Interspeech 2022
18-22 September 2022, Incheon, Korea

Dynamic Sliding Window Modeling for Abstractive Meeting Summarization

Zhengyuan Liu[†‡], *Nancy F. Chen*[†‡]

[†]Institute for Infocomm Research, A*STAR, Singapore [‡]CNRS@CREATE, Singapore

liu_zhengyuan,nfychen@i2r.a-star.edu.sg

Abstract

Summarizing spoken content using neural approaches has raised emerging research interest lately, as sequence-to-sequence approaches have improved abstractive summarization performance. However, summarizing long meeting transcripts remains challenging. Meetings are multi-party spoken discussions where information is topically diffuse, making it harder for neural models to distill and cover essential content. Such meeting summarization tasks cannot readily benefit from pre-trained language models, which typically have input length limitations. In this work, we take advantage of the intuition that the topical structure of meetings tends to correlate with the meeting agendas. Inspired by this phenomenon, we propose a dynamic sliding window strategy to elegantly decompose the long source content of meetings to smaller contextualized semantic chunks for more resourceful modeling, and propose two methods without additional trainable parameters for context boundary prediction. Experimental results show that the proposed framework achieves state-of-the-art abstractive summarization performance on the AMI corpus and obtains higher factual consistency on competitive baselines.

Index Terms: Spoken language summarization, dialogue summarization

1. Introduction

Abstractive summarization is one of the most challenging tasks in natural language processing, involving context understanding, information collecting, and language generation [1]. Various neural models, from pointer-generator networks [2] to pre-trained language models [3], have been proposed and brought significant improvement on the generation quality, especially in fluency and readability [4]. Although most prior work focuses on monologues like news [5, 6], summarizing dialogues is gaining research attraction [7, 8, 9, 10]. Meetings are a type of multi-party conversation [11] that set a high bar for information gathering, consolidation, and distilling. A typical meeting conversation covers multiple sub-topics and the average turn number is much higher than that of short conversations like social chats [12] and inquiring-answering [9]. Neural language generation quality also tends to degrade when faced with massive contextual information [13]. Moreover, when adopting Transformer-based pre-trained language models like BERT [14] and BART [3], the transcript length exceeds their maximum positional embedding limitations. Text truncation, the default approach to bypass such limitations, is ill-suited in the use case of meeting summarization, as it often results in irreversible loss of useful context. In recent studies of summarizing long monologic documents, researchers have proposed efficient Transformer attention schemes [15], multi-block aggregation [16], and static sliding window have also been investigated [17]. However, most previous approaches need to modify the original Transformer architecture and introduce additional training parameters, and it is not straightforward to adopt such methods on the off-the-shelf language backbones.

Meetings usually consist of a set of agenda items [11], where each item focuses on a specific topic. Thus the entire conversation is inherently organized at the topic level. This is also reflected in the human-written summaries, which condense key information for each discussion point. Therefore, we postulate the divide-and-conquer strategy can be useful to tackle the aforementioned challenges of abstractive meeting summarization. More specifically, we aim to decompose the long transcript into multiple segments, and obtain the final output by aggregating all summary snippets of each segment. In this work, we propose to enhance a sequence-to-sequence summarizer with a dynamic sliding window strategy in order to tackle the length limitation of Transformer-based language models. Unlike conventional fixed sliding window approaches [14], our proposed framework can dynamically decide the start position of each context segment during the generation process. To this end, we introduce two parameter-free methods for automatic context boundary prediction, which can be readily integrated into the existing language backbones without any layer modification and additional trainable parameters: (1) a learning-based approach that directly generates the supporting utterance; (2) an attention-based approach that uses decoder-to-encoder cross-attention scores as the context span pointer. While the proposed methods are inspired by the organizational features of meeting transcripts, the generality of them is not limited to the meeting domain, and is in principle applicable to any task that might face input length limitations when exploiting pre-trained language models. Experimental results on the representative corpus AMI [11] show that the summary generation benefits from our dynamic sliding window strategy and achieves state-of-the-art performance without additional in-domain pre-training and discourse information. Further sample analysis demonstrates that the enhanced framework performs better than the baseline model when considering factual consistency.

2. Related Work

Text summarization is mainly studied in abstractive and extractive paradigms [18]. For extractive summarization, traditional methods study various linguistic and statistical features via lexical [19] and graph-based modeling [20]. Much progress has been made by contextualized neural extractors [21, 22]. While abstractive summarization is challenging, its performance has achieved substantial improvements in the news domain [5]: sequence-to-sequence models were introduced for fluent generation [1], pointer-generator network [2] elegantly handles out-of-vocabulary issues, and large-scale pre-trained language models bring further performance improvement [3, 4].

Recently, neural summarization for spoken languages has become an emerging research area [12]. With the abundance of automatic meeting transcripts and practical value of real-world use cases, meeting summarization attracts much attention

10.21437/Interspeech.2022-121

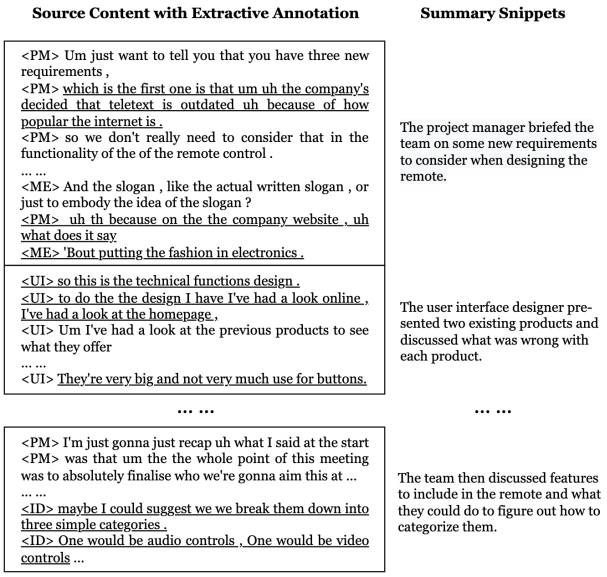

Figure 1: *One example of the meeting transcript and reference summary in the AMI corpus. Here we only show several extractive utterances (underlined) in three sub-topic chunks and their corresponding abstractive summary snippets.*

[11, 23, 24]. Based on the linguistic characteristics of meetings, some previous work focused on incorporating auxiliary information for better modeling meetings, such as dialogue act features [7], topic and vision features [25], dialogue discourse information [26]. The impact of domain terminologies has been investigated [27]. In this work, we focus on improving long transcript summarization via parameter-free approaches.

3. Meeting Transcript Analysis

In this section, we first conduct analysis on the representative meeting corpus AMI [11], where the participants work in a team and conduct meetings to discuss product design, development, and planning. There are four main speaker roles: a project manager (PM), a marketing expert (ME), an industrial designer (ID), and a user interface designer (UI). Following previous work [24], we split the whole dataset into train (97 transcripts), validation (20 transcripts), and test (20 transcripts) sets. The data statistics are shown in Table 1, where we count one utterance in the conversation as one sentence, and conduct word-level tokenization. Compared with news summarization benchmarks, the average length of meeting transcripts as well as the reference summaries are much larger (In our settings, we use the long version abstracts in the AMI corpus, as previous work [28]). Moreover, human-written summaries of news articles often concentrate on the first few parts of source content [29], thus the truncation with a fixed length does not affect the final performance significantly [30]. However, summarizing meetings requires grasping useful contextual information across the entire conversation. Text truncation will lead to information loss, as summary content tends to be more evenly spread out in meetings instead of having a positional bias as in news articles [31].

3.1. Organization Analysis

Since meetings are usually taken with a set of sub-topics, we analyze the organizational correlation of these sub-topics and sentences in meeting summaries. In the AMI corpus, for each meeting transcript, aside from the abstractive summary, there is an additional extractive annotation. As shown in Figure 1, human annotators were asked to choose sentences from the source conversation, as the supporting segments for each abstractive summary sentence. Based on this extractive annotation, we obtained the start/end supporting utterance indices in the conversation of each summary sentence. Then we sorted them by their occurrence order in the source content, and observed that 80% sentences in reference summaries match the same agenda order as in the source content. This finding suggests that when humans write a summary, they might refer to the source content and record the key points sequentially.

Table 1: *Data statistics of the news summarization benchmarks and the AMI meeting corpus.*

	CNN	DailyMail	NYT	AMI
Average Source Content Length:				
Sentence Level	33.98	29.33	35.55	288.7
Word Level	760.5	653.3	800.1	4757
Average Reference Summary Length:				
Sentence Level	3.59	3.86	2.44	17.55
Word Level	45.8	54.65	45.54	323.3

3.2. Segmentation Analysis

When adopting a divide-and-conquer strategy, only a part of the conversation will be extracted. To assess whether sufficient context information is provided by the segmented source content, we calculate the word level coverage with extractive annotations (see Figure 1) in the AMI corpus. With the start/end supporting utterance indices described in Section 3.1, we split the original transcripts into a number of segments. Then word-level recall is calculated between each summary sentence and its corresponding segment. The overall word-level coverage between summary snippets and their corresponding source segments is 74%, which is substantial since the abstractive summaries are often written via sentence paraphrasing and introducing some novel words [2, 31].

3.3. Summary Conversion

As the final output under a divide-and-conquer strategy is produced by aggregating summary snippets, to build the training ground-truth, we split each reference summary into multiple parts based on the analysis in Section 3.1 and Section 3.2. For each summary sentence, we constructed its context segment by using the corresponding extractive annotation. Then we ordered the summary sentences by their occurrence indices in the source conversation, and merged adjacent summary sentences to summary snippets if their context segments had a certain overlap. For summary sentences without extractive annotation, we reorganized them via calculating word-level overlap with existing content segments. Furthermore, we observed that some summary snippets started with a pronoun that refers to a precedent personal named entity. To maximize semantic integrity when producing one snippet, we used coreference resolution on the original summary; if one summary snippet starts with a pronoun, it will be replaced by its referring personal named entity.

4. Proposed Neural Meeting Summarizer

In this section, we elaborate on our framework for meeting summarization with the dynamic sliding window strategy.

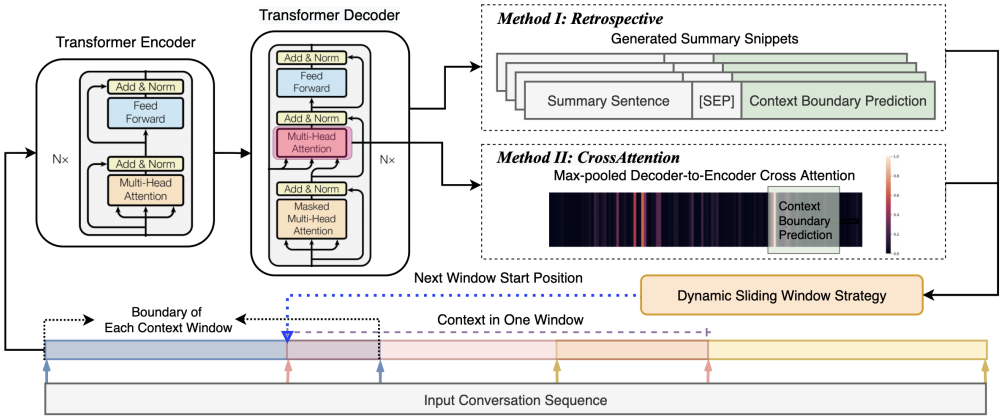

Figure 2: *Overview of the meeting summarization framework with the proposed dynamic sliding window strategy.*

4.1. Dynamic Sliding Window

The sliding window splits a long input into a number of shorter spans, processes the spans in order (e.g., from left to right), and adopts aggregation for the final output. It is a straightforward but useful method that is commonly applied to long sequence encoding [14], and it is generally controlled by two parameters: *window size* denotes the context span size, and *stride size* is the amount of movement at each sliding step. In previous work, these two parameters are often fixed. Here we propose a dynamic sliding window strategy, where the stride size at each sliding step is predicted by the model. More specifically, given the window size k, at each sliding step, one index j is selected (equals stride size is j) in the range of $[0, k-1]$ as the start position of the next window.

4.2. Base Neural Architecture

We use the Transformer-based sequence-to-sequence architecture [32], and select the large-scale pre-trained language backbone *'BART-base'* [3] for model initialization. The encoder consists of 6 stacked Transformer layers, where each layer has two sub-components: a multi-head self-attention layer and a position-wise feed-forward layer. Between the two sub-components, residual connection and layer normalization are used. The u-th encoding layer is formulated as:

$$\widetilde{h}^u = \text{LayerNorm}(h^{u-1} + \text{MHAttention}(h^{u-1})) \quad (1)$$

$$h^u = \text{LayerNorm}(\widetilde{h}^u + \text{FFN}(\widetilde{h}^u)) \quad (2)$$

where h^i is the i-th layer input. MHAttention, FNN, and LayerNorm are multi-head attention, feed-forward, and layer normalization, respectively.

The decoder consists of 6 stacked Transformer layers as well. In addition to the two encoding sub-components, the decoder performs another multi-head attention over the previous decoding hidden states, and all the encoded representations (i.e., cross-attention). Then, the decoder generates tokens in an auto-regressive manner from left to right.

4.3. Adopting Dynamic Sliding Window for Meeting Summarization

The proposed dynamic sliding window is a general design that can be applied to various neural architectures, and its overview is shown in Figure 2. In our summarization setting, the source content is a sequence with n tokens $C = \{w_1, w_2, ..., w_n\}$, and the summary is composed of m snippets $S = \{s_1, s_2, ..., s_m\}$. In a traditional sequence-to-sequence process, the whole source content C is fed to a model as input, and the output summary is generated at one decoding stage. With the dynamic sliding window strategy, the encoding-decoding process will be conducted in a number of steps, as shown in Figure 2. Given the window size k, at i-th sliding step, the start token index is i_{left}, and the end token index i_{right} is $i_{left} + k$. Then the input sequence of encoder at i-th step is $c_i = \{w_{i_{left}}, ..., w_{i_{right}}\}$. After encoding, the contextualized hidden representation $h_i = \{h_{i_{left}}, ..., h_{i_{right}}\}$ is fed to the decoder to generate a summary snippet, and all snippets are merged to form the final output.

4.4. Context Boundary Prediction

To achieve dynamic sliding prediction, one approach is to directly predict the stride size as in [33], where reinforcement learning is used to decide the value, and Schüller et al., [17] proposed to generate a special indicator for the moving operation. In this paper, we introduce two approaches that can be easily integrated into existing language models without any additional neural components: (1) a learning-based method *Retrospective* and (2) an attention-based method *Cross-Attention*.

Retrospective Since the supporting utterance list of each summary snippet is labeled, one way to obtain the dynamic strive step is to encourage the model to learn context boundary generation via the sequence-to-sequence process. More specifically, at i-th step, the decoder not only generates the summary snippet s_i, but also the last supporting utterance of its context segment that is described in Section 3.1 (they are concatenated with a special token *'[SEP]'*). Therefore, each predicted snippet s_i is formulated as $[\langle s \rangle, t^s_0, t^s_1, .., t^s_j,$ *[SEP]*, $t^c_0, t^c_1, .., t^c_k, \langle /s \rangle]$, where t^s and t^c denote the summary and the supporting utterance span respectively. The next context window will start from the boundary span t^c (see Figure 2, Method I).

Cross-Attention While the *Retrospective* approach does not require any additional trainable parameter, it needs a few more decoding steps for boundary generation. Here we propose another parameter-free method that has the same inference complexity as the original summarization process. As the Transformer decoder conducts a cross-attention on the encoded representation [32], we use the attentive salience as the boundary indicator. More specifically, we first normalize the cross-attention matrices extracted from top decoding layers with max-pooling, and then take the last utterance span that has the attentive score above average as the prediction of current context

Table 2: *ROUGE F1 scores on the AMI test set from baseline models and our framework. * Results are reported as in [28].*

	R-1	R-2	R-L
Extractive baselines*:			
TextRank	35.19	6.13	15.70
SummaRunner	30.98	5.54	13.91
Abstractive baselines*:			
Seq2Seq+Attention	36.42	11.97	21.78
Pointer-Generator	42.60	14.01	22.62
Sentence-Gated	49.29	19.31	24.82
TopicSeg-Summarizer	51.53	12.23	25.47
HMNet + domain pre-training	53.02	18.57	24.85
DDA-GCN + discourse info.	53.15	22.32	25.67
BART-SW-GoldSeg	53.42	22.57	28.52
BART-Truncate	48.31	17.52	20.13
BART-SW-Fixed	50.02	19.86	23.98
BART-SW-Retrospective (Ours)	52.83	21.77	26.01
BART-SW-CrossAttention (Ours)	**52.91**	**21.91**	**26.18**

boundary. Since the predicted supporting utterance/extracted attentive scores indicate the most salient span at the current generation step on the right, it is used to determine the amount of sliding movement (see Figure 2, Method II).

After all sliding steps, generated snippets $\{s_1, s_2, ..., s_m\}$ are concatenated as the final output summary S, and duplicate sentences will be removed in a post-processing step.

5. Experimental Results and Analysis

5.1. Configuration

The proposed framework was implemented using PyTorch. Learning rate was set at 2e-5, and AdamW optimizer was applied. We trained each model for 20 epochs, and selected best checkpoints based on ROUGE-2 validation score [34]. Input sequences were processed with the sub-word tokenization scheme as in [3], and repeated output sentences were removed.

Based on the summary conversion described in Section 3.3, we obtained 598/131/143 snippet-level samples as training, validation, and test set. At the training stage, to simulate the input noise during inference time, we randomly added k adjacent utterances ($5<k<15$) in each context chunk. At the testing stage, for each sample, the generation process of our ***BART-SW-Retrospective*** model starts with the first chunk of a default window size, which is initialized as 1024.[1] Then we used the sliding prediction described in Section 4.3 to obtain the next chunk, until the entire transcript was processed. We also assessed the ***BART-SW-Fixed*** model by fixing all window size at 1024, the ***BART-Truncate*** model by truncating the transcript of 1024 tokens, and the ***BART-SW-GoldSeg*** model by using the gold context segmentation.

5.2. Results on the AMI Corpus

Following previous work [28, 35], we used the ROUGE score [34] for evaluation, and reported ROUGE-1 (**R-1**), ROUGE-2 (**R-2**) and ROUGE-L (**R-L**). We selected strong baselines for extensive comparison including Pointer-Generator [2], Sentence-Gated [7], TopicSeg [25], HMNet [35], and DDA-GCN [28]. As shown in Table 2, for meeting summarization, abstractive approaches generally perform much better than extractive ones, due to the need of information collecting and paraphrasing. Our proposed models *BART-SW-Retrospective* and *BART-SW-CrossAttention* obtain similar generation scores, and they both outperform models with a fixed window (*BART-SW-Fixed*) and with text truncation (*BART-Truncate*), and this shows the effectiveness of adopting sliding window strategy for long transcript summarization. Moreover, our framework achieves the comparable performance of the contemporary state-of-the-art models, while those models require to modify the original Transformer architecture [8], or use additional in-domain pre-training and discourse information [28].

[1]The token-level maximum input length of the language backbone BART [3] is set at 1024 by default.

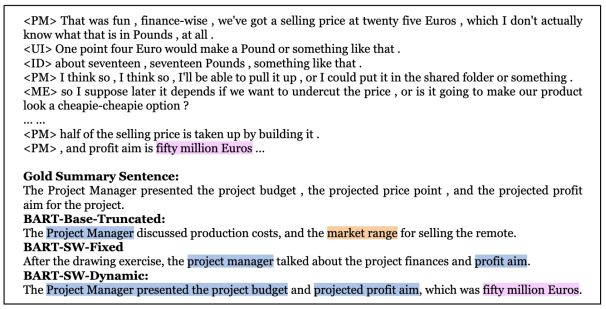

Figure 3: *One summarization example. Spans in blue and pink are consistent with the gold summary and source content. Spans in yellow are factually inconsistent.*

5.3. Factual Consistency Analysis

We first conducted a text quality analysis on the generated summaries across models. As shown in Figure 3, based on the strong generation capability of the language backbone, all BART-based models can produce fluent and grammatically correct summaries. While the model with text truncation achieves relatively acceptable ROUGE scores (as reported in Table 2), it produces sentences that are factually inconsistent with the source content, as shown in Figure 3. We speculate that this is caused by over-fitting the limited training samples, and we observed that some phrases that are irrelevant to the source content are repeated in summary generation. In contrast, as our proposed framework can produce the final summary based on relevant context segments without truncation, it performs better than the base model considering factual consistency. We observed a 37% relative reduction of inconsistency errors on the AMI test set via human evaluation. Additionally, we conducted a stride prediction assessment. For the *Retrospective* method described in Section 4.3, the predicted context boundaries are expected to be located closely to the ground-truth. With the best checkpoint, we observed that the average utterance-level distance between gold boundary span and model prediction is 2.7 (48 characters), indicating that the model is able to predict the correct start position at each sliding step.

6. Conclusion

In this work, to tackle the challenges of summarizing lengthy meeting transcripts, we introduced a dynamic sliding window strategy to resourcefully decompose the long source content to smaller contextualized semantic chunks for summarization modeling, and proposed two methods without any additional training parameters for context boundary prediction. Our experiments on the AMI corpus achieved state-of-the-art abstractive summarization performance, and we also obtained higher factual consistency on competitive baselines.

7. References

[1] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of EMNLP*, Sep. 2015. [Online]. Available: https://www.aclweb.org/anthology/D15-1044

[2] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of ACL*, 2017.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of ACL*, Jul. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.703

[4] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *Proceedings of EMNLP*, 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-1387

[5] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of NeurIPS*, 2015.

[6] N. F. Chen, B. Ma, and H. Li, "Minimal-resource phonetic language models to summarize untranscribed speech," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8357–8361.

[7] C.-W. Goo and Y.-N. Chen, "Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts," in *IEEE SLT*, 2018.

[8] C. Zhu, R. Xu, M. Zeng, and X. Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," *Proceedings of EMNLP*, November 2020. [Online]. Available: https://www.microsoft.com/en-us/research/publication/end-to-end-abstractive-summarization-for-meetings/

[9] Z. Liu, A. Ng, S. Lee, A. T. Aw, and N. F. Chen, "Topic-aware pointer-generator networks for summarizing spoken conversations," in *IEEE ASRU*. IEEE, 2019.

[10] Z. Liu and N. Chen, "Controllable neural dialogue summarization with personal named entity planning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 92–106.

[11] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of the 5th Measuring Behavior*, vol. 88. Citeseer, 2005.

[12] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Nov. 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-5409

[13] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *Proceedings of ACL*, 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL*, Jun. 2019. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[15] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[16] Q. Grail, J. Perez, and E. Gaussier, "Globalizing bert-based transformer architectures for long document summarization," in *Proceedings of EACL*, 2021.

[17] L. Schüller, F. Wilhelm, N. Kreiling, and G. Glavaš, "Windowing models for abstractive summarization of long texts," *arXiv preprint arXiv:2004.03324*, 2020.

[18] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proceedings of EMNLP*, Oct.-Nov. 2018. [Online]. Available: https://www.aclweb.org/anthology/D18-1443

[19] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of SIGIR*, ser. SIGIR '95. New York, NY, USA: Association for Computing Machinery, 1995. [Online]. Available: https://doi.org/10.1145/215206.215333

[20] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, 2004.

[21] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of AAAI*, 2017.

[22] C. Kedzie, K. McKeown, and H. Daume III, "Content selection in deep learning models of summarization," in *Proceedings of EMNLP*, Oct.-Nov. 2018. [Online]. Available: https://www.aclweb.org/anthology/D18-1208

[23] G. Murray, G. Carenini, and R. Ng, "Generating and validating abstracts of meeting conversations: a user study," in *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics, Jul. 2010. [Online]. Available: https://aclanthology.org/W10-4211

[24] G. Shang, W. Ding, Z. Zhang, A. Tixier, P. Meladianos, M. Vazirgiannis, and J.-P. Lorré, "Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization," in *Proceedings of ACL*, Jul. 2018. [Online]. Available: https://www.aclweb.org/anthology/P18-1062

[25] M. Li, L. Zhang, H. Ji, and R. J. Radke, "Keep meeting summaries on topic: Abstractive multi-modal meeting summarization," in *Proceedings of ACL*, Jul. 2019. [Online]. Available: https://www.aclweb.org/anthology/P19-1210

[26] X. Feng, X. Feng, B. Qin, and T. Liu, "Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks," *arXiv preprint arXiv:2010.10044*, 2020.

[27] J. J. Koay, A. Roustai, X. Dai, D. Burns, A. Kerrigan, and F. Liu, "How domain terminology affects meeting summarization performance," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5689–5695. [Online]. Available: https://aclanthology.org/2020.coling-main.499

[28] X. Feng, X. Feng, B. Qin, X. Geng, and T. Liu, "Dialogue discourse-aware graph convolutional networks for abstractive meeting summarization," *arXiv preprint arXiv:2012.03502*, 2020.

[29] Z. Liu, K. Shi, and N. Chen, "Conditional neural generation using sub-aspect functions for extractive news summarization," in *Findings of EMNLP*, 2020.

[30] T. Jung, D. Kang, L. Mentch, and E. Hovy, "Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization," in *Proceedings of EMNLP-IJCNLP*, 2019.

[31] W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," in *Proceedings of the EMNLP*, Nov. 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-1051

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NeurIPS*, 2017.

[33] H. Gong, Y. Shen, D. Yu, J. Chen, and D. Yu, "Recurrent chunking mechanisms for long-text machine reading comprehension," in *Proceedings of ACL*, Jul. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.603

[34] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Jul. 2004. [Online]. Available: https://www.aclweb.org/anthology/W04-1013

[35] J. Chen and D. Yang, "Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization," in *Proceedings of EMNLP*, Nov. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.336