CLSR: End-to-end Contrastive Language-Speech Retriever For Better Speech Retrieval Augmented Generation

Anonymous ACL submission

Abstract

Significant progress has been made in spoken question answering in recent years. However, many of the existing methods including Large Audio Language Models (LALMs), have only been developed for short audio files and have difficulty in processing long audio. Speech Retrieval Augmented Generation (SRAG) follows the success of RAG in processing long-form speech, where an effective retriever serves as a critical first step. However, cross-modal retrievers in SRAG remain understudied, with current approaches either relying on pipeline methods (ASR followed by text RAG) or generic audio-text alignment models. To address this challenge, we propose proposes CLSR, an end-to-end contrastive language-speech retriever that efficiently extracts question-relevant segments from long audio recordings for downstream RAG processing. Unlike conventional speech-text contrastive models that directly align cross-modal representations, CLSR introduces an intermediate step by first mapping acoustic features into text-like representations before alignment, bridging the modality gap more effectively. Experimental results across four cross-modal retrieval datasets demonstrate that CLSR outperforms both end-to-end speech-text retrievers and pipeline approaches combining ASR with text retrieval. Our pre-trained CLSR model establishes a new state-of-the-art in cross-modal language-speech alignment, significantly surpassing previous general language-audio model like CLAP, thereby providing a robust foundation for advancing practical SRAG applications.

1 Introduction

Question Answering (QA) task requires the model to find the answer to a question from a given context. If the answer is a span in the context, then the task is called extractive QA; If the answer cannot be directly obtained from the context and requires further reasoning by the model, this task is called abstractive QA (Shih et al., 2023a). In the Spoken Question Answering (SQA) task, the given context is in audio format (Li et al., 2018), and some complex SQA tasks require questions also in audio format (Shon et al., 2022). Although there are many improvement on SQA (Lee et al., 2019; You et al., 2022), most SQA models are only applicable to short audio (less than 1 minute). In real life, many dialogue scenarios, such as meetings, lectures and online conversations, involve voice recordings of 10 minutes or more, which is difficult for existing SQA methods.

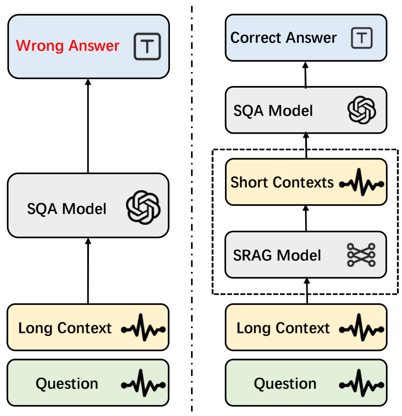

Figure 1: Using a small speech RAG model to simplify long audio context into several audio segments can help improve the quality of subsequent LALM response.

At present, Large Language Model (LLM) is developing rapidly. Represented by GPT (Brown, 2020) and LLaMA (Touvron et al., 2023), LLMs have achieved success in many traditional NLP tasks, including QA task. In the speech domain, there are also many LLMs that demonstrate impressive speech understanding capabilities (Chu et al., 2023; Radford et al., 2023). Retrieval augmented generation (RAG) introduces external knowledge into LLM to enhance their natural language understanding capabilities (Gupta et al., 2024). Specifically, it introduces a retriever before the LLM, which calculates the similarity between each chunk in the database and the user's input query, and then selects the top-k chunks with the highest similarity as additional inputs for the LLM. In this way, LLM can better understand the user's query and provide more satisfactory answers. For QA task, if the input context is a thousand-word article, the role of RAG is to extract the most relevant chunks from the article as the input for the LLM, avoiding the introduction of invalid information to decrease the answer accuracy and inference speed. Given this, in long SQA tasks, can we also use RAG to extract problem-related segments and use them as input for subsequent LALM?

In this paper, we propose CLSR, an end-to-end contrastive language-speech retriever, which simplifies long speech recordings into several audio clips that are most relevant to the question. Then the audio clips is used for subsequent LALM inference. Unlike typical end-to-end speech-to-text contrastive learning models, CLSR does not attempt to align acoustic representations and text representations into the same semantic space. Instead, it first converts the acoustic representations into text-like representations, and then aligns the text-like representations with the real text representations. For the extraction of text-like representations, we mainly use Continuous Integrate-and-Fire (CIF) to achieve the mapping of acoustic representations from time steps to token numbers, and then use an adaptor based on vector quantizer (VQ) to refine the acoustic representations into text-like representations. We compare CLSR with typical end-to-end speech-text retriever and pipeline retriever which combines speech-to-text model and text contrastive learning model on four datasets: Spoken-SQuAD, LibriSQA, SLUE-SQA-5, and DRCD. The experimental results show that CLSR has the strongest retrieval performance, which indicates that with text-like representation as a bridge between acoustic representation and text representation, CLSR can better capture the similarities and differences between the two modalities, thus more accurately pairing speech and text or speech and speech. The contributions of this paper are as follows:

(1) To our knowledge, this is the first work to introduce the concept of RAG into the field of SQA and use it to solve long speech problems.

(2) The CLSR we propose first converts acoustic representations into text-like representations, and then aligns the text-like representations with text representations, which can better alleviate modal differences and achieve cross-modal alignment.

(3) The proposed model achieves SOTA on four four datasets: Spoken-SQuAD, LibriSQA, SLUE-SQA-5 and DRCD.

2 Related Work

Currently, there are many works related to SQA. Chuang et al. (2019) propose a pre-trained model called SpeechBERT for the end-to-end SQA task. Through the training stage called initial phonetic spatial joint embedding for audio words, it aligns the generated audio embeddings with the text embeddings generated by BERT in the same hidden space. Shih et al. (2023a) introduce GSQA, which empowers the SQA system to engage in abstractive reasoning. They firstly utilize HuBERT to convert the input speech into discrete units, then use a sequence-to-sequence SQA model finetuned from text QA model, LongT5, to generate answers in the form of discrete units. Lin et al. (2024) foucus on the open-domain SQA and the scenario where paired speech-text data is unavailable. They propose SpeechDPR, which uses the bi-encoder retriever framework and learns a sentence level semantic representation space by extracting knowledge from the combined model of ASR and text retriever. Johnson et al. (2024) introduce a retriever that employs deep Q-learning to bypass irrelevant audio segments in longer audio files, enhancing SQA efficiency. The latter two articles are related to retriever, which is similar to our paper, but they have defects: the performance of the former is worse than that of the pipeline model, and the latter can only segment the audio at a fixed length, which can not guarantee that all the key information is in the same segment.

Since the birth of GPT, RAG has developed rapidly, while speech RAG has less work. Yang et al. (2024) use RAG for spoke lanauage understanding (SLU). They first use a pre-trained ASR encoder to extract acoustic features, and then use similarity calculation to find similar audio-text label pairs in the training set, and then introduce the label information into the SLU decoder through the cross attention mechanism. Wang et al. (2024) propose a joint speech and language model based on RAG, which can better perform the name entity recognition task. They calculate the similarity between the input speech query embeddings and the entity embeddings in the database to extract K entities most related to the problem, and use these entities as additional inputs to the model. There is currently no SRAG model for long SQA task.

3 Method

3.1 Preliminary

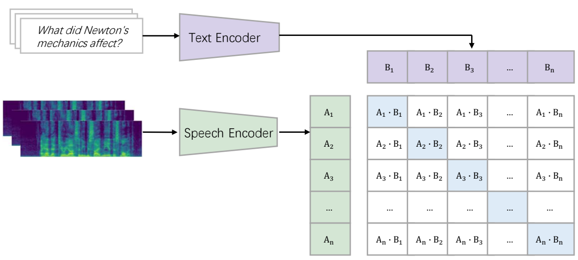

Figure 2: The architecture of typical end-to-end speech-text contrastive model.

Take the SQA task whose questions are in text format and contexts are in speech format as the example. Let X be the context, which is a speech sequence with T frames, $X = \{x_1, x_2, x_3, \ldots, x_t\}$. Let Y be the question, which is a sequence of tokens, and its length is n. Each token is in the vocabulary V, $Y = \{y_1, y_2, y_3, \ldots, y_n \mid y_i \in V\}$. Figure 4 shows the architecture of typical end-to-end text- speech contrastive model, such as CLAP (Wu et al., 2023). This kind of model first uses a speech encoder $A(.)$ and a text encoder $B(.)$ to extract acoustic features $A(X)$ and text features $B(Y)$, respectively, and then uses cosine similarity to characterize the similarity Z between the two features. The formula is as follows, where $||.||$ refers to taking the L2 norm.

$$Z_{X,Y} = ||A(X)|| \cdot ||B(Y)||$$

The features contrastive learning model used are sentence level. There are generally two methods for extracting sentence level features. One is to introduce a trainable `CLS` token and encode it together with other tokens. Then the score of the `CLS` token is used as the feature of the entire sentence; Another method is to average the token-level features of length n into the features of length 1. These two methods are also applicable for extracting features of the entire audio.

When training, the model learns to minimize the negative log likelihood (NLL) between the representation of the question and its paired context. The NLL loss is divided into two parts, one is the retrieval from question to context, and the other is the retrieval from context to question. The specific formula is as follows, where n refers to the total number of problem context pairs in the dataset.

$$NLL_{A,B} = -\frac{1}{2}\left(\sum_{i=0}^{n} log\frac{e^{Z_{X,y_i}}}{e^{Z_{A,Y}}} + \sum_{i=0}^{n} log\frac{e^{Z_{x_i,Y}}}{e^{Z_{X,Y}}}\right)$$

3.2 Overview

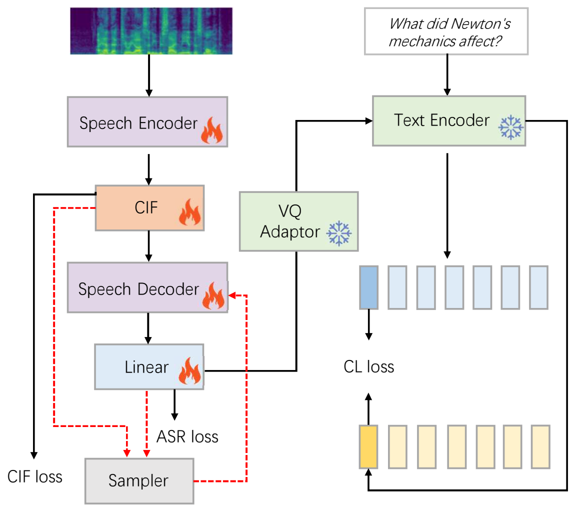

Figure 3: The architecture of proposed model, CLSR. CIF stands for Continuous Integration and File, while VQ stands for vector quantizer. The red line is only used during training.

Figure 4 shows the specific architecture of CLSR. The left half is a non-autoregressive attention encoder-decoder framework based on CIF (Dong and Xu, 2020). It receives the speech context X and outputs the corresponding token probability distribution D, $D = \{d_1, d_2, d_3, \ldots, d_n\}$. Both speech encoder and decoder adopt the SAN-M (Gao et al., 2020) structure, which is a special Transformer (Vaswani et al., 2017) layer that combines self-attention mechanism with deep feed-forward sequential memory networks (DFSMN). Firstly, the framework uses the speech encoder to extract acoustic features H^s.

$$H^s = SpeechEncoder(X)$$

And then maps H^s from the time step to the number of tokens through the soft and monotonic alignment mechanism, CIF, obtaining an acoustic representation E^a, which is aligned with the token probability distribution.

$$E^a = CIF(H^s)$$

Then, it predicts the corresponding token distribution through the speech decoder and a full-connected layer.

$$D = W \cdot Decoder(H^s, E^a) + b$$

Follow Gao et al. (2022), we use a sampler to optimize the training process of this framework. The sampler does not contain learnable parameters and aims to enhance the context modeling ability of the decoder by sampling text features into E^a.

The right half of CLSR is a Transformer-based text encoder that receives either a text embeddgins E^Y or a text-like embeddings $E^{Y'}$ as input and output corresponding text representation. We get the sentence-level representation by inserting `CLS` token.

$$H^t = TextEncoder(E^Y)$$

The text-like embeddings is obtained by mapping the token distribution through the VQ adaptor.

$$E^{Y'} = VQAdaptor(D)$$

3.3 Continuous Integrate-and-Fire

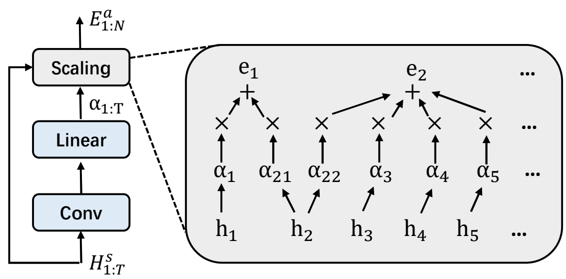

Figure 4: The explanation of CIF workflow. The gray box on the right shows an example of CIF, where $\alpha = \{0.8, 0.3, 0.4, 0.4, 0.1\}$ and the threshold β=1.

Figure 4 explains the workflow of CIF. Through convolution operation and linear mapping, it calculates the weight distribution α, $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_t \mid \alpha_i \in [0, 1]\}$. Each α_i shows the valid information contained in relevant h_i of the acoustic feature $H^s_{1:T}$.

$$\alpha_{1:T} = W \cdot conv(H^s_{1:T}) + b$$

Then, it gathers the weights and combines $H^s_{1:T}$ until the total weight hits a specified threshold β, signaling that an acoustic boundary has been attained. When reaching the threshold, if the current state of α overflows, it will be used for the next round of weight accumulation. The right side of Figure 4 provides an example of a scaling process, where $\alpha = \{0.8, 0.3, 0.4, 0.4, 0.1\}$ and the threshold β=1. It is clear that $\beta - \alpha_1 = 0.2 < \alpha_2$, so α_2 is divided into $\alpha_{21} = 0.2$ and $\alpha_{22} = 0.1$, where α_{21} is used to calculate the first integrated embedding c_1 and α_{21} is used for subsequent embedding calculations. So, $e_1 = \alpha_1 \times h_1 + \alpha_{21} \times h_2$, and $e_2 = \alpha_{22} \times h_2 + \alpha_3 \times h_3 + \alpha_4 \times h_4 + \alpha_5 \times h_5$.

3.4 Sampler

To enhance the ability of the selected non autoregressive AED framework to model token probability distributions, we introduce a training optimization module called sampler. If we enable sampler, the training of the framework will become two rounds. In the first round of training, we do not use samplers and directly use the acoustic features E^a obtained from the CIF module to predict the probability distribution of tokens. Through $argmax$, we can obtain the transcription result Y^{asr}.

$$Y^{asr} = \arg\max_{y_i \in V}(W \cdot Decoder(H^s, E^a) + b)$$

By comparing Y^{asr} with the real context Y^{con}, we can determine the tokens with transcription errors and their locations. In the second round of training, sampler is enabled. It strengthens acoustic representation E^a by incorporating text features E^c, which is the embedding of Y^{con}. Specifically, the sampler combines the correct embeddings of error tokens in E^c into E^a, and generates the semantic features E^s. Not every error token's correct embedding will be incorporated into E^a, this is determined by the mixing ratio λ, $\lambda \in (0, 1)$.

$$E^s = \text{sampler}(E^a, E^c, \lceil \lambda \sum_{i=1}^{N} (y_i^{asr} \neq y_i^{con}) \rceil)$$

Afterwards, use E^sinstead of E^a to calculate the probability distribution of the tokens.

$$D' = W \cdot Decoder(H^s, E^s) + b$$

It should be noted that, during the first pass of training, no gradient backpropagation is performed and Y^{asr} is only used to determine the sampling number of the sampler. D' obtained in the second pass is used to calculate the ASR loss.

Regarding the real text embeddings E^c, Gao et al. (2022) uses the embedding layer of the speech decoder to obtain it. However, in our proposed model, this layer is not trained and its weights will be difficult to represent the text embedding space. Therefore, we use the weights of linear layer which is used to obtain the probability distribution of the tokens to calculate E^c.

$$E^c = W \cdot Y^{con}$$

3.5 Adaptor

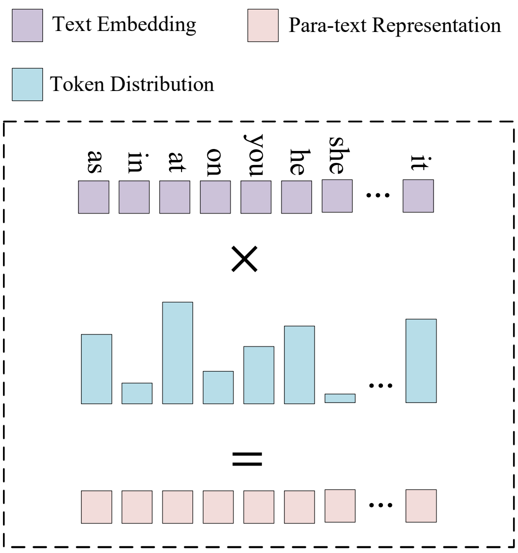

Figure 5: The mapping process of the adaptor.

After obtaining the probability distribution D of the tokens, we will use an adaptor to map it to the text-like embedding $E^{Y'}$. The adaptation process is divided into two steps: quantification and mapping. The quantization process converts the probability distribution of each token into the index of token which has the highest probability in the vocabulary. The design of VQ is inspired by (Shih et al., 2023b), we firstly choose the token index q_v with the highest probability in each token distribution d_{iV}, which can be expressed as:

$$q_v, where \ \ v = \underset{v_i \in V}{\arg\max} d_{iV}$$

q_v is not differentiable,if q_v is directly introduced into the training process, the computational graph will break. When not considering q_v, the value for gradient propagation should be the token probability distribution processed by $softmax$, P, and the formula for p_i is as follows, where γ is a hyper-parameter and we set $\gamma = 0.1$.

$$\overline{p_i} = softmax([D_{i1}, \dots, D_{iV}]^T / \gamma)$$

Through straight-through gradient estimator (Bengio et al., 2013), we can remove p_i from the computational graph and introduce $q_v$into the graph while ensuring gradient continuity. The specific formula is as follows, where $sg(x) = x$ and $\frac{d}{dx} sg(x) = 0$ is the stop gradient operator.

$$p_i == q_v + \overline{p_i} - sg(\overline{p_i})$$

Let's denote the quantized token probability distribution as D^{vq}. Next, we will map the distribution to the embedding layer of the text encoder. The specific operation is showed in the 5, that is, multiplying distribution and the weights of embedding layer in a matrix.

$$E^{Y'} = Matmul(D^{vq}, W^{te})$$

3.6 Loss Function

The adopted framework calculates three loss functions when training: the cross-entropy (CE), the mean absolute error (MAE), and the minimum word error rate (MWER) loss. CE and MWER are used to optimize the model's transcription ability, while MAE guides the CIF to convergence. According to Gao et al. (2022), the loss function of the ASR part is:

$$\mathcal{L}_{ASR} = \gamma \mathcal{L}_{CE} + \mathcal{L}^N_{werr}(x, y^*)$$

$$\mathcal{L}^N_{werr}(x, y^*) = \sum_{y_i \in sample} p(y_i \mid x)[\mathcal{W}(y_i, y^*) - W]$$

We also use NLL loss to optimize the model's ability for aligning the question representation and context representation. The total loss function can be formulated as follows, where α and β are used to control the proportion of CIF loss and contrastive loss, $\alpha \in (0, 1)$, $\beta \in (0, 1)$.

$$\mathcal{L}_{total} = (1 - \alpha - \beta)\mathcal{L}_{ASR} + \alpha \mathcal{L}_{MAE} + \beta \mathcal{L}_{NLL}$$

4 Experiment

4.1 Configuration

Dataset	Language	Type		Size		
		Question	Context	Train	Val	Test
Spoken-SQuAD	English	Text	Speech	37,107	5,351	-
Spoken-SQuAD*	English	Text	Speech	29,227	3,884	-
LibriSQA	English	Text	Speech	104,014	2620	-
SLUE-SQA-5	English	Speech	Speech	46,186	1,939	2,382
DRCD*	Chinese	Speech	Speech	25,321	1,425	-

Table 1: Datasets used in experiments. The dataset with asterisks has been filtered to achieve one-to-one correspondence between problems and contexts

We conduct experiments on four datasets: Spoken-SQuAD (Li et al., 2018), LibriSQA (Zhao et al., 2024), SLUE-SQA-5 (Shon et al., 2022), and DRCD. Table 1 displays detailed information about these datasets.

Li et al. (2018) use Google text-to-speech (TTS) system to generate the spoken version of the articles in SQuAD (Rajpurkar, 2016). Considering that SQuAD is a many-to-one dataset, where multiple questions correspond to the same context, it is not suitable for training text-speech retrievers. Therefore, we filter the original Spoken-SQuAD dataset to ensure that each question and context corresponded one-to-one, and the filtered dataset is referred to as Spoken SQuAD*.

LibriSQA is adapted from the ASR dataset librispeech (Panayotov et al., 2015). The authors input the textual document of each speech segment into Librispeech into ChatGPT and request ChatGPT to generate corresponding text question-answer pairs. We use the first part of LibriSQA which presents questions without options, and the answers are complete sentences.

SLUE-SQA-5 is adapted from 5 text QA datasets and the questions and contexts in it are all authentic audio recordings. DRCD (Shao et al., 2018) is originally a Chinese QA dataset. Similar to SQuAD, it is also a many-to-one dataset. We first filter it into a one-to-one dataset, and then use the TTS model (Li et al., 2020) to synthesize the speech versions of each question-context pair for its training set. Lee et al. (2018) offer spoken version of DRCD's dev set and we use it for testing.

We use 220M Paraformer (Gao et al., 2022) and BGE-base (Chen et al., 2024) to build CLSR. And BGE is freezed when training. We consider two models as baseline: one is the end-to-end text-speech contrastive model like Fig 4, and the other is the cascaded model that first uses automatic speech recognition (ASR) model to convert speech into text and then performs text QA task. For the former, we choose CLAP and SpeechDPR for comparison. For the latter, we use Whisper (Radford et al., 2023), which is promising in ASR, as ASR module and BGE-base as the text QA module. The Whisper's size is 244M. In the experiment, word error rate (WER) is used to measure the ASR performance, and top-k question-to-context and context-to-question retrieval recall are used to measure the retrieval performance. We build the experiment environment based on Funasr (Gao et al., 2023) and ModelScope. The α and β of the loss is set to $\frac{1}{3}$. We train until the model converges and the training epoch is at most 60. We consistently use the Adam optimizer with a learning rate of 5e-5, and the training is conducted on a GeForce RTX-3090.

4.2 Main Result

Table 2 shows the comparison results of CLSR and other models on four datasets. We additionally provide the results of using BGE for clean text question-context retrieval. In terms of end-to-end text-to-speech contrastive models, the results of CLSR are significantly better than those of CLAP and SpeechDPR. We found that CLAP cannot learn the relevance between text question and speech context well on Spoken-SQuAD* and LibriSQA, which indicates that CLAP is not suitable for text-to-speech content alignment. In fact, CLAP is more suitable for audio and text alignment. Additionally, since CLAP cannot perform speech to speech alignment, we do not perform experiments on the other two datasets.

SpeechDPR is committed to using text-less data for training. Although they use ASR models and text QA models for knowledge distillation, the lack of data makes it difficult for them to achieve good performance. It is worth noting that we do not conduct large-scale pre-training before training CLSR. All excellent contrastive learning models like BGE have undergone long-term pre-training, so they have strong retrieval capabilities. Nonetheless, CLSR still achieves results second only to BGE for clean text retrieval and even exceeded BGE's results on Spoken-SQuAD*, which reflects the superiority of CLSR's structure.

Compared with conventional end-to-end contrastive models that directly perform text-to-speech alignment (or speech-to-speech alignment), CLSR uses text-like representations to alleviate the differences between speech and text modalities. It first maps speech representations into text-like representations, and then aligns the text-like representations with the real text representations (or text-like representations with text-like representations) on the text modality. With the powerful performance of text contrastive models, this can better achieve alignment between speech and text (or speech and speech), thereby more accurately pairing with the context closest to the question.

When conducting a comparative analysis of CLSR and Whisper+BGE, we find that their retrieval performances on three English datasets are very close, but CLSR had certain advantages. In terms of transcription ability, CLSR is significantly stronger than WhisBGE. This shows that joint train-

Dataset	Model	Paradigm	Type		ASR	Q-C Retrieval (↑)			C-Q Retrieval (↑)		
			Question	Context	WER (↓)	R@1	R@5	R@10	R@1	R@5	R@10
Spoken-SQuAD*	BGE	E2E	Text	Text	0	67.12	85.20	89.44	65.63	84.14	89.06
	CLAP	E2E	Text	Speech	-	2.93	9.92	14.84	3.20	10.15	15.23
	Whisper+BGE	Pipeline	Text	Transcript	19.39	69.93	86.61	90.53	**67.97**	**85.76**	89.65
	CLSR	E2E	Text	Speech	**15.14**	**70.03**	**86.90**	**90.68**	67.84	85.69	**90.17**
LibriSQA	BGE	E2E	Text	Text	0	86.91	94.31	95.92	86.87	94.73	96.60
	CLAP	E2E	Text	Speech	-	0.04	0.19	0.38	0.08	0.19	0.50
	Whisper+BGE	Pipeline	Text	Transcript	4.32	83.70	93.28	94.92	85.15	93.40	95.27
	CLSR	E2E	Text	Speech	**4.09**	**85.04**	**93.36**	**95.04**	**85.53**	**94.01**	**95.57**
SLUE-SQA-5	BGE	E2E	Text	Text	0	38.71	72.26	84.34	35.68	70.11	82.28
	SpeechDPR	E2E	Speech	Speech	-	-	-	19.94*	-	-	-
	Whisper+BGE	Pipeline	Transcript	Transcript	36.41	29.98	60.41	72.71	29.85	60.75	**73.47**
	CLSR	E2E	Speech	Speech	**16.69**	**30.65**	**62.19**	**74.43**	**29.89**	**62.18**	73.05
DRCD*	BGE	E2E	Text	Text	0	90.67	97.12	98.74	89.26	97.75	98.39
	CLSR	E2E	Speech	Speech	**5.56**	76.21	87.79	90.03	75.23	88.21	91.51

Table 2: Main results of proposed model in four datasets. Results for BGE are included as a reference benchmark, showing theoretical limits under optimal ASR conditions (100% accuracy). The SpeechDPR's paper just offers the result of R@20. CLAP is composed of HTSAT (Chen et al., 2022) and RoBERTa (Liu, 2019).

ing of CLSR can optimize both the ASR module and the contrastive learning module. Considering that Whisper's Chinese speech recognition ability is not outstanding, we don't train Whisper on DRCD*.

4.3 Ablation Result

To demonstrate the effectiveness of the quantizer and sampler in CLSR, as well as the possibility of multi-stage training to improve model performance. We conduct a series of ablation experiments on Spoken-SQuAD, and the results are shown in Table 3. The first two rows of the results show the value of the quantizer. When the quantizer is not used, although the model can have a lower WER, the model's comparative learning ability will significantly decrease: The top-10 retrieval recall rate of "CLSR w/o VQ" can only be comparable to top-1 retrieval recall rate of "CLSR w/ VQ". The results of the sixth and seventh rows show the effectiveness of sampler. After introducing sampler, CLSR not only improves retrieval ability, but also improves ASR performance.

Before joint training, we can pre-train the ASR module and BGE module of CLSR separately. In the experiment, we use 460 hours of clean librispeech data to pre-train Paraformer, and use Spoken-SQuAD's clean text question-context pairs to train BGE. Comparing the second and fourth rows of the experimental results, it is not difficult to find that pre-training BGE is meaningful, and using pre-trained BGE in joint training improves the various retrieval metrics of CLSR by about 6%. In addition, through the comparison between the fourth and sixth rows, it can be found that pre-training Paraformer can improve the model's transcription performance while also slightly improving its retrieval ability. It should be noted that in order to improve the training speed of the model, we froze BGE, which has strong retrieval performance, during joint training. Therefore, we can freeze the ASR module after joint training and train BGE for a few epochs separately, which is called post-train in the table. It is hoped that this approach can make BGE better adapt to the text-like representation provided by the ASR module. Unfortunately, post-train can only slightly improve the performance of the model, as evidenced by rows 2 and 3, 4 and 5, 7 and 8 in the table. In short, through ablation experiments, we have shown that both quantizers and samplers are inseparable for CLSR, and that pre-training the ASR module and BGE module of CLSR is of significant importance.

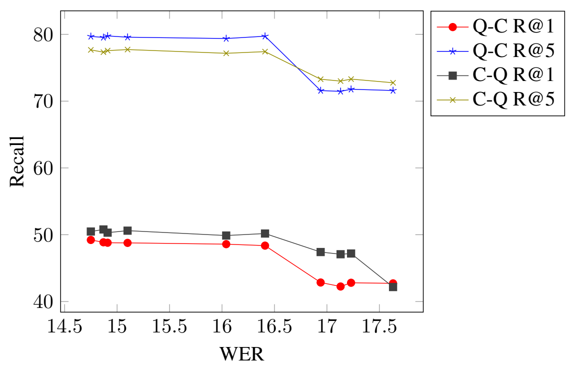

Figure 6: The correlation between the retrieval ability and speech recognition ability of CLSR.

Pre-train		Joint-train		Post-train	ASR	Q-C Retrieval (↑)			C-Q Retrieval (↑)		
ASR	BGE	VQ	Sampler	BGE	WER (↓)	R@1	R@5	R@10	R@1	R@5	R@10
×	×	×	×	×	16.13	15.29	34.14	44.18	15.75	36.11	46.16
×	×	✓	×	×	17.00	42.52	71.46	78.36	46.86	72.66	79.95
×	×	✓	×	✓	17.00	45.11	75.31	82.90	48.05	75.82	83.18
×	✓	✓	×	×	17.00	48.10	78.28	84.98	49.45	76.79	83.42
×	✓	✓	×	✓	17.00	48.31	78.55	84.73	50.08	77.16	83.68
✓	✓	✓	×	×	16.18	49.00	79.20	85.69	50.31	77.48	84.21
✓	✓	✓	✓	×	15.01	49.65	79.61	85.91	50.59	77.71	84.38
✓	✓	✓	✓	✓	**15.01**	**49.82**	**79.63**	**85.83**	**50.63**	**77.69**	**84.56**

Table 3: Ablation results in Spoken-SQuAD.

Dataset	Model	Paradigm	ASR	Q-C Retrieval (↑)			C-Q Retrieval (↑)		
			WER (↓)	R@1	R@5	R@10	R@1	R@5	R@10
Spoken-SQuAD	ParaBGE	E2E	-	17.79	38.68	48.35	17.03	38.31	48.91
	CLSR	E2E	15.01	**49.82**	**79.63**	**85.83**	**50.63**	**77.69**	**84.56**
LibriSQA	ParaBGE	E2E	-	29.31	50.27	59.70	20.57	39.28	49.28
	CLSR	E2E	4.09	**85.04**	**93.36**	**95.04**	**85.53**	**94.01**	**95.57**
SLUE-SQA-5	ParaBGE	E2E	-	7.31	21.83	32.75	7.52	21.96	33.12
	CLSR	E2E	16.69	**30.65**	**62.19**	**74.43**	**29.89**	**62.18**	**73.05**

Table 4: Comparison results between traditional E2E contrastive model and CLSR.

To evaluate the impact of transcription error on CLSR's retrieval ability, we conduct the experiment on Spoken-SQuAD and present the results on Fig 6. Overall, WER is positively correlated with retrieval recall rate, with smaller WER resulting in higher recall rates. Specifically, on Spoken-SQuAD, the WER of approximately 16.75 is the watershed of CLSR retrieval capability. If the WER is greater than 16.75, the recall rate of the model will significantly decrease.

In order to further demonstrate the superiority of the proposed model over the traditional E2E speech-related contrastive model which is composed of two encoders, we construct a new baseline: ParaBGE, to compare the retrieval capability with CLSR. ParaBGE is composed of speech encoder of Paraformer and text encoder of BGE. The size of each module in both models are the same as those in CLSR. The experimental results are shown in Table 4. All retrieval metrics of CLSR far exceed ParaBGE, indicating that CLSR has a stronger question-context alignment ability. Although ParaBGE can optimize parameters towards the direction of aligning question and context representation during training, its performance is not ideal. As we mentioned earlier, such model heavily rely on pre-training with large-scale corpora. However, high-quality speech-text pairs are already very scarce, so for E2E speech related retrieval models, it is difficult to achieve excellent results. However, CLSR alleviates the modal differences between speech and text by using text-like representation as a bridge, shifting the alignment of speech to text alignment. With the powerful generalization ability of text contrastive learning models, it can achieve excellent retrieval capabilities comparable to cascade models and text contrastive models without the need for long-term, large-scale pre-training.

5 Conclusion

In this paper, we propose CLSR, an end-to-end contrastive language-speech retriever, which can simplifies long speech recordings' clips into a few clips that are most relevant to the question. By using text-like representation as a transition state, CLSR can better achieve cross-modal or speech modal alignment between question and context than ordinary end-to-end speech-related contrastive models. The experimental results show that the retrieval performance of CLSR not only far exceeds existing end-to-end speech-related retriever, but is also comparable to cascaded models and text retriever. In the future, we will attempt to combine CLSR with LALM to enable it to perform various complex long audio comprehension tasks.

Limitations

While CLSR demonstrates strong performance in speech retrieval tasks, there are two limitations. First, the current model primarily focuses on speech content, but future work could extend its capabilities to handle general audio signals, including environmental sounds, music, and other acoustic events, thereby enabling more comprehensive audio-based retrieval augmented generation. Second, the present implementation is limited to single-language support, necessitating future development of multilingual capabilities through additional training on diverse language datasets. These extensions would significantly enhance the model's versatility and practical applications across different audio domains and linguistic contexts.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2022. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 646–650. IEEE.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Yung-Sung Chuang, Chi-Liang Liu, Hung-Yi Lee, and Lin-shan Lee. 2019. Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering. *arXiv preprint arXiv:1910.11559*.

Linhao Dong and Bo Xu. 2020. Cif: Continuous integrate-and-fire for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083. IEEE.

Zhifu Gao, Zerui Li, Jiaming Wang, Haoneng Luo, Xian Shi, Mengzhe Chen, Yabin Li, Lingyun Zuo, Zhihao Du, Zhangyu Xiao, et al. 2023. Funasr: A fundamental end-to-end speech recognition toolkit. *arXiv preprint arXiv:2305.11013*.

Zhifu Gao, Shiliang Zhang, Ming Lei, and Ian McLoughlin. 2020. San-m: Memory equipped self-attention for end-to-end speech recognition. *arXiv preprint arXiv:2006.01713*.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. 2022. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. *arXiv preprint arXiv:2206.08317*.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Alexander Johnson, Peter Plantinga, Pheobe Sun, Swaroop Gadiyaram, Abenezer Girma, and Ahmad Emami. 2024. Efficient sqa from long audio contexts: A policy-driven approach. In *Proc. Interspeech 2024*, pages 1350–1354.

Chia-Hsuan Lee, Yun-Nung Chen, and Hung-Yi Lee. 2019. Mitigating the impact of speech recognition errors on spoken question answering by adversarial domain adaptation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7300–7304. IEEE.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018. Odsqa: Open-domain spoken question answering dataset. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 949–956. IEEE.

Chia-Hsuan Li, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. 2018. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. *arXiv preprint arXiv:1804.00320*.

Naihan Li, Yanqing Liu, Yu Wu, Shujie Liu, Sheng Zhao, and Ming Liu. 2020. Robutrans: A robust transformer-based text-to-speech model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8228–8235.

Chyi-Jiunn Lin, Guan-Ting Lin, Yung-Sung Chuang, Wei-Lun Wu, Shang-Wen Li, Abdelrahman Mohamed, Hung-yi Lee, and Lin-Shan Lee. 2024. Speechdpr: End-to-end spoken passage retrieval for open-domain spoken question answering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12476–12480. IEEE.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. Drcd: A chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

Min-Han Shih, Ho-Lam Chung, Yu-Chi Pai, Ming-Hao Hsu, Guan-Ting Lin, Shang-Wen Li, and Hung-yi Lee. 2023a. Gsqa: An end-to-end model for generative spoken question answering. *arXiv preprint arXiv:2312.09781*.

Yi-Jen Shih, Hsuan-Fu Wang, Heng-Jui Chang, Layne Berry, Hung-yi Lee, and David Harwath. 2023b. Speechclip: Integrating speech with pre-trained vision and language model. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 715–722. IEEE.

Suwon Shon, Siddhant Arora, Chyi-Jiunn Lin, Ankita Pasad, Felix Wu, Roshan Sharma, Wei-Lun Wu, Hung-Yi Lee, Karen Livescu, and Shinji Watanabe. 2022. Slue phase-2: A benchmark suite of diverse spoken language understanding tasks. *arXiv preprint arXiv:2212.10525*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Mingqiu Wang, Izhak Shafran, Hagen Soltau, Wei Han, Yuan Cao, Dian Yu, and Laurent El Shafey. 2024. Retrieval augmented end-to-end spoken dialog models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12056–12060. IEEE.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Hao Yang, Min Zhang, Daimeng Wei, and Jiaxin Guo. 2024. Srag: speech retrieval augmented generation for spoken language understanding. In *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, pages 370–374. IEEE.

Chenyu You, Nuo Chen, Fenglin Liu, Shen Ge, Xian Wu, and Yuexian Zou. 2022. End-to-end spoken conversational question answering: Task, dataset and model. *arXiv preprint arXiv:2204.14272*.

Zihan Zhao, Yiyang Jiang, Heyang Liu, Yu Wang, and Yanfeng Wang. 2024. Librisqa: A novel dataset and framework for spoken question answering with large language models. *IEEE Transactions on Artificial Intelligence*.