ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095019

SPEECH SUMMARIZATION OF LONG SPOKEN DOCUMENT: IMPROVING MEMORY EFFICIENCY OF SPEECH/TEXT ENCODERS

Takatomo Kano[1], *Atsunori Ogawa*[1], *Marc Delcroix*[1], *Roshan Sharma*[2], *Kohei Matsuura*[1], *and Shinji Watanabe*[2]

[1]NTT Corporation, Japan, [2]Carnegie Mellon University, Pittsburgh, USA,

ABSTRACT

Speech summarization requires processing several minute-long speech sequences to allow exploiting the whole context of a spoken document. A conventional approach is a cascade of automatic speech recognition (ASR) and text summarization (TS). However, the cascade systems are sensitive to ASR errors. Moreover, the cascade system cannot be optimized for input speech and utilize para-linguistic information. Recently, there has been an increased interest in end-to-end (E2E) approaches optimized to output summaries directly from speech. Such systems can thus mitigate the ASR errors of cascade approaches. However, E2E speech summarization requires massive computational resources because it needs to encode long speech sequences. We propose a speech summarization system that enables E2E summarization from 100 seconds, which is the limit of the conventional method, to up to 10 minutes (i.e., the duration of typical instructional videos on YouTube). However, the modeling capability of this model for minute-long speech sequences is weaker than the conventional approach. We thus exploit auxiliary text information from ASR transcriptions to improve the modeling capabilities. The resultant system consists of a dual speech/text encoder decoder-based summarization system. We perform experiments on the How2 dataset showing the proposed system improved METEOR scores by up to 2.7 points by fully exploiting the long spoken documents.

Index Terms— end-to-end modeling, long spoken document, memory efficient encoders, dual speech/text encoder

1. INTRODUCTION

Speech summarization is a technology that generates an abstractive summary from a lengthy spoken document, such as an introduction video. Unlike other speech processing tasks, such as automatic speech recognition (ASR) or speech translation, generating an accurate summary requires access to the entire content. Since speech signals are very long sequences, the speech summarization task is particularly challenging as it requires handling hundred to tens of thousands of input frames to process several minute-long spoken documents.

A conventional approach to tackle this problem consists of first performing ASR and then text summarization (TS) [1–3]. ASR can transcribe an entire spoken document by performing utterance-wise recognition. TS can operate on the entire transcribed spoken document because the text sequences of an utterance are much shorter than speech. Such a cascade approach provides a modular system where we can optimize the components for each sub-task on dedicated datasets. It can exploit the entire context of the document to generate summaries [4, 5]. However, the cascade system cannot optimize the summary for the input speech and utilize prosodic information. Moreover, ASR systems inevitably introduce errors that can affect the quality of the summaries [1, 2, 6].

Table 1. Summarization performance (ROUGE-L score [13]) on the How2 dataset [14] of a Transformer TS with truncated input text.

	10%	30%	60%	90%	100%
ROUGE-L	10.0	18.9	45.3	50.8	51.2

Recently, an end-to-end (E2E) speech summarization [7] that directly generates a summary from speech has been proposed as an alternative to cascade systems. An E2E system consists of a speech encoder module that extracts embedding from the speech signal and a decoder module that generates a summary from the speech embedding in an autoregressive manner. The system is based on the Transformer architecture [8], which has become the standard for many speech and language processing tasks. E2E systems offer the possibility of optimizing the whole system for the summarization task, eliminating the intermediate ASR, therefore, avoiding the impact of ASR errors. Moreover, the E2E system has the possibility to use speech features such as pitch and power to determine the important point of the spoken document. However, E2E speech summarization needs to encode very lengthy speech signals because the system must process all utterances simultaneously, as in the TS model. This long speech encoding requires massive computational resources and memory, especially during training. The Transformer performs better than alternatives that use, e.g., long short-term memory networks [9] and can process the entire sequence at once using self-attention; however, the self-attention memory usage and computation increase quadratically with the length of the sequences [8, 10–12].

In a previous work [7], E2E speech summarization was achieved using Longformer [12] architecture for the speech encoder, which limits the range of the self-attention to reduce the amount of computation. However, it also impedes considering a long context in the self-attention module, which may be an issue for summarization. Consequently, the approach has only been applied to data truncated up to 100 seconds to avoid out-of-memory (OOM) issues during training [7]. Restricting the input of the speech summarization can have a significant impact on performance. For example, Table 1 shows the Recall-Oriented Understudy for Gisting Evaluation-Longest (ROUGE-L) score of TS when truncating the length of the text. We observe a clear performance drop when the TS system sees only part of the input document (e.g., 60 % or less). This preliminary experiment reveals the importance of developing speech summarization systems exploiting long spoken documents.

In this paper, we propose extending the E2E speech summarization system proposed in [7] to allow for handling longer spoken documents. First, we investigate memory- and computationally-efficient

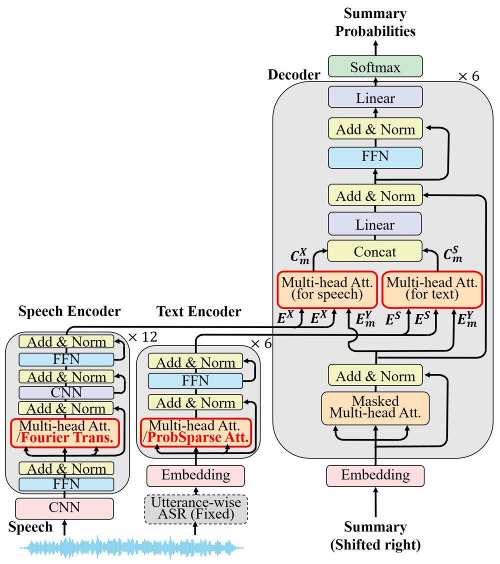

Fig. 1. Overview of the proposed model of dual speech/text encoders model. The FNet and Informer are memory efficient encoders.

alternatives to the self-attention module of the speech encoder, such as FNet [15]. The FNet changes the self-attention function by the 2D Fourier transform. Thus the FNet does not have costly project mappings for the query, key, and value. This modification drastically reduces the memory consumption, thereby allowing for the training of an E2E speech summarization system with input of up to 10 minutes. However, the FNet has less modeling power than the standard self-attention module, which may limit the performance [15].

As a second contribution, we propose using the text sequence provided by an ASR system as auxiliary information to help model a long context and strengthen the FNet encoder-based model. We introduce a dual (speech and text) encoder model that can combine these two modalities using the decoder cross attention mechanism to align them to the output summary.

Finally, as a side contribution, we investigate the recently proposed Informer model [16] for the text encoder. The Informer model performs efficient attention by sampling the important features of the input sequence. Consequently, this model could be well-suited for the summarization task. To the best of our knowledge, this is the first usage of this model for the summarization task.

The proposed dual-encoder model outperforms text- and speech-only approaches for both conventional and memory-efficient architectures. It combines the advantages of cascade and E2E speech summarization systems, i.e., it can process long spoken documents and optimize E2E.

2. PROPOSED DUAL ENCODER-BASED SPEECH SUMMARIZATION SYSTEM

We first introduce the dual speech and text encoders summarization system and then introduce the different efficient encoders.

2.1. Overall system configuration

Fig. 1 is a diagram of the proposed speech summarization system with the dual speech and text encoders. It is based on the Transformer architecture [8] with encoder and decoder modules, which we modified to include both speech and text encoders shown in left side of Fig. 1. The proposed model with only the speech encoder is similar to the E2E speech summarization system in [7]. The system with only the text encoder corresponds to a cascade speech summarization system since the input text consists of the transcribed text obtained with a pretrained ASR system that recognizes each utterance of the spoken document at a time.

We denote by $X = [x_1, ..., x_L]$ the input speech of the filter-bank and pitch features, and by $S = [s_1, ..., s_N]$ the corresponding Byte-Pair Encoding (BPE) [17] token of the ASR results, where L and N are the number of ASR result's speech frames and words, respectively. We denote by $Y = [y_1, ..., y_m, ..., y_M]$ BPE token of the output summary, where M is the length of the summary. In general, we have $L \gg N \gg M$.

As shown in the left side of Fig. 1, the first part of the model consists of the speech and text encoders that generate embeddings from the speech and text in the encoder as follows:

$$E^X = \text{Enc}_{\text{sp}}(X), \tag{1}$$

$$E^S = \text{Enc}_{\text{txt}}(S), \tag{2}$$

where $\text{Enc}_{\text{sp}}(\cdot)$ and $\text{Enc}_{\text{txt}}(\cdot)$ denote the speech and text encoders, and $E^X \in \mathbb{R}^{L\times D}$ and $E^S \in \mathbb{R}^{N\times D}$ denote the embeddings of speech and text, respectively. The D denotes the dimension of the embedding vector.

Then, as shown in the right side of Fig. 1, the decoder accepts these two embedding sequences and generates the summary autoregressively. However, since the speech and text embedding sequences have different lengths and are not aligned, we perform the cross modal attention [18–20] as follows:

$$C_m^X = \text{Att}_{\text{sp}}(E^X, E_m^Y, E^X), \tag{3}$$

$$C_m^S = \text{Att}_{\text{txt}}(E^S, E_m^Y, E^S), \tag{4}$$

where C_m^X and C_m^S are the speech and text context vectors. $\text{Att}_{\text{sp}}(\cdot)$ and $\text{Att}_{\text{txt}}(\cdot)$ denote attention modules for speech and text. $E_m^Y = \text{Dec}(y_m)$ denotes the m-th embedding vector of summary Y and $\text{Dec}(\cdot)$ denotes the summary decoder process. After the attention, context vectors C_m^X and C_m^S are aligned and can thus be combined. In this work, we concatenate the two vectors and treating them as a single super vector as, $E_m = \text{Linear}(\text{Concat}(C_m^X, C_m^S))$, where $\text{Concat}(\cdot)$ denotes the concatenation operation. The entire model can be optimized by minimizing the cross-entropy loss between the estimated and reference summaries.

2.2. Efficient encoders

The encoder of the speech summarization model receives a very long sequence, creating a bottleneck when tackling a lengthy spoken document. Below, we review the baseline Transformer-based text encoder and Conformer-based speech encoder and then introduce computation and memory-efficient variations.

2.2.1. Transformer-based text encoder

First, we describe the baseline text encoder (Eq. (2)), which is borrowed from the Transformer model [1, 6, 8, 21]. The Transformer encodes the document S of the ASR results as follows:

$$S' = \text{Emb}(S) + \text{PE}(N) \tag{5}$$

$$E'^S = \text{Norm}(S' + \text{MHAtt}(S', S', S')) \tag{6}$$

$$E^S = \text{Norm}(E'^S + \text{FFN}(E'^S)), \tag{7}$$

where $\text{Emb}(\cdot)$ denotes the word embedding function, $\text{PE}(\cdot)$ the position encoding function, $\text{MHAtt}(\cdot)$ a multi-head attention, $\text{Norm}(\cdot)$ the layer normalization, and $\text{FFN}(\cdot)$ a feed-forward network. Eq. (5) is the input layer of the encoder, which is followed by multiple Transformer-blocks composed of Eq. (6) and (7). Here, we omit the layer index in the Transformer-block for simplicity. The attention function receives S' as query $Q \in \mathbb{R}^{L \times D}$, key $K \in \mathbb{R}^{L \times D}$, and value $V \in \mathbb{R}^{L \times D}$, and perform self-attention for modeling sequential information.

2.2.2. *Informer-based text encoder*

We consider a variation of the text encoder, which is based on the recently proposed Informer model [16]. The Informer model was proposed for forecasting a long series such as a weather report. It replaces the self-attention layer of the Transformer with ProbSparse attention that down-samples the query sequence Q based on information theory and uses only important queries for dot product. The Informer chooses the top u important vectors among L row vectors in $Q \in \mathbb{R}^{L \times D}$ based on the Kullback-Leibler divergence of the K given Q, and reconstructs a new query $\tilde{Q} \in \mathbb{R}^{u \times D}$ where $u \gg L$ [16]. This reduces the computation complexity to $\mathcal{O}(Du^2)$.

In this work, we modified the sampling parameter for the summarization task. The summary length is determined by the layout of the scenario, such as an abstract of a newspaper headline. Therefore, we considered the length of the summary text as a parameter given by the user. We designed the sampling number u to be determined by the lengths of the summary M and the ASR transcript N as $u_p = M + (N - M)^{\frac{(P-p)}{P}} - 1$. Here, p and P denote the encoder layer index and depth. During encoding, the sampling number u is asymptotically close to M from N.

2.2.3. *Conformer-based speech encoder*

Next we describe the baseline speech encoder (Eq. (1)), which consists of a Conformer-based encoder that is widely used for ASR [22, 23]. The Conformer processes the input speech sequence X using a convolution neural network (CNN) [24] as follows:

$$X' = \mathrm{LN}(\mathrm{ReLU}(\mathrm{CNN}(\mathrm{ReLU}(\mathrm{CNN}(X))))), \tag{8}$$

Here ReLU is an activation function, LN is a linear mapping function [25]. The remaining processing is similar to that of the Transformer, except that the input of $\mathrm{MHAtt}(\cdot)$ (Eq. (6)) is processed with the feed-forward layer, and then the output of the $\mathrm{MHAtt}(\cdot)$ is processed with a convolutional layer before being fed to the feed-forward layer of Eq. (7) as shown in Fig. 1 left side.

We tried using the Informer model for the speech encoder, but it did not work well for the ASR task in our preliminary experiment because the speech is a boundaryless continuous signal, so it is more difficult to find u important vectors than for the text sequences. Therefore, we utilize other efficient encoders described in the following sections.

2.2.4. *Longformer-based speech encoder*

The Longformer model was originally proposed for text summarization [12], and subsequently used for E2E speech summarization [7]. The only difference between the Transformer and Longformer models is the self-attention module. The Longformer reduces the size of the key $K \in \mathbb{R}^{L \times D}$ by masking out elements outside a window and generates a new key $K' \in \mathbb{R}^{J \times D}$, where J denotes the window size. The Longformer then performs the dot products using Q and K'. The Longformer improves self-attention efficiency by performing the local sequence's dot product. Therefore, the Longformer can reduce the calculation complexity to $\mathcal{O}(DLJ)$. On the other hand, the adaptation of the window function restricts only the local context to be considered. The width of this window is a hyperparameter. The longer the input sequence, the narrower range the attention can see. Thus, it becomes more challenging to represent the entire context.

2.2.5. *FNet-based speech encoder*

FNet [15] is a recently proposed approach to reduce the computation of the attention module. It has achieved performance approaching that of models such as Bidirectional Encoder Representations from Transformers [26] in the language modeling tasks. We proposed introducing it to replace the attention in the speech encoder of our dual speech summarization system.

FNet interprets the Q, K, and V's dot product as a convolution representation, and the FNet changes the Transformer's self-attention function as $\mathrm{Att}(Q) = F_{\mathrm{seq}}(F_{\mathrm{dim}}(Q))$. Here, F_{dim} and F_{seq} denote the Fourier transform of the hidden dimension direction and time direction. The computation complexity of FNet is $\mathcal{O}(DL \log L) + \mathcal{O}(DL \log D)$. In general, we have $L \gg D$; thus, in this work, we omit the complexity of $\mathcal{O}(DL \log D)$ for simplicity.

The advantage of this replacement is that, in addition to reducing the computational complexity to the log order of the sequence length, it saves many resources compared to other methods since the conventional query, key, and value projection matrices are no longer needed. In addition, it continues to be able to view the entire series. However, since the Fourier transform has no learnable parameters, it may have weaker modeling capabilities than other attention modules.

We used these memory-efficient Transformer-based encoders to build our proposed model. Each encoder differs from the Transformer/Conformer in the self-attention, and other processes are the same as the original Transformer/Conformer model.

3. SPEECH SUMMARIZATION EXPERIMENT

3.1. Experimental settings

We follow related works [1, 6, 7, 27] to perform How2 speech summarization experiments. The dataset [14] contains 2k hours of instructional videos with corresponding text transcripts, speech, and summaries. The original E2E speech summarization recipe for How2 considers only the first 100 sec of each video for summarization [7]. Here we also explore using the full spoken document, which amounts to up to 10 minutes[1]. The 100-second data is short and can be trained with Conformer, but the 10-minute video cannot be trained without memory-efficient encoders such as FNet.

We perform experiments based on the recipe released in the ESPnet toolkit [28][2]. All models are based on the Transformer architecture with different encoders. For all experiments, the number of the encoder layers is 12, and 6 for the speech and text encoders, respectively. The size of the hidden layers of the encoder and decoder is 512, and the number of multi-head is 4. We performed the following procedure to train the models. First, we trained ASR models using the Conformer and FNet-based encoders and a TS model using the Transformer and Informer-based encoders, respectively. Next, we tuned the trained ASR model for the E2E speech summarization system using a summary dataset. Finally, we added the trained TS encoder to the trained E2E speech summarization model to create the dual encoder-based model and tune the model.

We investigated different encoders for the TS, E2E, and dual models. The TS systems see the entire document as we input the ASR transcriptions of all the utterances of the spoken document. Because of their high memory consumption, the baseline E2E systems are trained and tested with only the first 100 sec of the documents. For the proposed system that uses the memory-efficient FNet encoder, we also retrained it on the entire spoken document. We evalu-

[1]In the dataset, only two recordings were longer than 10 minutes. We removed these two data from a dataset.

[2]https://github.com/espnet/espnet/egs2/_how2_2000h

Table 2. Summarization performance in terms of METEOR (ROUGE-1, ROUGE-2, ROUGE-L). Here, OOM denotes a failed experiment due to out of memory. The results for Longformer are taken from the original paper. The number of ASR morel's parameters is the same as the End-to-end speech summarization models 3 and 5, and the WER rates are 8.9% and 9.3 % respectively.

	Speech Enc.	Text Enc.	100 sec	Full	Complexity	Memory	Parameters
1	-	Transformer	-	27.4 (57.1, 39.2, 51.2)	$\mathcal{O}(DN^2)$	0.9 GB	64.4 M
2	-	Informer	-	27.8 (58.4, 41.2, 55.1)	$\mathcal{O}(Du^2)$	0.6 GB	64.6 M
3	Conformer	-	28.9 (60.0, 39.0, 56.6)	OOM	$\mathcal{O}(DL^2)$	18.4 GB	95.0 M
4	Longformer [7]	-	29.3 (60.7, 44.9, 56.1)	-	$\mathcal{O}(DLJ)$	7.14 GB	100.8 M
5	FNet	-	28.5 (59.8, 39.8, 56.6)	29.0 (61.9, 42.3, 58.8)	$\mathcal{O}(DL \log L)$	6.28 GB	81.9 M
6	Conformer	Informer	28.9 (61.8, 44.9, 56.5)	OOM	$\mathcal{O}(DL^2)$	18.9 GB	136.1 M
7	**FNet**	**Informer**	30.0 (62.6, 43.6, 59.7)	**32.0 (64.9, 46.5, 61.9)**	$\mathcal{O}(DL \log L)$	6.8 GB	123.5 M

ate results in terms of METEOR [29] and ROUGE-1, ROUGE-2 and ROUGE-Longest(ROUGE-L) scores [13].

3.2. Experimental results

Table 2 summarizes the experimental results in terms of summarization performance. It also shows the memory consumption of representative models for the forward and backward steps during training when using a batch size of 1, for an input speech consisting of L=10k frames (i.e., 100 sec), and a text length N=500 BPE tokens.

The first two systems in Table 2 consist of TS systems using Transformer and Informer-based encoders. We confirm that the memory usage of these models is moderate. The Informer-based encoder improves performance and reduces memory consumption, showing that it is a promising approach for TS.

Systems 3-5 are E2E speech summarization systems using Longformer [12], Conformer or FNet-based encoders. The conformer-based model has a high memory usage, preventing it from training on the full data. In contrast, the proposed Fnet-based system (system 5) has significantly lower memory consumption and model parameters. When trained and tested on the 100 sec, it performs slightly worse than the baseline Longformer and Conformer-based systems. However, we can retrain it on the full (10-minute) data, which ultimately performs better than the Conformer model, which can only handle 100 seconds of speech. These results demonstrate the importance of exploiting the entire spoken document for speech summarization.

The last part of Table 2 shows results with our proposed dual encoder model when using Conformer or FNet as a speech encoder. The Longformer outperforms the FNet slightly. However, the FNet has smaller model parameters and memory usage. Thus we use the FNet for the speech encoder. The dual encoder slightly improves the performance when using the Conformer-based speech encoder but has a larger impact on the FNet-based system. Using the dual encoder, we can recover the performance loss caused by the weaker speech encoder. The resultant system (system 7) outperforms the baseline systems (systems 3 and 4) on all metrics.

Next, full-length data evaluation shows the performance of the memory-efficient encoders. These models were trained on 100 seconds of data and then tuned on 10 minutes of data. The results show that the summary performance is lower with FNet than with Conformer. However, FNet can be trained on 10 minutes of data and ultimately performs better than the Conformer model, which can only handle 100 seconds due to the high-memory consumption. This results from the advantages of considering longer inputs due to the memory saving outweighing the performance loss due to the memory saving.

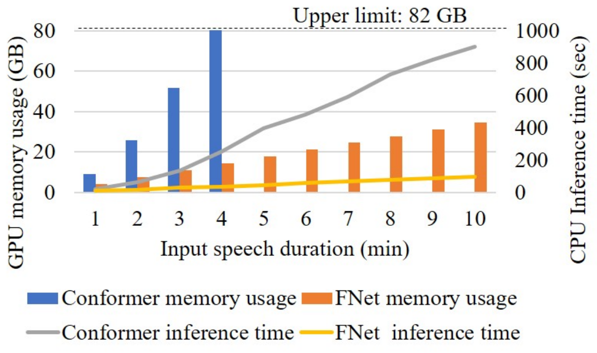

Fig. 2. GPU memory usage and inference time on CPU

3.3. Computation and memory usage

We conducted a further investigation of the proposed model, to determine its computational and memory efficiency. First, we investigated the relationship between the length of the input speech and the amount of memory consumed during summarization. Figure 2 shows memory usage on one training step for each model when the batch size is one. The Conformer model consumes GPU memory quadratically as the input speech becomes longer. If the input length is over four minutes, the memory limit of the A100 GPU is exceeded by 82 GB, making learning difficult. On the other hand, FNet consumes memory linearly with the length of the input data, and even for ten minutes of speech, the memory consumption is kept at about 30 GB. Therefore, the FNet can learn summarization using even longer speech data.

Next, Fig. 2 also shows the CPU inference time of the encoder. It shows that the Conformer complexity is a square of the computation time for the input length, so the computation time increases significantly with the input length. However, the FNet allows for a nearly linear increase in computation time and can infer about ten times faster when summarizing 10 minutes of speech. This analysis suggests that our proposed model could handle even longer spoken documents.

4. CONCLUSION

We proposed using a dual speech/text encoder model with computation- and memory-efficient encoders to perform the summarization of long spoken documents. The proposed system with the FNet speech encoder has low memory usage, allowing it to train on documents of up to 10 minutes. Consequently, the summarization accuracy improved because the model could exploit a longer context. In future work, we will tackle longer summarization tasks such as TED talk summary [6].

5. REFERENCES

[1] Shi-Yan Weng, Tien-Hong Lo, and Berlin Chen, "An effective contextual language modeling framework for speech summarization with augmented features," in *EUSIPCO*, 2020, pp. 316–320.

[2] Takatomo Kano, Atsunori Ogawa, Marc Delcroix, and Shinji Watanabe, "Integrating multiple ASR systems into NLP backend with attention fusion," in *IEEE ICASSP*, 2022, pp. 6237–6241.

[3] Shih-Hsiang Lin and Berlin Chen, "Improved speech summarization with multiple-hypothesis representations and kullback-leibler divergence measures," in *INTERSPEECH*, 2009, pp. 1847–1850.

[4] Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke, "Keep meeting summaries on topic: Abstractive multi-modal meeting summarization," in *ACL*, 2019, pp. 2190–2196.

[5] Alexandra Savelieva, Bryan Au-Yeung, and Vasanth Ramani, "Abstractive summarization of spoken and written instructions with BERT," *ArXiv*, 2020.

[6] Takatomo Kano, Atsunori Ogawa, Marc Delcroix, and Shinji Watanabe, "Attention-based multi-hypothesis fusion for speech summarization," in *ASRU*, 2021, pp. 487–494.

[7] Roshan Sharma, Shruti Palaskar, Alan W. Black, and Florian Metze, "End-to-end speech summarization using restricted self-attention," in *ICASSP*, 2022, pp. 8072–8076.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[9] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[10] Shigeki Karita, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, and Ryuichi Yamamoto, "A comparative study on transformer vs RNN in speech applications," in *ASRU*, 2019, pp. 449–456.

[11] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed, "Big bird: Transformers for longer sequences," in *NIPS*, 2020, pp. 17283–17297.

[12] Iz Beltagy, Matthew E. Peters, and Arman Cohan, "Longformer: The long-document transformer," *ArXiv*, 2020.

[13] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *ACL*, 2004, pp. 74–81.

[14] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze, "How2: A large-scale dataset for multimodal language understanding," *ArXiv*, 2018.

[15] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón, "Fnet: Mixing tokens with fourier transforms," in *NAACL*, 2022, pp. 4296–4313.

[16] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, 2021, pp. 11106–11115.

[17] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016, pp. 1715–1725.

[18] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiro Sumi, "Attention-based multimodal fusion for video description," in *ICCV*, 2017, pp. 4203–4212.

[19] Chunlei Wu, Yiwei Wei, Xiaoliang Chu, Weichen Sun, Fei Su, and Leiquan Wang, "Hierarchical attention-based multimodal fusion for video captioning," *Neurocomputing*, vol. 315, pp. 362–370, 2018.

[20] Ziwang Fu, Feng Liu, Hanyang Wang, Jiayin Qi, Xiangling Fu, Aimin Zhou, and Zhibin Li, "A cross-modal fusion network based on self-attention and residual structure for multimodal emotion recognition," *CoRR*, vol. abs/2111.02172, 2021.

[21] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu, "A survey of transformers," *ArXiv*, 2021.

[22] Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, Jing Shi, Shinji Watanabe, Kun Wei, Wangyou Zhang, and Yuekai Zhang, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP*, 2021, pp. 5874–5878.

[23] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020, pp. 5036–5040.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[25] Abien Fred Agarap, "Deep learning using rectified linear units (relu)," *ArXiv*, 2018.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[27] Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney, "Tight integrated end-to-end training for cascaded speech translation," in *SLT*, 2021, pp. 950–957.

[28] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "Espnet: End-to-end speech processing toolkit," in *INTERSPEECH*, 2018, pp. 2207–2211.

[29] Michael J. Denkowski and Alon Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *ACL*, 2014, pp. 376–380.