

Importance Guided Query Focused Long-Input Summarization

Shuaiyao Ning¹, Caixia Yuan^{1*}, Xiaojie Wang¹, Ling Huang²

¹Beijing University of Posts and Telecommunications, Beijing, China

²AHI Fintech Limited, Beijing, China

nsy@bupt.edu.cn, yuancx@bupt.edu.cn, xjwang@bupt.edu.cn, linghuang@fintec.ai

Abstract: Query focused long-document summarization aims to generate a summary depending on both the given query and the given long text. Previous end-to-end method inputs the query and long text into a single model, and generates the summary. However, it's a difficult task for the end-to-end model to recognize which parts of the long text are more important for answering the query. Besides, the length of the query is too small, so the information of the query is diluted after concatenating with the long text. As a result, many summarized sentences are irrelevant to the query. In this paper, we propose IGQFS (Importance Guided Query Focused long-input Summarization), which predicts an importance score for each sentence as an auxiliary task, and simultaneously generates the summary. To this end, we design an improved cross-attention module for the decoder according to the predicted utterance importance scores. Experimental results on QMSum dataset show that the auxiliary task and improved cross attention module generate summaries more related to the queries, and our method performs better than the end-to-end baseline model.

Keywords: Query focused summarization; Utterance importance scores prediction; Utterance importance scores guided cross attention

1 Introduction

Text summarization [1,21,22,23,24,25] is a classic task of natural language processing, which requires a model to generate shorter texts that best summarize the important information of the input text. However, people usually don't just want to get a generic summary about the entire input text, but also want to get a summary about some details of some parts of the text, especially when faced with long texts containing thousands of words. Query focused summarization (QFS) can effectively meet this need by allowing people to propose a query about the content described in the text and help people obtain personalized summaries from long meeting records, multi-party dialogue data, and so on.

Recently, Zhong et al released QMSum dataset [5], which significantly promotes the research of QFS. QMSum is a query focused long meeting summarization dataset with the average 11,810 tokens of these meetings, which greatly exceeds the maximum input length of most existing state-of-the-art summarization pre-trained

models (512 or 1024 tokens). So it's not feasible to input query and meeting transcripts into the pre-trained models directly after concatenating them.

To solve this problem, two classes of methods were proposed: two-stage extractive-abstractive models and end-to-end models [6]. The two-stage models extract utterances related to the query and then input the concatenation of related utterances and query into the generator. Such two-stage extractive-abstractive models are widely used for long document summarization tasks [7,20], and reasonable for query focused long-input summarization. However, error propagation is an inherent flaw of two-stage methods. Besides, incoherent input texts may cause the generator to make more factual errors [7]. End-to-end models split the source meeting into fixed-length overlapping segments, each of which is separately appended to the query and encoded using a standard Transformer [16]. These encodings are then concatenated into a single embedding sequence and passed to a decoder that generates the summary.

End-to-end models solve the problem of error propagation and incoherent input texts. However, the length of the meeting is too long, and only a small part of it is related to the query. Therefore, it's a difficult task for the end-to-end model to recognize which contents of the meeting are more important for answering the query. Besides, the length of the query is too small, and the information of the query is diluted after being concatenated with the long meeting. As a result, many summaries generated by the current end-to-end models are irrelevant to the query according to our observations.

To address these challenges, inspired by current studies about adding extractive summaries as guidance for abstractive summarization [8], we propose IGQFS, an importance scores guided query focused end-to-end summarization model. Firstly, we consider one statement by a participant in the meeting transcript as an utterance, and compute the ROUGE scores between each utterance and the gold summary, which can be regarded as how important this utterance is for generating the query focused summarization. Secondly, we add a special token at the beginning of each utterance, the slightly modified input sequence is input into the encoder after being split into segments, and the last-layer hidden vector of each special token can be regarded as a representation of its following utterance.

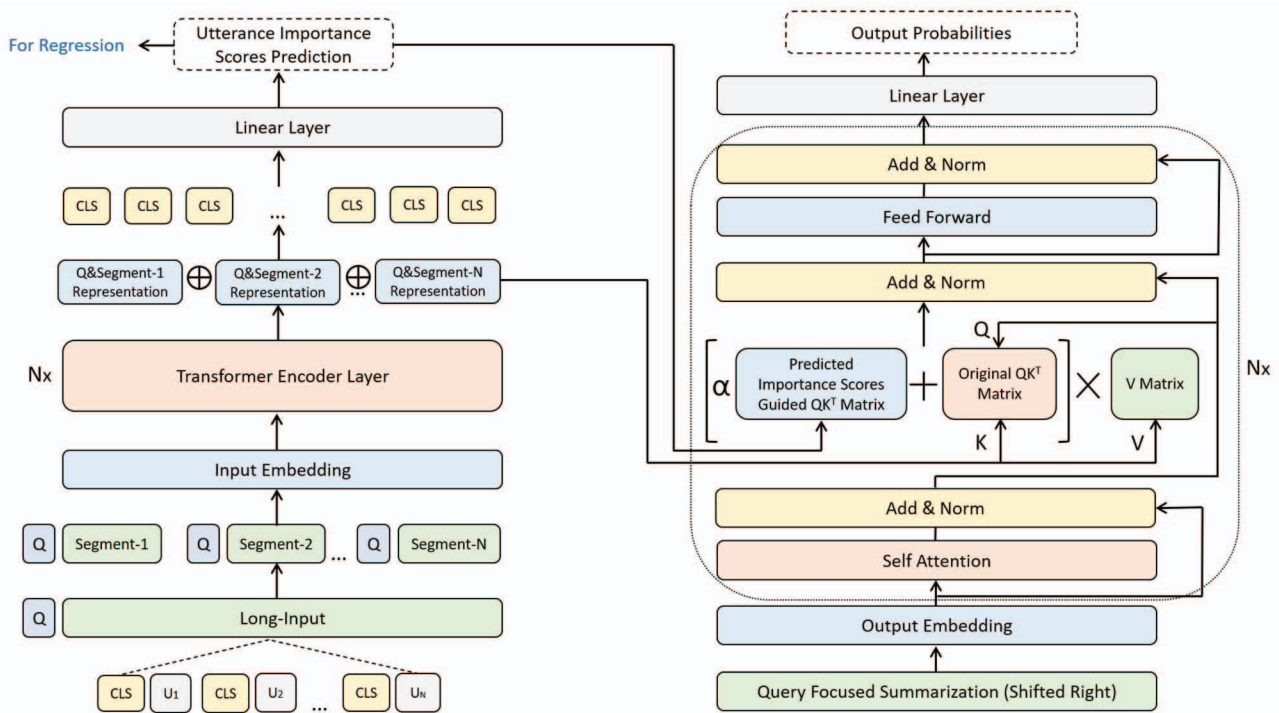


Figure 1 The Framework of the Proposed IGQFS

We input all the last-layer hidden vectors of special tokens into a linear layer, in order to predict the ROUGE scores between their following utterances and the gold summary. Finally, we modify the cross-attention module of the decoder according to the predicted utterance importance scores, aiming to teach the decoder to pay more attention to the more important utterances while decoding.

We believe our utterance importance prediction auxiliary task can help the model have the ability to recognize the relative importance of utterances, and help the model strengthen the interaction between query and meeting utterances implicitly, because it's necessary to interact with the query for each utterance to predict the ROUGE score between itself and the gold query focused summary. Experimental results on QMSum dataset show that our modified cross-attention module can help the model generate summaries more related to the queries, and the proposed method performs better than other end-to-end baseline models.

2 Related work

2.1 Query Focused Summarization

Query focused summarization aims to generate a summary depending on both the given query and the given text. The first query focused summarization dataset was released in DUC2005 summarization evaluation [2]. In the next two years, another two QFS datasets DUC2006 and DUC2007 were released. DUC2005-2007 are small-scale QFS datasets with only 50 questions per dataset, and they can be used to evaluate the performance of summarization models

before deep learning is widely used. However, after most of the summarization models are based on deep learning, the amount of data is too small to train neural networks with these datasets.

Therefore, some works [3,4,17] try to train a QFS model based on weakly supervised methods using data-rich question answering, machine reading comprehension, and generic text summarization datasets. Xu et al [3] propose a coarse-to-fine query focused extractive summarization model, whose core extraction module is trained with QA dataset WikiQA and MRC dataset SQuAD. Lapata et al [4] propose a query focused abstractive summarization model trained with generic text summarization datasets, regarding generic text summary as a response to some queries. They mask some keywords in the generic text summary, and the masked summary is regarded as a query, getting a large amount of QFS training data.

2.2 Long Document Summarization

Long document summarization addresses the setting where source document length exceeds the input limits of standard pretrained models. Main methods for long document summarization include sparse-attention based methods and extractive-abstractive methods. Sparse-attention based methods [15,18] use pretrained models based on sparse attention, like Longformer [15], which reduces the quadratic complexity of Transformer into sub-quadratic complexity, allowing much longer input. Extractive-abstractive methods first reduce the source text into a shorter subset, allowing it to fit within the input token limit of a Transformer-based pretrained model. It's necessary to train an extractor to reduce the

source text and train a generator to generate final summary based on the reduced source text.

3 IGQFS

The overall framework of IGQFS is illustrated in Figure 1. IGQFS is based on classic encoder-decoder architecture, like BART [9]. However, we adopt a segmentation strategy for long-input meeting transcripts (in Section 3.1). After encoding the input sequence, we use the last-layer hidden vector of each special token to predict the importance scores of each utterance (in Section 3.2). We modify the cross-attention module of the decoder in order to generate summaries more related to the queries (in Section 3.3).

3.1 Data Preprocessing

We add a special token ([CLS]) at the beginning of each utterance, in order to get the vector representation of these special tokens after encoding, representing the semantic information of their following utterances. Besides, we compute the ROUGE scores between each utterance and the gold summary for the auxiliary task in section 3.2. Following the idea of Fusion-in-Decoder (FiD) [10] and previous end-to-end QFS model [6], we split the long meeting transcripts into fixed-length overlapping segments (50% overlap) and append the query before each segment.

3.2 Utterance Importance Scores Prediction

After encoding the input sequence using BART encoder, we can get the final-layer hidden state of each special token [CLS]: $h_i^{utt} \in R^d$, $i \in \{1, \dots, N\}$, N is the number of utterances. Then we input the hidden state into a linear layer to predict the importance score of its following utterance:

$$s_i^p = W_s h_i^{utt} + b_s \in R^1 \quad (1)$$

Where s_i^p is the predicted importance score for utterance i . Assume s_i^g is the ROUGE score between utterance i and the gold summary. We use MSELoss as the loss function of utterance importance scores prediction auxiliary task:

$$L_{isp} = \frac{1}{N} \sum_{i=1}^N (s_i^p - s_i^g)^2 \quad (2)$$

3.3 Importance Scores Guided Cross Attention

In the Transformer-based pretrained encoder-decoder model, each layer of the decoder consists of three modules: self-attention module, cross-attention module and feed forward module. In cross attention module, the queries come from the output of previous self-attention module, the memory keys and values come from the output of the encoder. Through effective training, the model can learn to identify the relative importance of the words in the input by itself, and pay more attention to the more important words while decoding, similar to encoder-decoder attention mechanisms in sequence-to-sequence models [11]. However, the

number of QMSum's query-summary pairs is quite small, the length of encoded input sequence is quite long, and the summary we need is just related to the query, it's hard for a model to learn to pay how much attention to the words of a long input sequence using a small-scale dataset.

Our auxiliary task in Section 3.2 helps the model have the ability to recognize the relative importance of utterances, and we want to apply this ability to cross-attention module to help the model learn to pay more attention to the more important words while decoding more easily. In detail, original cross attention of i -th layer can be computed as:

$$A_i = \text{softmax} \left(\frac{Q_i K^T}{\sqrt{d_k}} \right) V \quad (3)$$

$Q_i \in R^{n \times d_k}$, $K \in R^{m \times d_k}$, $V \in R^{m \times d_v}$, n is the length of target summary, and m is the length of concatenated embedding sequence. We add another matrix $M \in R^{n \times m}$ to $\frac{Q_i K^T}{\sqrt{d_k}}$:

$$A_i = \text{softmax} \left[(1 - \alpha) \frac{Q_i K^T}{\sqrt{d_k}} + \alpha M \right] V \quad (4)$$

Through the auxiliary task in section 3.2, we can predict the importance score of each utterance. And in this section, we use these predicted utterance importance scores to construct a vector of dimension m . In detail, as shown in Figure 1, after encoding, for Segment- i , we can get the encoding representation E_i of Q&Segment- i , $E_i \in R^{L \times d}$, and we want to construct a vector V_i of dimension L (same with the length of Q&Segment- i): For each position in V_i , if the word of this position in Q&Segment- i is a word of query, we fill this position with the maximum predicted importance score of all utterances in Segment- i ; if the word of this position in Q&Segment- i is a word of an utterance of the meeting, we fill this position with the predicted importance score of this utterance. We concatenate the vector V_i of each segment, and get a vector of dimension m . Finally, we repeat this vector with n times, and get matrix $M \in R^{n \times m}$. Adopting the idea of teacher forcing, in the training stage, we use the gold utterance importance scores (true ROUGE value) to fill the matrix M , instead of the predicted utterance importance scores.

3.4 Learning Objectives

In the training stage, IGQFS learns to predict utterance importance scores and generate query focused summary simultaneously. For summary generation task, we use the NLLLoss L_{gen} as previous work. The overall training objective is:

$$L_{overall} = L_{gen} + \lambda L_{isp} \quad (5)$$

λ is a hyperparameter to balance the loss components.

4 Experiments

4.1 Experiment Details

Table I Result Comparisons on QMSum test dataset

	ROUGE-1	ROUGE-2	ROUGE-L
Random	12.03	1.32	11.76
Pointer-Generator Network[12]	28.74	5.98	25.13
BART[9]	29.20	6.37	25.49
DIALOGLM[13]	33.69	9.32	30.01
DYLE[14]	34.42	9.71	30.10
Longformer[15]	34.18	10.32	29.95
SEGENC[6]	37.05	13.03	32.62
SEGENC (Our Implementation)	36.79	13.02	32.46
IGQFS	37.53	13.61	33.16

Table II Ablation study of IGQFS on QMSum development dataset

	ROUGE-1	ROUGE-2	ROUGE-L
IGQFS	38.00	12.76	33.43
w/o modifying cross attention	37.57	12.59	32.99
w/o modifying cross attention, w/o auxiliary task	37.50	12.19	32.77
w/o modifying cross attention, w/o auxiliary task, w/o adding special tokens	37.55	12.21	32.76

We conduct experiments on QMSum dataset, with 1257, 272 and 279 query-summary pairs in its training, validation and test set respectively. The backbone encoder-decoder model is based on BART-large, and initialized with BART’s pre-trained parameters. QMSum is used to fine-tune backbone BART and additional linear layer on top of BART encoder. Models were trained for 10 epochs with final checkpoints selected based on the average of ROUGE- $\{1, 2, L\}$ (R-1, R-2, R-L) scores achieved on the validation set. Following previous work [6], we set segment size to 512, and set max number of segments to 32. While decoding, we used beam search with beam size of 4. In the test stage, models were automatically evaluated using the ROUGE- $\{1, 2, L\}$ metrics. We performed a grid search for determining the hyperparameter λ and α , and we found that when λ was set between 0.2 and 0.4, α was set between 0.1 and 0.2, IGQFS performed best. Finally, we set hyperparameter λ , α to 0.2 and 0.2 respectively.

4.2 Baselines

We compare IGQFS with several baselines: **Random** baseline randomly sample 10 turns of the meeting for each query as a summary. **Pointer-Generator Network** [12] is a popular sequence-to-sequence model with copy mechanism and coverage loss. **BART** [9] is a denoising pre-trained encoder-decoder model for language generation, translation and comprehension. **DIALOGLM** [13] adopts a window-based denoising approach and augments the model with sparse attention to process longer dialogue. **DYLE** [14] adopts an extractive-abstractive strategy for long-input summarization, allows dynamic snippet-level attention weights during decoding, and trains an extractor and a generator jointly. **Longformer** [15] replaces the quadratic self-attention mechanism of traditional Transformer with a memory-efficient version that combines local attention with sparse global attention,

which extends the maximum input size to 16,384 tokens. **SEGENC** [6] is a strong end-to-end baseline model, which splits the input text into fixed-length overlapping segments, each of which is separately appended to the query and encoded using a standard Transformer model.

4.3 Results & Analysis

We compare IGQFS with the above baselines, and the results are shown in Table I. Compared with strong end-to-end baseline SEGENC [6], our model improves 0.74, 0.59, 0.7 F1 scores in terms of ROUGE-1/2/L F1 scores [27]. Besides, we conducted ablation study to investigate the influence of various components of the IGQFS on the output summary, and the results are shown in Table II. We can conclude that both our utterance importance prediction auxiliary task and modified cross-attention module contribute to final performance, while utterance importance prediction auxiliary task contributes more to ROUGE-2 F1 scores and modified cross-attention module contributes more to ROUGE-1 & ROUGE-L F1 scores. Besides, we find that adding special token [CLS] has almost no impact on final performance.

5 Conclusions

In this paper, we propose to use utterance importance scores to guide an end-to-end model to generate summaries more related to the queries. In the training stage, our model IGQFS learns to predict utterance importance scores and generate query focused summary simultaneously. Besides, we modify the cross-attention module according to the predicted utterance importance scores, in order to help the model learn to pay more attention to the more important utterances while decoding more easily. Experimental results on QMSum dataset show our method performs better than the baseline end-to-end model and ablation study verifies the necessity of our two innovations.

References

- [1] Abigail See et al. Get To The Point: Summarization with Pointer-Generator Networks. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017:1073-1083.
- [2] Hoa Trang Dang. DUC 2005: Evaluation of Question-Focused Summarization Systems. Proceedings of the Workshop on Task-Focused Summarization and Question Answering. 2006:48-55.
- [3] Yumo Xu, Mirella Lapata. Coarse-to-Fine Query Focused Multi-Document Summarization. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. 2020:3632-3645.
- [4] Yumo Xu, Mirella Lapata. Generating Query Focused Summaries from Query-Free Resources. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. 2021:6096-6109.
- [5] Ming Zhong, Da Yin et al. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021:5905-5921.
- [6] Jesse Vig et al. Exploring Neural Models for Query-Focused Summarization. Findings of the Association for Computational Linguistics: NAACL 2022.2022:1455-1468.
- [7] Huan Yee Koh, Jiaxin Ju et al. How Far are We from Robust Long Abstractive Summarization. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. 2022:2682-2698.
- [8] Zi-Yi Dou et al. GSum: A General Framework for Guided Neural Abstractive Summarization. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021:4830-4842.
- [9] Mike Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020:7871-7880.
- [10] Gautier Izacard and Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. 2021: 874–880.
- [11] Yonghui Wu et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144, 2016.
- [12] Abigail See et al. Get To The Point: Summarization with Pointer-Generator Networks. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017:1073-1083.
- [13] Ming Zhong, Yang Liu et al. DIALOGLM: Pre-trained Model for Long Dialogue Understanding and Summarization. The Thirty-Sixth AAAI Conference on Artificial Intelligence. 2022:11765-11773.
- [14] Ziming Mao et al. DYLE: Dynamic Latent Extraction for Abstractive Long-Input Summarization. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022:1687-1698.
- [15] Iz Beltagy et al. Longformer: The Long-Document Transformer. arXiv preprint arXiv:2004.05150, 2020.
- [16] Ashish Vaswani et al. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017:5998–6008.
- [17] Md Tahmid Rahman Laskar et al. WSL-DS: Weakly Supervised Learning with Distant Supervision for Query Focused Multi-Document Abstractive Summarization. Proceedings of the 28th International Conference on Computational Linguistics. 2020:5647-5654.
- [18] Manzil Zaheer et al. Big Bird: Transformers for Longer Sequences. 34th Conference on Neural Information Processing Systems. 2020
- [19] Chenguang Zhu et al. A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining. Findings of the Association for Computational Linguistics: EMNLP 2020.2020:194-203.
- [20] Yusen Zhang et al. An Exploratory Study on Long Dialogue Summarization: What Works and What's Next. Findings of the Association for Computational Linguistics: EMNLP 2021.2021:4426-4433.
- [21] Chris Kedzie, Kathleen McKeown et al. Content Selection in Deep Learning Models of Summarization. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018:1818-1828.
- [22] Yang Liu. Fine-tune BERT for Extractive Summarization. arXiv preprint arXiv:1903.10318, 2019.
- [23] Yang Liu, Mirella Lapata. Text Summarization with Pretrained Encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019:3730–3740.
- [24] Ming Zhong, Pengfei Liu et al. Extractive Summarization as Text Matching. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020:6197-6208.
- [25] Jiacheng Xu, Zhe Gan, Yu Cheng, Jingjing Liu. Discourse-Aware Neural Extractive Text Summarization. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020:5021-5031.
- [26] Jingqing Zhang, Yao Zhao, Mohammad Saleh and Peter Liu. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. International Conference on Machine Learning. 2020: 11328–11339.
- [27] Chin-Yew Lin et al. ROUGE: A Package for Automatic Evaluation of Summaries. Text Summarization Branches Out. 2004: 74-81.