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/ICASSP49357.2023.10094993

LEVERAGING LARGE TEXT CORPORA FOR END-TO-END SPEECH SUMMARIZATION

Kohei Matsuura, Takanori Ashihara, Takafumi Moriya, Tomohiro Tanaka, Atsunori Ogawa, Marc Delcroix, Ryo Masumura

NTT Corporation, Japan

ABSTRACT

End-to-end speech summarization (E2E SSum) is a technique to directly generate summary sentences from speech. Compared with the cascade approach, which combines automatic speech recognition (ASR) and text summarization models, the E2E approach is more promising because it mitigates ASR errors, incorporates nonverbal information, and simplifies the overall system. However, since collecting a large amount of paired data (i.e., speech and summary) is difficult, the training data is usually insufficient to train a robust E2E SSum system. In this paper, we present two novel methods that leverage a large amount of external text summarization data for E2E SSum training. The first technique is to utilize a text-to-speech (TTS) system to generate synthesized speech, which is used for E2E SSum training with the text summary. The second is a TTS-free method that directly inputs phoneme sequence instead of synthesized speech to the E2E SSum model. Experiments show that our proposed TTS- and phoneme-based methods improve several metrics on the How2 dataset. In particular, our best system outperforms a previous state-of-the-art one by a large margin (i.e., METEOR score improvements of more than 6 points). To the best of our knowledge, this is the first work to use external language resources for E2E SSum. Moreover, we report a detailed analysis of the How2 dataset to confirm the validity of our proposed E2E SSum system.

***Index Terms*—** End-to-end speech summarization, synthetic data augmentation, multi-modal data augmentation, How2 dataset

1. INTRODUCTION

With the introduction of deep learning, automatic speech recognition (ASR) technology has made significant progress [1]. Despite its success, ASR transcriptions are not highly readable because they often contain fillers, disfluencies, or redundant expressions [2]. To generate more readable and informative text, speech summarization (SSum) technology has attracted much attention for meetings [3], educational videos [4], and patient-physician conversations [5].

In general, SSum has been achieved with the cascade approach, which combines ASR and text summarization (TSum) models, as shown in Fig. 1(a). Cascade SSum achieves high performances with a combined system of highly-accurate ASR models and TSum models pre-trained on large amounts of unpaired text data [6, 7]. Despite the success of the cascade approach, it suffers from ASR error propagation and a lack of nonverbal and acoustical information [8]. To tackle these problems, [9] fed a TSum model with N-best ASR hypotheses to mitigate the effect of ASR errors, and [10] added acoustic features to the input text.

More recently, end-to-end SSum (E2E SSum) has also been proposed to address the aforementioned problems. The E2E SSum system is similar to the E2E ASR system, which jointly optimizes acoustic and language models, and generates an abstractive summary directly with a single model as illustrated in Fig. 1(b). Since this method does not include text as an intermediate output, the model can utilize fully acoustic information to generate summaries and is probably free from ASR errors. [11] used this approach with restricted self-attention and reported that it performed better than the cascade approach.

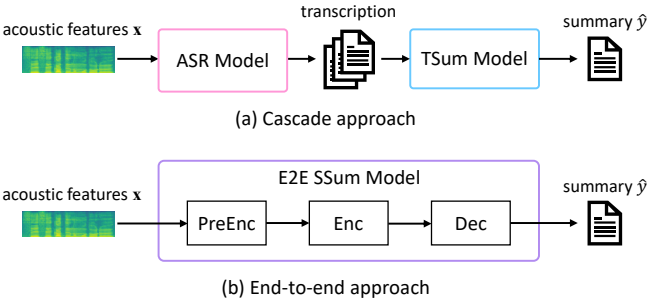

Fig. 1. Comparison between the (a) Cascade and (b) End-to-end (E2E) approaches for speech summarization.

However, the E2E SSum model requires a large number of costly speech-summary pairs, while most speech summarization tasks have a limited amount of pair data. We assume that the huge external language resources could improve the E2E SSum models as in other E2E speech processing fields [12–18].

In this paper, we attempt to leverage TSum data, i.e., Gigaword [19, 20], to enable the model to learn better linguistic representation. We propose a text-to-speech (TTS)-based speech augmentation approach that synthesizes speech from the text of an external TSum dataset to obtain pseudo-speech and summary paired data for E2E SSum training. Although the application of this technique is rather simple, synthesizing pseudo-speech requires high computational costs. To eliminate these costs, we also propose a method that extends the model with a phoneme pre-encoder, inspired by [16, 17]. The phoneme pre-encoder is separated from the speech pre-encoder when we input phoneme sequences instead of synthesized speech. This method requires much less computational cost and storage space than TTS-based data augmentation.

All experiments are conducted on the How2 dataset [21] composed of short video clips and their summaries. We performed a deep analysis of the corpus and found that part of the training and evaluation data were very similar, which could bias the results. We thus propose an evaluation scheme that filters out such data using quantitative metrics. With the proposed evaluation, we can attest summarization performance with more confidence.

Our major contributions are listed as follows:

1. Our best system achieves state-of-the-art performance on the How2 summarization task.

2. We confirm that external TSum data enables E2E SSum model improvements with TTS technology.
3. We propose a phoneme-based data augmentation method, resulting in lower computational costs than TTS-based one.
4. We suggest that filtering the evaluation set of the How2 dataset is important for assessing the quality of summarization abilities of models more reasonably.

2. RELATED WORK

Previous studies have demonstrated the effectiveness of utilizing external text data for E2E ASR and E2E speech translation systems. For E2E ASR systems, [12] generated additional speech-transcription data by using Tacotron2 [22] to synthesize acoustic features from external text data in the context of domain adaptation. Other studies utilized text data via discrete speech representations extracted with vq-wav2veq [13], latent representations of E2E ASR encoders [14], and phoneme sequences [16, 17]. Furthermore, [23] proposed a method called "Speech Chain" that jointly optimizes TTS and ASR models with additional unpaired text and speech. [24] expanded this method with a consistency loss. Similar ideas were also used to generate training data for E2E speech translation systems [18]. Our study is inspired by these successes and attempts to apply data augmentation approaches to an E2E SSum system.

3. E2E SSUM MODEL

As shown in Fig. 1, the E2E SSum system unifies the ASR and TSum models into a single model, while the cascade system processes input speech via transcriptions. Specifically, we adopt the transformer-based attentional encoder-decoder (AED) model [1] in this paper. First, the pre-encoder PreEnc embeds acoustic features $\mathbf{x}$ into a sub-sampled sequence of hidden representations $\mathbf{h}^{\mathrm{pre}}$:

$$\mathbf{h}^{\mathrm{pre}} = \mathrm{PreEnc}(\mathbf{x}), \tag{1}$$

where $\mathrm{PreEnc}(\cdot)$ is composed of 2-D convolutional neural networks (CNNs). Second, the encoder $\mathrm{Enc}(\cdot)$ calculates a latent representation $\mathbf{h}^{\mathrm{enc}}$ from $\mathbf{h}^{\mathrm{pre}}$:

$$\mathbf{h}^{\mathrm{enc}} = \mathrm{Enc}(\mathbf{h}^{\mathrm{pre}}, \mathbf{R}), \tag{2}$$

where $\mathrm{Enc}(\cdot)$ consists of Conformer [25] blocks that achieve high accuracy in the ASR task, and $\mathbf{R}$ denotes the relative positional embeddings. Finally, the decoder $\mathrm{Dec}(\cdot)$ receives $\mathbf{h}^{\mathrm{enc}}$ and previous outputs $\{\hat{y}_i\}_{i=1}^{l-1}$ as a clue to autoregressively infer each token:

$$\mathbf{h}_l^{\mathrm{dec}} = \mathrm{Dec}([\mathrm{Emb}(\hat{y}_i) + \mathrm{PE}_i]_{0\le i\le l-1}, \mathbf{h}^{\mathrm{enc}}), \tag{3}$$

$$\hat{\mathbf{y}}_l = \mathrm{softmax}(\mathbf{h}_l^{\mathrm{dec}}), \tag{4}$$

where $\mathrm{softmax}(\cdot)$ is the softmax activation function, and PE_i is the i-th absolute positional encoding [26]. The decoder $\mathrm{Dec}(\cdot)$ is composed of Transformer blocks, and the token embedding module $\mathrm{Emb}(\cdot)$ is a learnable linear layer. The whole summary sentence $\hat{y}$ is estimated by a beam search technique. At the beginning of the inference, the decoder receives the special symbol <sos>, which indicates the start of the sentence, and it outputs tokens until the special end token <eos> is estimated. During training, the E2E SSum model is optimized by the cross-entropy loss between $\hat{\mathbf{y}}_l$ and the one-hot vector of y_l. To stabilize model training, the ground truth token y_i is used in Eq. (3) instead of $\hat{y}_i$.

In addition to the aforementioned architecture, we apply the following two modifications to the E2E SSum model. First, we replace batch normalization (BN) in the Conformer blocks with layer normalization (LN). Since the input lengths of the SSum system are very long, the batch size is too small to learn well-generalized statistics of the BN layers during training. Second, we propose to replace the absolute positional encoding in Eq. (3) with the learnable positional embedding (LPE) [27] as

$$\mathbf{h}_l^{\mathrm{dec}} = \mathrm{Dec}([\mathrm{Emb}(\hat{y}_i) + \mathrm{LPE}(i)]_{0\le i\le l-1}, \mathbf{h}^{\mathrm{enc}}), \tag{5}$$

where a learnable linear layer $\mathrm{LPE}(i)$ embeds the position of the i-th token. Since the LPE has been successfully used in the natural language processing field [27, 28], we expect LPE could also improve performance for the SSum task.

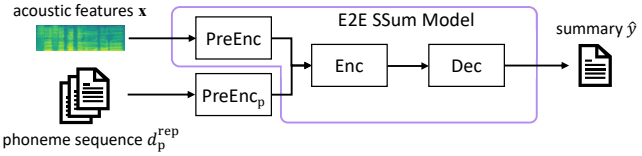

Fig. 2. E2E SSum model with phoneme pre-encoder $\mathrm{PreEnc_p}$.

4. PROPOSED METHODS

4.1. Multi-stage training

Our E2E SSum model receives three training datasets: paired data of speech and transcription $(\mathbf{x}, y_{\mathrm{asr}}) \in \mathcal{R}_{\mathrm{ASR}}$, paired data of speech and summary $(\mathbf{x}, y) \in \mathcal{R}_{\mathrm{Sum}}$, and external text summarization data $(d, y) \in \mathcal{R}_{\mathrm{Ext}}$. Here, d is a TSum input text. To leverage a variety of knowledge, we adopt a training strategy consisting of multiple stages as follows:

(i) An AED model is trained as an ASR model using the pairs in $\mathcal{R}_{\mathrm{ASR}}$ to acquire the transcribing capability internally.

(ii) To build the E2E SSum model, the aforementioned model is additionally fine-tuned with $\mathcal{R}_{\mathrm{Sum}}$.

(iii) The model is further fine-tuned with the combined dataset of $\mathcal{R}_{\mathrm{Sum}}$ and artificial data derived from $\mathcal{R}_{\mathrm{Ext}}$.

We define the model after stage (ii) as a baseline model and after stage (iii) as an augmented model by the TTS- or phoneme-based method.

4.2. TTS- and phoneme-based data augmentation

To utilize the TSum data for E2E SS, we propose two approaches: TTS- and phoneme-based data augmentation. For the TTS-based approach, we synthesize the raw waveforms from the input texts d of the TSum dataset and extract acoustic features for the stage (iii).

Although this TTS-based method is straightforward to augment speech summarization data, it requires high computational costs to synthesize speech. Therefore, we also propose an alternative approach that feeds phoneme sequences into the E2E SSum model instead. This approach is motivated by the performance improvement in [16], in which a multi-modal (i.e., acoustic and textual) data augmentation technique for E2E ASR was proposed. As a procedure, a word sequence d is first automatically converted into a phoneme sequence d_p by utilizing a grapheme-to-phoneme model. The phoneme duration is then modeled by repeating each phoneme in d_p in accordance with separately calculated statistics. The repeated phoneme sequence is denoted by d_p^{rep}. To input the d_p^{rep} together with $\mathbf{x}$, we also add and optimize the phoneme pre-encoder

Table 1. Results of previous work (P-1) [11], our baseline (C-1, B-1, B-2), and proposed models (T-1, 2).

	ID	Model	augmented data	#params.	ROUGE-1, 2, L (↑)	METEOR (↑)	BERTScore (↑)
baseline	C-1	cascade	-	201M+140M	60.9, 43.3, 55.4	30.3	92.6
	P-1	[11]		104M	60.9, 43.0, 55.9	28.8	91.0
	B-1	base		98M	65.3, 51.4, 62.1	32.5	93.0
	B-2	large		203M	65.6, 50.9, 62.0	32.8	93.2
proposed	T-1	large	TTS	203M	**68.4**, **54.1**, **65.0**	**34.9**	**93.8**
	T-2	large	phoneme	203M(+7M)[1]	67.4, 53.2, 64.1	33.9	93.6

$\mathrm{PreEnc_p}(\cdot)$ in addition to PreEnc simultaneously as

$$\mathbf{h}^{\mathrm{pre}} = \begin{cases} \mathrm{PreEnc}(\mathbf{x}) \\ \mathrm{PreEnc_p}(d_p^{\mathrm{rep}}). \end{cases} \quad (6)$$

Both inputs $\mathbf{x}$ and d_p^{rep} share other modules as depicted in Fig. 2. Here, we choose bidirectional LSTMs followed by a LN layer and subsampling CNNs for $\mathrm{PreEnc_p}$.

5. EXPERIMENTS

5.1. Datasets

We used the How2 dataset, which has approximately 2,000 hours of speech and is suitable for the E2E SSum task because of its relatively short speech inputs and abstractive summaries with a high compression rate. The training, validation, and evaluation sets contain 68,336, 1,600, and 2,127 pairs, and the average durations of their spoken documents are 84.7, 76.0, and 98.7 seconds, respectively. The input speech is truncated up to 100 seconds and composed of 40-dimensional log Mel-filterbank outputs with 3-dimensional pitch features. According to [21], the summaries are automatically collected from "description" tags on YouTube.

For the data augmentation, we utilized the Gigaword corpus [19, 20] composed of 3.8M pairs of the first sentences and titles of news articles. The average word counts on this dataset for the input text and summaries are 31.4 and 8.3, respectively. Since the average input length is much smaller than that of other text summarization corpora, such as CNN Daily Mail [29], the synthesizing cost for the Gigaword corpus is relatively lower. Even though the domain of this external corpus mismatches that of the How2 dataset, we expect that the model could learn additional words and expressions.

5.2. Training and evaluation

We adopted a 12-layer Conformer encoder and a 6-layer Transformer decoder for the model architecture. The pre-encoder was composed of 4-layer CNNs with a subsampling rate of 4. We evaluated two model sizes: the base and large models with 512- and 768-dimensional embeddings, respectively. Their encoders commonly had 2048 feed-forward (FF) units, 8 attention heads, and a kernel size of 31. The decoder of the base model had 2048 FF units and 4 attention heads, and that of the large model had 3072 FF units and 12 attention heads.

In the training stage (i) described in Section 4.1, we trained the E2E ASR model with the Adam optimizer [30] and Noam scheduler [26] with a learning rate of $2 \cdot 10^{-3}$, warmup steps of 40k, a weight decay rate of 10^{-5}, and a batch size of 512. The word error rates (↓) of the base and large models on the evaluation set were 9.8% and 9.5% with a beam size of 16, respectively.

For the training stage (ii), the model was fine-tuned as the E2E SSum model with a learning rate of 10^{-4} with a reduction factor of 0.5, and batch size of 30.

[1] The phoneme encoder (7M) is not needed during inference.

The model was further fine-tuned with augmented data in the training stage (iii). For the TTS model, we used VITS [31] trained with the LJSpeech corpus [2]. The synthesized speech does not have a diversity of speakers because the LJSpeech contains only a single speaker. This enables us to evaluate only the effectiveness of additional linguistic information excluding the effect of augmenting speaker varieties. After generating the raw waveform, we extracted 43-dimensional acoustic features with a window size of 25ms and a shift of 10ms from the waveform and applied cepstral mean-variance normalization (CMVN) to match the original features in the How2 corpus. When training the model in the stage (iii), we adopted a learning rate of 10^{-4} with a reduction factor of 0.5, and the learning rate of its decoder was 10^{-3}. The maximum total length of input sequences in one batch was set to 300,000. Each batch contains only real or artificial samples.

For the phoneme-based method, we used the g2pE toolkit[3] to convert word sequences into phoneme sequences. The number of phoneme units was 42. The duration of phoneme p was determined following the normal distribution $\mathcal{N}(\mu_p, \sigma_p)$ at each time it appeared. Here, the mean μ_p and standard deviation σ_p of p were estimated with the TIMIT corpus [32]. The phoneme pre-encoder had 3-layer bidirectional LSTMs followed by LN and 4-layer CNNs with a subsampling rate of 4. The learning rate of the phoneme pre-encoder was 10^{-3}. The other configurations were the same as those of the TTS-based method. In the stage (i), (ii), and (iii), we used byte-pair encoding and set the vocabulary size to 1,000.

When evaluating model performance, we chose the checkpoints with the best validation accuracy and decoded them with a beam width of 4 for all conditions. For evaluation metrics, we selected the ROUGE [33], METEOR [34], and BERTScore [35] scores commonly used in TSum tasks.

We also evaluated the conventional cascade model. The ASR model was the same as the large model obtained after the training stage (i). The TSum model was a BART-based model fine-tuned with the CNN/Daily Mail corpus[4] [29] and the How2 dataset. During the fine-tuning with the How2 dataset, we trained the model for 1M steps with the Adam optimizer, a learning rate of 5×10^{-5} with linear decay, and a batch size of 8. We used ESPnet2[5] for all of the implementation, training, and evaluation.

5.3. Results on entire How2 evaluation set

Table 1 shows the evaluation scores of the augmented model (T-1, T-2) with prior state-of-the-art (P-1) [11], the cascade model (C-1), and our baselines (B-1, B-2) on the How2 evaluation set. Compared with Model P-1, Model B-1 with full attention, LN, and LPE remarkably improved the scores and achieved the current SoTA performance.

[2] https://github.com/espnet/espnet/tree/master/egs2/ljspeech/tts1#pretrained-models-1

[3] https://github.com/Kyubyong/g2p

[4] https://huggingface.co/ainize/bart-base-cnn

[5] https://github.com/espnet/espnet

Table 2. Part of summaries of ID: zpoDukAvPPY in the evaluation set generated by Model B-2, T-1, and T-2 with the ground truth (g.t.). The How2 training set does not contain the bolded word ***gladiator***, while the Gigaword training set contains it.

ID	Summary
B-2	*learn to draw calf shoes with tips from a fashion expert in this free fashion design video.*
T-1	*learn to draw **gladiator** sandals in this free fashion illustration video from a fashion design graduate student.*
T-2	*learn to draw **gladiator** sandals with tips from a fashion expert in this free fashion design video.*
g.t.	*learn to draw **gladiator** sandals in this free fashion video from a fashion design graduate student.*

Table 3. Number of samples contained in each evaluation set filtered with various threshold α.

α	0.5	0.6	0.7	0.8	0.9	1.0 (full)
# samples	110	441	921	1336	1666	2127

There was a slight improvement by enlarging the model size as in Model B-2, and this strong model is the baseline for the proposed data augmentation methods. The result of Model T-1 shows that the TTS-based data augmentation method further improved the METEOR, ROUGE-L, and BERTScore scores by as much as 2.1, 3.0, and 0.6 points, respectively. In addition, despite its low cost, Model T-2 trained with the phoneme-based method was also improved by 1.1, 2.1, and 0.4 points in terms of the METEOR, ROUGE-L, and BERTScore scores, respectively. Although the domain of the augmented data is completely different from that of the How2 dataset, it helped the model expand its vocabulary. For instance, as seen in Table 2, Model B-2 could not output a word *gladiator*, which does not appear in the How2 training set, while Model T-1 and T-2 seemed to have learned it from the Gigaword training set.

5.4. Analysis and results on filtered How2 evaluation sets

While we showed the effectiveness of our system in Section 5.3, we observed several evaluation examples with extraordinarily high scores only for the E2E SSum models compared with the cascade model. We analyzed these samples and found that the training dataset contained samples with similar or perfectly matched summaries to the evaluation set. For example, the summary of ID: 6D0hfwYIwm4 from the training set:

> *Indian food is full of interesting spices and flavors. Learn how to **serve** Indian potatoes, carrots & peas ...*

is very similar to that of ID: 2NdV94T-JNc from the evaluation set:

> *Indian food is full of interesting spices and flavors. Learn how to **season** Indian potatoes, carrots & peas ...*

The audio data corresponding to such summaries are often different parts of a single video (e.g., the first and second parts of a video are in the training and evaluation sets, respectively.) This is possibly because the How2 dataset defines the "description" tags of YouTube as the summary, as mentioned in Section 5.1, and the training and evaluation sets are partitioned randomly. Under this condition, it is difficult to evaluate the generalization capabilities of the models. For example, a model overfitting the training data could achieve high scores on the test samples that overlap with the training set.

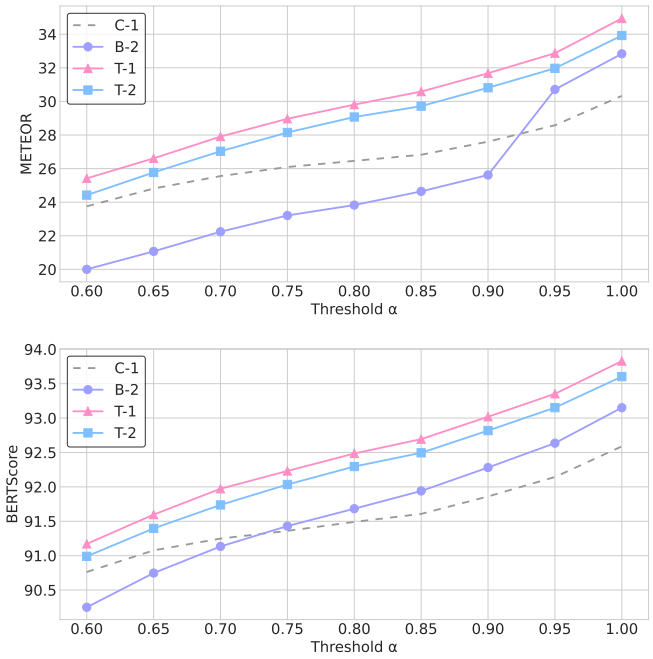

Fig. 3. The METEOR (above) and BERTScore (below) scores of Models B-2, T-1, T-2, and C-1 changing the filtering threshold α from 0.6 to 1.0.

Therefore, we propose to filter the evaluation set to omit such samples by utilizing a quantitative criterion. First, we computed the ROUGE-L scores between each summary in the evaluation set and the whole How2 dataset. We defined the highest score as the *leakage score* for each summary in the evaluation set. For example, the leakage score of ID: 2NdV94T-JNc is 0.95. Then, we removed the samples whose leakage score was larger than a threshold α from the evaluation set. Table 3 shows the number of samples contained in each filtered evaluation set for different threshold values.

Figure 3 shows the METEOR and BERTScore scores of Models B-2, T-1, T-2, and C-1 with a different threshold α. While the scores for all models tend to be worse as α decreases, Model B-2 shows the clearest trend. This result suggests that the baseline E2E system overfits the training data and is vulnerable to unfamiliar samples. Model C-1 shows robustness for unseen samples and the slowest score reductions, probably owing to the abundant linguistic knowledge of its TSum model. For our proposed method, Models T-1 and T-2 achieved the highest accuracy in all conditions and seemed to have gained linguistic information from the external TSum corpus.

6. CONCLUSION AND FUTURE WORK

In this paper, we studied the effectiveness of data augmentation methods for E2E SSum with external text summarization data on the How2 dataset. Compared with a strong baseline, the TTS- and phoneme-based data augmentation methods improved the METEOR score by 2.1 and 1.1 points, respectively. In addition, we also showed a clear trend of acquiring linguistic knowledge by utilizing TSum data in our E2E SSum model.

In this paper, we leverage only text summarization data as input for the E2E SSum models. As part of future research, we will attempt to leverage large external language models to feed the models with knowledge from diverse unpaired text data.

7. REFERENCES

[1] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Yalta, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on Transformer vs RNN in speech applications," in *Proc. of ASRU*, 2019.

[2] J. Goldman, S. Renals, S. Bird, F. De Jong, M. Federico, C. Fleischhauer, M. Kornbluh, L. Lamel, D. W. Oard, C. Stewart, and R. Wright, "Accessing the spoken word," *Int. J. Digit. Libr.*, 2005.

[3] G. Murray, G. Carenini, and R. Ng, "Generating and validating abstracts of meeting conversations: a user study," in *Proc. of INLG*, 2010.

[4] R. Sharma, M. Mahrishi, S. Morwal, and G. Sharma, "Index point detection for text summarization using cosine similarity in educational videos," *Materials Science Horiand Engineering*, 2021.

[5] G. Finley, E. Edwards, A. Robinson, M. Brenndoerfer, N. Sadoughi, J. Fone, N. Axtmann, M. Miller, and D. Suendermann-Oeft, "An automated medical scribe for documenting clinical encounters," in *Proc. of NAACL*, 2018.

[6] Y. Zhang, A. Ni, T. Yu, R. Zhang, C. Zhu, B. Deb, A. Celikyilmaz, A. H. Awadallah, and D. Radev, "An exploratory study on long dialogue summarization: What works and what's next," in *Proc. of EMNLP*, 2021.

[7] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, and D. R. Radev, "QMSum: A new benchmark for query-based multi-domain meeting summarization," in *Proc. of NAACL-HLT*, 2021.

[8] M. Á. Tündik, V. Kaszás, and G. Szaszák, "Assessing the semantic space bias caused by ASR error propagation and its effect on spoken document summarization," in *Proc. of INTERSPEECH*, 2019.

[9] T. Kano, A. Ogawa, M. Delcroix, and S. Watanabe, "Attention-based multi-hypothesis fusion for speech summarization," in *Proc. of ASRU*, 2021.

[10] T.-E. Liu, S.-H. Liu, and B. Chen, "A hierarchical neural summarization framework for spoken documents," in *Proc. of ICASSP*, 2019.

[11] R. Sharma, S. Palaskar, A. W. Black, and F. Metze, "End-to-end speech summarization using restricted self-attention," in *Proc. of ICASSP*, 2022.

[12] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, "Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition," in *Proc. of SLT*, 2018.

[13] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Data augmentation for ASR using TTS via a discrete representation," in *Proc. of ASRU*, 2021.

[14] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end ASR," in *Proc. of SLT*, 2018.

[15] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Synthesizing waveform sequence-to-sequence to augment training data for sequence-to-sequence speech recognition," *Acoustical Science and Technology*, 2021.

[16] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end ASR," *Proc. of INTERSPEECH*, 2018.

[17] R. Masumura, N. Makishima, M. Ihori, A. Takashima, T. Tanaka, and S. Orihashi, "Phoneme-to-grapheme conversion based large-scale pre-training for end-to-end automatic speech recognition," 2020.

[18] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *Proc. of ICASSP*, 2019.

[19] D. Graff, J. Kong, K. Chen, and K. Maeda, "English Gigaword," *Linguistic Data Consortium, Philadelphia*, 2003.

[20] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," 2015.

[21] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: A large-scale dataset for multimodal language understanding," in *Proc. of NeurIPS Workshop on ViGIL*, 2018.

[22] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. of ICASSP*, 2018.

[23] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in *Proc. of ICASSP*, 2019.

[24] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *Proc. of ICASSP*, 2020.

[25] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. of INTERSPEECH*, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of NIPS*.

[27] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. of ICML*, 2017.

[28] 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 *Proc. of ACL*, 2020.

[29] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. of NIPS*, 2015.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[31] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. of ICML*, 2021.

[32] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. of the DARPA CSR*, 1989.

[33] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004.

[34] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. of EACL WMT*, 2014.

[35] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. of ICLR*, 2020.