INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

GhostT5: Generate More Features with Cheap Operations to Improve Textless Spoken Question Answering

Xuxin Cheng[†]*, Zhihong Zhu*[†]*, Ziyu Yao, Hongxiang Li, Yaowei Li, Yuexian Zou*[*]

ADSPLAB, School of ECE, Peking University, China

{chengxx, zhihongzhu, yaozy, lihongxiang, ywl}@stu.pku.edu.cn,
zouyx@pku.edu.cn

Abstract

Spoken question answering (SQA) aims to identify the correct answer to the given the question from a spoken passage. Most conventional SQA frameworks combine an automatic speech recognition (ASR) module and a text question answering (TQA) module in a cascaded manner, which might suffer from *error propagation* and *high latency*. To tackle these issues, several end-to-end SQA frameworks based on *Textless NLP* are proposed. However, existing end-to-end models still fail to outperform the cascade models with the similar number of parameters. In this paper, to improve textless SQA, we propose GhostT5, which generates more features from the remaining features with very cheap operations for stronger performance. Experiment results and further analysis show that our GhostT5 achieves the new state-of-the-art performance on NMSQA dataset and surpasses cascaded SQA models. More encouragingly, GhostT5 surpasses the previous best end-to-end SQA model with less than half of the parameters.

Index Terms: spoken question answering, Textless NLP, end-to-end frameworks, cheap operations

1. Introduction

As a crucial task for personal assistants, spoken question answering (SQA) has attracted more and more attention from researchers. SQA requires the machine to retrieve and summarize the spoken passage from a fixed collection, and then analyze the retrieved information to provide the answer based on the given question, where the answer is a span from the spoken passage. In contrast to other relatively simple spoken language understanding tasks [1, 2, 3, 4], SQA is recognized as a more challenging task which involves understanding the meaning of each spoken phrase and requires advanced comprehension and reasoning abilities to process longer audio content.

Most of the conventional SQA frameworks are cascaded by connecting separately an automatic speech recognition (ASR) module and a text question answering (TQA) module. The ASR module transcribes speech sequences into corresponding texts, and the TQA module predicts a concrete answer using natural language processing (NLP) techniques. However, the cascade SQA frameworks might suffer from error propagation and high latency. The inevitable errors (e.g., *pull* to *pool*) from the ASR module may cause catastrophic problems to the TQA module. As other tasks [5], label noise is also common in SQA. Meanwhile, correct answers to the questions may include name entities or out-of-vocabulary (OOV) words that can never be recognized. As a result, the key information will be lost when the speech sequences are transformed into transcripts with errors. Furthermore, the ASR module is trained to achieve a lower word error rate (WER), which treats all words equally without distinction [6]. However, different words have different importance for SQA task. Some words appear frequently in the correct answers, while some words are absolutely irrelevant to the SQA task. Minimizing WER on words which are irrelevant to the SQA task is meaningless and may even affect the performance of the SQA model. Therefore, it may be sub-optimal to individually train ASR and TQA models in a cascaded manner.

To address these issues, several text-based methods [7, 8, 9] and fusion-based methods [10, 11, 12] are applied in several works. [7, 13] utilize a sub-word unit strategy to mitigate the negative impacts of ASR errors. Knowledge distillation is adopted by [14, 15] to improve ASR robustness of the TQA model. [16] designs a two-stage training framework including a self-supervised training stage and a contrastive representation learning stage to capture semantic similarity while modelling the interactions between speech and text data. [17] introduces a teacher-student framework to transfer knowledge from a weighted teacher to improve the robustness of ASR module of the student model. [9] constructs a cross-modal speech and text pre-trained BERT-based language model with aligned semantics for SQA task to jointly learn audio-text features to alleviate the ASR errors. [6] proposes the first textless end-to-end SQA framework DUAL based on Longformer [18], which utilizes no ASR transcripts and thus not suffering from ASR errors. Owing to bypassing the ASR stage, it is more robust to real-world data. Recently, [19] proposes ByT5lephone based on ByT5 [20], which has more than twice the parameters than DUAL. ByT5lephone justifies the utilization of byte-level models over subword-level models in the SQA task and outperforms other end-to-end SQA methods.

Despite there have been some models which leverage *Textless NLP* technique to avoid suffering from ASR errors [6, 19], they still fail to outperform the cascade models with the similar number of parameters. We believe a critical reason is that there are not enough features in these models, resulting in inadequate modeling. To tackle this issue, we propose a novel end-to-end SQA framework termed GhostT5. Specifically, we introduce the ghost module, whose effectiveness is demonstrated in many computer vision (CV) tasks [21] and NLP tasks [22]. We utilize ByT5lephone [19] for initialization and prune away unimportant attention heads with the width multiplier m. Then, we use the efficient 1-Dimensional Depthwise Separable Convolution (DWConv) [23] as the fundamental operation in the ghost module. Meanwhile, softmax function is applied to normalize the convolution kernel to ensure that the generated ghost features have comparable scales to the original ones. We proceed

[†] Equal contribution.
[*] Corresponding author.

10.21437/Interspeech.2023-41

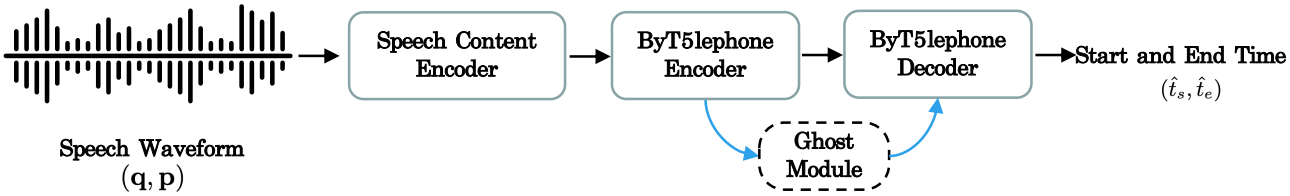

Figure 1: *The architecture of the GhostT5 framework, which consists of three components: (1) original speech waveform is fed into speech content encoder to obtain the discrete unit sequence; (2) ByT5lephone transforms the discrete unit sequence to start and end time; (3) Ghost Module generates more features with very cheap operations.*

with fine-tuning the parameters of both the ByT5lephone backbone model and the ghost modules.

GhostT5 leverages pre-trained models to extract quantized and condensed speech representations from speech sequences, and adapts a pre-trained language model to produce answers to the questions without relying on any ASR transcripts. Through directly locating the answer span from the speech sequence, the extracted answers are not affected by ASR errors or OOV issues which often occur in ASR frameworks. Experiment results show that our GhostT5 ($m = 1/3$) significantly outperforms previous models, and the unpruned GhostT5 ($m = 1$) can even perform better than the pruned version. Model analysis further verifies the advantages of our model.

The contributions of this work are four-fold:

- To the best of our knowledge, we make the first attempt to generate more features with cheap operations by applying the ghost module for the SQA task.
- Experiment results demonstrate that our model achieves new state-of-the-art performance on the NMSQA dataset and surpasses the cascaded SQA models with the similar number of parameters for the first time.
- Our model exceeds the previous best end-to-end SQA model with less than half of the parameters.
- In parallel, our proposed method is a unified framework, which can be easily applied to a variety of downstream speech processing tasks.

2. Method

2.1. Problem Formulation

The corpus of SQA usually contains *question-passage-answer* triples, which can be formulated as $\mathcal{D} = \{(\mathbf{q}, \mathbf{p}, \mathbf{a})\}$, where $\mathbf{q}$ is the audio sequence of a question, $\mathbf{p}$ is the audio sequence of a passage and $\mathbf{a}$ is the audio sequence of the answer to the question. An end-to-end SQA framework aims to directly generate the starting time t_s and the ending time t_e, denoted as the answer span $\mathbf{a}$, from the spoken passage $\mathbf{p}$ given the spoken question $\mathbf{q}$ without generating any ASR transcripts.

2.2. Model Architecture

As shown in Figure 1, our framework comprises of three major components, including the Speech Content Encoder (§2.2.1), ByT5lephone (§2.2.2) and the Ghost Module (§2.2.3).

2.2.1. Speech Content Encoder

Given a question-passage audio waveform $(\mathbf{q}, \mathbf{p})$ pair, unlike conventional cascade SQA frameworks, the Speech Content Encoder transforms the pair $(\mathbf{z_q}, \mathbf{z_p})$ into the sequences of discrete units rather than corresponding ASR transcripts.

Following [6], we adopt HuBERT-Large[1] [24] for feature extraction, which is trained by masking prediction objectives similar to BERT [25]. HuBERT-Large contains 24 transformer encoder layers and is pre-trained on the Libri-Light 60k hours dataset[2] [26], to encode the raw waveform into frame-level 1024 dimension features, with each frame equivalent to 20 ms. Then we apply the open-source S3PRL[3] [27] toolkit to extract the representations of the HuBERT-Large model.

To feed the extracted features into the following pre-trained language model, we perform a quantization operation on them. We adopt K-means clustering to layer-wise representations of HuBERT-Large and train the K-means clustering model on the 100-hours subset of LibriSpeech dataset[4] [28]. For the clustering process, we use 64, 128, and 512 clusters, respectively. The resulting discrete units are represented by the clustering indices. To reduce sequence length and remove duration information, we merge repetitive discrete units to form a dense discrete unit sequence for the question and passage, denoted as $(\mathbf{z_q}, \mathbf{z_p})$. We also record the repetition of every discrete units in $\mathbf{z_q}$ and $\mathbf{z_p}$, denoted as $\mathbf{c_q}$ and $\mathbf{c_p}$, respectively, which allows us to recover the frame-level indices and convert the answer span back to the time interval during the inference stage.

2.2.2. Pre-trained Language Model

Pre-trained language models (PLMs) are often introduced to improve the performance [29, 30, 31, 32, 33]. To improve the downstream SQA task, we leverage the transferability of PLMs from related fields. Since the previous work [19] verifies that byte-level models outperform subword-level models in SQA, we choose ByT5lephone [19] as the pre-trained language model, which is first initialized with ByT5 [20] and then fine-tuned with the phonemicized inputs from the wiki text corpus. Specifically, we initialize the network using the weights of ByT5lephone and randomly assign pre-trained embeddings to the discrete units. The input of ByT5lephone is the concatenated sequences of discrete units of the question and paragraph pairs $(\mathbf{z_q}, \mathbf{z_p})$. A randomly initialized linear layer is added on the top of ByT5lephone, and outputs the predicted start and end time (y_s, y_e) as the answer to the question.

The length of a discrete unit sequence is much longer than its corresponding text, and the duration of a spoken passage is also very long. Following [19], we slice the context into several segments to concatenate them with the question and force the model to find the correct segment and the correct span. After

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/hubert

[2] https://github.com/facebookresearch/libri-light

[3] https://github.com/s3prl/s3prl

[4] http://www.openslr.org/12

this operation, we are able to reduce the model length constraint to 1024 to accommodate ByT5lephone.

2.2.3. Ghost Module

Despite existing models have obtained the promising results via *Textless NLP* technique, they still fail to surpass cascade SQA models with the similar number of parameters. We believe the reason is that there are not enough features in ByT5lephone. To tackle this issue, we add ghost modules to the top of its encoder to generate more features with negligible additional parameters.

Meanwhile, since there are much more parameters than the previous model [6] in ByT5lephone, we also prune away unimportant attention heads of ByT5lephone with the width multiplier m. Previous works [34, 22] illustrates that the calculations involved in attention heads of multi-head attention (MHA) can operate concurrently. Consequently, by removing the parameters associated with the attention heads, ByT5lephone can undergo a systematic compression process.

We denote the sequence length as n and denote the hidden state size as d, respectively. For the input matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the output of the h-th attention head is as follows:

$$\mathbf{H}_h\left(\mathbf{X}\right) = \mathrm{Softmax}\left(\frac{\mathbf{X}\mathbf{W}_h^Q\mathbf{W}_h^{K\top}\mathbf{X}^\top}{\sqrt{d}}\right)\mathbf{X}\mathbf{W}_h^V\mathbf{W}_h^{O\top} \quad (1)$$

where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V, \mathbf{W}_h^O \in \mathbb{R}^{d\times d_h}$ are parameter matrices, and d_h is computed as follows:

$$d_h = d/N_H \quad (2)$$

In the original MHA, the final output of N_H heads are computed in parallel as follows:

$$O_{\mathrm{ORIG}}(\mathbf{X}) = \sum_{h=1}^{N_H}\mathbf{H}_h(\mathbf{X}) \quad (3)$$

Convolution operation is capable of encoding local context dependency, which can serve as an effective complement to global self-attention mechanisms [35, 36]. In addition, many features, such as vertical and diagonal attention maps, can be easily generated by applying convolution operation on similar other features [37]. Moreover, 1-Dimensional convolution (Conv1D) operation applied to the sequence direction can capture the local dependency and has been shown to be highly effective for many NLP tasks [23, 36]. In order to take advantage of the representational power of Conv1D while minimizing additional memory and computation requirements, we finally choose DWConv as the ghost module. DWConv applies convolution independently over each channel, reducing the number of parameters from d^2k to dk compared to Conv1D, where k denotes the convolution kernel size. We input the output $\mathbf{H}_h(\mathbf{X})$ of the h^{th} head in MHA to DWConv. By utilizing DWConv, the output O_{D} for the i^{th} token and the c^{th} channel is as follows:

$$O_{\mathrm{D}}(\mathbf{H}_h(\mathbf{X})) = \sum_{m=1}^{k}\mathbf{W}_{c,m}\cdot\mathbf{H}_h(\mathbf{X})_{i-\lceil\frac{k+1}{2}\rceil+m,c} \quad (4)$$

Following [23], we also apply the softmax function to normalize the convolution kernel to ensure that the output has a similar scale as the input. The normalized output $\hat{O}_{\mathrm{D}}$ is:

$$\hat{O}_{\mathrm{D}}(\mathbf{H}_h(\mathbf{X})) = \sum_{m=1}^{k}\mathrm{Softmax}(\mathbf{W}_{c,m})\cdot\mathbf{H}_h(\mathbf{X})_{i-\lceil\frac{k+1}{2}\rceil+m,c} \quad (5)$$

Given a width multiplier $m \le 1$, we keep $M = \lfloor N_h m\rfloor$ heads and utilize them to generate F ghost features. The f^{th} ghost feature $\mathbf{G}_f(\mathbf{X})$ is generated by:

$$\mathbf{G}_f(\mathbf{X}) = \mathrm{ReLU}\left(\sum_{h=1}^{M}\hat{O}_{\mathrm{D}}(\mathbf{H}_h(\mathbf{X}))\right) \quad (6)$$

where ReLU [38] is applied as the nonlinearity function. Thus the final output of MHA in GhostT5 is as follows:

$$O_{\mathrm{Ghost}}(\mathbf{X}) = \sum_{h=1}^{M}\mathbf{H}_h(\mathbf{X}) + \sum_{f=1}^{F}\mathbf{G}_f(\mathbf{X}) \quad (7)$$

To further improve the performance, we adopt knowledge distillation method. We denote the layers of encoder as S. The final distillation loss $\mathcal{L}_{dis}$ is the sum of the embedding distillation loss $\mathcal{L}_{emb}$ and the hidden states distillation loss $\mathcal{L}_{mha}$:

$$\mathcal{L}_{emb} = \mathrm{MSE}(\mathbf{E}^G, \mathbf{E}) \quad (8)$$

$$\mathcal{L}_{mha} = \sum_{s=1}^{S}\mathrm{MSE}(\mathbf{M}_s^G, \mathbf{M}_s) \quad (9)$$

$$\mathcal{L}_{dis} = \mathcal{L}_{emb} + \mathcal{L}_{mha} \quad (10)$$

where MSE denotes the mean squared error, E^G and E denote the output of the embedding layer from the student model GhostT5 and the full-sized teacher model ByT5lephone, and M_s^G and M_s denote the hidden states after MHA from GhostT5 and ByT5lephone, respectively.

2.3. Training Objective

The overall training objective for SQA is as follows:

$$\mathcal{L} = \mathcal{L}_{dis} - \sum log\, P(y_s|\mathbf{z_q}, \mathbf{z_p}) + log\, P(y_e|\mathbf{z_q}, \mathbf{z_p}) \quad (11)$$

where $(\mathbf{z_q}, \mathbf{z_p})$ denotes the discrete unit sequence for the question and passage, (t_s, t_e) denotes the ground truth start and end time, and (y_s, y_e) denotes the converted version at index level.

During the inference stage, we utilize the repetition of every discrete unit $\mathbf{c_p}$ to convert the predicted start and end indices $(\hat{y}_s, \hat{y}_e)$ to frame level, and then transform them to the time level $(\hat{t}_s, \hat{t}_e)$ as the final prediction.

3. Experiments

3.1. Datasets and Metrics

We conduct all the experiments on NMSQA dataset[5][6], which is a publicly available SQA dataset derived from SQuAD-v1.1 dataset[6][39]. For the `train` set and the `dev` set, the speech is generated with Amazon Polly Text-to-Speech service[7]. And for the `test` set, the speech is read by 60 human speakers, including 30 males and 30 females. NMSQA contains 297.18 hours, 37.61 hours, and 2.67 hours of audio for the `train`, `dev`, and `test` set, respectively. Following the previous work [6], we evaluate the performance of SQA using the Frame-level F1 score (FF1) and Audio Overlapping Score (AOS) based on the predicted time intervals. Higher FF1 and AOS mean more overlap between the predicted and ground truth spans.

[5] https://github.com/DanielLin94144/DUAL-textless-SQA

[6] https://rajpurkar.github.io/SQuAD-explorer

[7] https://aws.amazon.com/polly

3.2. Implementation Details

We utilize ByT5lephone[8][19] as the PLM. The learning rate is set to 3e-5 and warm-up updates are set to 1k. We train the model up to 15 epochs to avoid overfitting, and the batch size is set to 128. We use Adam optimizer [40] to optimize the parameters in our model with $\beta_1 = 0.9, \beta_2 = 0.98$. We set the width multiplier m to 1/3. For all the experiments, we select the model which performs the best on `dev` set in FF1 and evaluate it on `test` set. All experiments are conducted on 4 Nvidia Tesla-V100 GPUs. The training process lasts several hours.

3.3. Main Results

As shown in Table 1, we compare our model with four SQA baselines, including SB [41], W2v2 [42], DUAL [6], ByT5lephone-small [19]. Our model achieves an improvement of 7.5 FF1 and 8.0 AOS over previous best end-to-end model with less than half of the parameters. Moreover, we surpasses the cascaded SQA models for the first time, which further verifies the effectiveness of *Textless NLP* in SQA task.

Table 1: *FF1 and AOS scores on the NMSQA `test` set. "#Params" indicates the number of parameters of the model. Best results are highlighted in* ***bold****.*

Model	#Params	FF1	AOS
Cascade Models (w/ ASR transcripts)			
SB [41]	148M	17.3	15.3
W2v2 [42]	148M	64.2	57.4
End-to-End Models (w/o ASR transcripts)			
DUAL [6]	148M	55.9	44.1
ByT5lephone-small [19]	299M	61.1	53.3
GhostT5 (ours)	121M	**68.6**	**61.3**

3.4. Model Analysis

3.4.1. Ablation Study

To verify the effectiveness of each component in our method, we perform several ablation experiments on NMSQA dataset, whose results are shown in Table 2. We can clearly observe that FF1 drops by 4.7, 4.1 and 1.2 and AOS drops by 5.1, 3.4, and 2.8 when knowledge distillation, Softmax and ReLU are removed, respectively. The drop caused by knowledge distillation is the largest of the three components, which indicates that knowledge distillation can significantly transfer knowledge from the full-sized teacher model to the pruned model to compensate for the pruned features and improve the performance.

3.4.2. Effect of the Width Multiplier

In our model, the width multiplier is set to 1/3. Table 3 demonstrates the experimental results with different width multiplier. We can find that as the width multiplier grows, FF1, AOS and the number of parameters also increase. In particular, the model performs best when the width multiplier increases to 1, which indicates that the ghost module can be applied directly to the ByT5lephone for better performance, while the additional parameters can be ignored.

3.4.3. Performance for Poor ASR Accuracy

We also conduct an experiment to compare the performance of the cascade W2v2 model and GhostT5 model at different levels

[8] `https://github.com/Splend1d/T5lephone`

Table 2: *Ablation study of knowledge distillation (KD), Softmax, and ReLU. Results are reported on NMSQA dataset.*

Models	FF1	AOS
GhostT5	**68.6**	**61.3**
w/o KD	63.9(↓ 4.7)	56.2(↓ 5.1)
w/o Softmax	64.5(↓ 4.1)	57.9(↓ 3.4)
w/o ReLU	67.4(↓ 1.2)	58.5(↓ 2.8)

Table 3: *Experimental results with different width multiplier.*

Multiplier	#Params	FF1	AOS
1/3	**121M**	**68.6**	**61.3**
1/2	154M(↑33M)	68.9(↑0.3)	61.7(↑0.4)
2/3	241M(↑120M)	69.3(↑0.7)	62.1(↑0.8)
1	303M(↑182M)	69.6(↑1.0)	62.4(↑1.1)

of WER. Specifically, we divide the NMSQA `dev` set into subsets based on the WER obtained with W2v2 ranging from 0% to 70%. As shown in Figure 2, FF1 score for W2v2 decreases significantly and continuously as the WER increases. In contrast, GhostT5 achieves similar FF1 scores for different levels of WER, even when the WER reaches 70%. The reason is that GhostT5 does not rely on ASR transcripts. Moreover, GhostT5 performs better than W2v2 when WER exceeds 5%, which further verifies the superiority of our model.

Figure 2: *FF1 score for GhostT5 and cascade W2v2, evaluated on small groups of NMSQA `dev` set at different levels of WER.*

4. Conclusions

In this paper, we propose a novel textless SQA framework termd GhostT5. Ghost module is introduced to generate more features with cheap operations. Experiments and analysis demonstrate the effectiveness of our proposed method, which can improve the performance with fewer parameters. In the future, we will explore how to further leverage *Textless NLP* technique to improve the SQA framework.

5. Acknowledgements

We thank all the anonymous reviewers and program chairs for their valuable and insightful feedback. This paper was partially supported by Shenzhen Science & Technology Research Program (No:GXWD20201231165807007-20200814115301001) and NSFC (No: 62176008) .

6. References

[1] X. Cheng, Q. Dong, F. Yue, T. Ko, M. Wang, and Y. Zou, "M 3 st: Mix at three levels for speech translation," in *Proc. of ICASSP*, 2023.

[2] Z. Zhu, X. Cheng, Z. Huang, D. Chen, and Y. Zou, "Towards unified spoken language understanding decoding via label-aware compact linguistics representations," in *Proc. of ACL Findings*, 2023.

[3] X. Cheng, B. Cao, Q. Ye, Z. Zhu, H. Li, and Y. Zou, "Ml-lmcl: Mutual learning and large-margin contrastive learning for improving asr robustness in spoken language understanding," in *Proc. of ACL Findings*, 2023.

[4] Z. Zhu, W. Xu, X. Cheng, T. Song, and Y. Zou, "A dynamic graph interactive framework with label-semantic injection for spoken language understanding," in *Proc. of ICASSP*, 2023.

[5] X. Cheng, Z. Zhu, H. Li, Y. Li, and Y. Zou, "Ssvmr: Saliency-based self-training for video-music retrieval," in *Proc. of ICASSP*, 2023.

[6] G.-T. Lin, Y.-S. Chuang, H.-L. Chung, S. wen Yang, H.-J. Chen, S. A. Dong, S.-W. Li, A. Mohamed, H. yi Lee, and L. shan Lee, "DUAL: Discrete Spoken Unit Adaptive Learning for Textless Spoken Question Answering," in *Proc. of Interspeech*, 2022.

[7] C. Li, S. Wu, C. Liu, and H. Lee, "Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension," in *Proc. of INTERSPEECH*, 2018.

[8] C. Lee, Y. Chen, and H. Lee, "Mitigating the impact of speech recognition errors on spoken question answering by adversarial domain adaptation," in *Proc. of ICASSP*, 2019.

[9] Y. Chuang, C. Liu, H. Lee, and L. Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," in *Proc. of INTERSPEECH*, 2020.

[10] N. Chen, C. You, and Y. Zou, "Self-supervised dialogue learning for spoken conversational question answering," in *Proc. of INTERSPEECH*, 2021.

[11] C. You, N. Chen, and Y. Zou, "Knowledge distillation for improved accuracy in spoken question answering," in *Proc. of ICASSP*, 2021.

[12] Y. Chenyu, C. Nuo, and Z. Yuexian, "Contextualized attention-based knowledge transfer for spoken conversational question answering," in *Proc. of INTERSPEECH*, 2021.

[13] C.-H. Lee, S.-M. Wang, H.-C. Chang, and H.-Y. Lee, "Odsqa: Open-domain spoken question answering dataset," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018.

[14] D. Su and P. Fung, "Improving spoken question answering using contextualized word representation," in *Proc. of ICASSP*, 2020.

[15] C. You, N. Chen, and Y. Zou, "Mrd-net: Multi-modal residual knowledge distillation for spoken question answering." in *Proc. of IJCAI*, 2021.

[16] Y. Chenyu, C. Nuo, and Z. Yuexian, "Self-supervised contrastive cross-modality representation learning for spoken question answering," in *Proc. of EMNLP Findings*, 2021.

[17] C. You, N. Chen, F. Liu, D. Yang, and Y. Zou, "Towards data distillation for end-to-end spoken conversational question answering," *ArXiv preprint*, 2020.

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

[19] C.-J. Hsu, H.-L. Chung, H.-y. Lee, and Y. Tsao, "T5lephone: Bridging speech and text self-supervised models for spoken language understanding via phoneme level t5," *ArXiv preprint*, 2022.

[20] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "ByT5: Towards a token-free future with pre-trained byte-to-byte models," *Transactions of the Association for Computational Linguistics*, 2022.

[21] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proc. of CVPR*, 2020.

[22] Z. Huang, L. Hou, L. Shang, X. Jiang, X. Chen, and Q. Liu, "GhostBERT: Generate more features with cheap operations for BERT," in *Proc. of ACL*, 2021.

[23] F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," in *Proc. of ICLR*, 2019.

[24] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL*, 2019.

[26] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. of ICASSP*, 2020.

[27] S. Yang, P. Chi, Y. Chuang, C. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. Lin, T. Huang, W. Tseng, K. Lee, D. Liu, Z. Huang, S. Dong, S. Li, S. Watanabe, A. Mohamed, and H. Lee, "SUPERB: speech processing universal performance benchmark," in *Proc. of INTERSPEECH*, 2021.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of ICASSP*, 2015.

[29] Z. Zhu, X. Cheng, D. Chen, Z. Huang, H. Li, and Y. Zou, "Mix before Align: Towards Zero-shot Cross-lingual Sentiment Analysis via Soft-Mix and Multi-View Learning," in *Proc. of Interspeech*, 2023.

[30] D. Yang, S. Liu, J. Yu, H. Wang, C. Weng, and Y. Zou, "Norespeech: Knowledge distillation based conditional diffusion model for noise-robust expressive tts," *ArXiv preprint*, 2022.

[31] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," *ArXiv preprint*, 2023.

[32] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[33] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu *et al.*, "Audiogpt: Understanding and generating speech, music, sound, and talking head," *ArXiv preprint*, 2023.

[34] L. Hou, Z. Huang, L. Shang, X. Jiang, X. Chen, and Q. Liu, "Dynabert: Dynamic BERT with adaptive width and depth," in *Proc. of NeurIPS*, 2020.

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

[36] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," in *Proc. of ICLR*, 2020.

[37] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, 2020.

[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. of ICML*, 2010.

[39] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. of EMNLP*, 2016.

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

[41] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *ArXiv preprint*, 2021.

[42] B. Alexei, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. of NeurIPS*, 2020.