920.
- [36] S. Kulkarni, S. Chammas, W. Zhu, F. Sha, and E. Ie, "CoMSum and SIBERT: A dataset and neural model for query-based multi-document summarization," in *Proc. 16th Int. Conf. Document Anal. Recognit.*, J. Lladós, D. Lopresti, and S. Uchida, Eds., 2021, pp. 84–98.
- [37] X. Zhang, F. Wei, and M. Zhou, "HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5059–5069. [Online]. Available: <https://aclanthology.org/P19-1499>
- [38] P. J. Liu *et al.*, "Generating Wikipedia by summarizing long sequences," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=HygOvbWC>
- [39] H. Hayashi, P. Budania, P. Wang, C. Ackerson, R. Neervannan, and G. Neubig, "WikiAsp: A dataset for multi-domain aspect-based summarization," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 211–225, 2021. [Online]. Available: <https://transacl.org/ojs/index.php/tacl/article/view/2511>
- [40] T. Kwiatkowski *et al.*, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 452–466, Mar. 2019. [Online]. Available: <https://aclanthology.org/Q19-1026>



Haichao Zhu received the M.S. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in June 2017. He is currently working toward the Ph.D. degree with the Harbin Institute of Technology. His research interests include machine reading comprehension, question generation, text summarization, and knowledge distillation.



Li Dong received the Ph.D. degree from The University of Edinburgh, Scotland, U.K., in July 2019. He is currently a Senior Researcher with Natural Language Computing Group, Microsoft Research Asia, Beijing, China. His research interests include language model pre-training, question answering, and multilingual representation learning.



Bing Qin received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 2005. She is currently a Full Professor at the Department of Computer Science, and the Director of Research Center for Social Computing and Information Retrieval (HIT-SCIR), Harbin Institute of Technology. Her research interests include natural language processing, information extraction, document-level discourse analysis, and sentiment analysis.



Furu Wei received the Ph.D. degree from the Department of Computer Science, Wuhan University, Wuhan, China, in June 2009. He is currently a Senior Principal Research Manager with Microsoft Research Asia, Beijing, China, where he is leading the Natural Language Processing Group. He was a Staff Researcher with IBM Research, China (IBMCRL) from July 2009 to November 2010. He works on natural language processing (understanding and generation).



Ting Liu received the Ph.D. degree from the Department of Computer Science, Harbin Institute of Technology, Harbin, China, in 1998. He is currently a full Professor and the Director of Faculty of Computing, with the Harbin Institute of Technology. His research interests include information retrieval, natural language processing, and social media analysis.