QFORMER BASED TEXT AUDIO RETRIEVAL SYSTEM

Technical Report

Ziye Fan

Lingban Technology Ltd.
Beijing, China
zyfan@ling-ban.com

Fengyun Zhu

Lingban Technology Ltd.
Beijing, China
fyzhu@ling-ban.com

ABSTRACT

This paper describes the system we submitted for DCASE2023 Challenge Task 6B. Task 6B involves audio retrieval using natural language. Our submitted retrieval system includes a frozen pre-trained audio encoder and a Qformer as text encoder. The system utilizes paired data provided by the AudioCaps and Clotho datasets for contrastive learning in the style of BLIP-2. Natural language query requests are first encoded by the text encoder, followed by a top-k recall in the pre-extracted audio embeddings. These are then paired with the query text to form k pairs of data, which are reranked based on the model's matching ability to produce the final retrieval results. This system achieved an mAP of 26.47% and a 16.02% R@1 on the Clotho test set, while the baseline system's performance being mAP of 22.2% and 13.0% R@1.

***Index Terms*—** Text-to-audio retrieval, contrastive learning, Qformer

1. INTRODUCTION

DCASE 2023 task 6b is about retrieving audio using natural language. [1] It is an important problem in cross-modal research field. The progress in this task will deepen the understanding of acoustic scenes and will enable us to manipulate audio signals in creative ways, which will in turn impact many application areas, such as audio content creation and acoustic scene analysis, among others.

In this work, we use the BEATs model[2] as our audio feature extractor. BEATs is a pre-trained audio model based on self-supervised learning. It is pre-training by iteratively training a discrete Tokenizer and a feature extractor guided by Masked Audio Modeling, and is subsequently applied to downstream tasks such as classification.

The emergence of CLIP[3] greatly advanced the progress of visual language multimodal models, followed by models like BLIP-2 [4] that aim to bridge multimodal and Large Language Models. In the field of audio-related multimodal research, there are also works like AudioClip[5], Wav2Clip [6], and CLAP [7] that extend CLIP-style contrastive learning to audio signals. In this work, we apply contrastive learning of audio and natural language based on the Qformer and its corresponding multitask training method that are proposed by BLIP-2[4]. Qformer is a transformer encoder similar to BERT; it can handle individual audio inputs or text inputs to obtain a single-modal representation. It can also encode both inputs of audio and text and produce multimodal embeddings, which can then be used to tell if the audio and text are matched.

In this paper, details of the system are presented in Section 2, the data and setup of experiments are introduced in Section 3, and then results of the experiments are in Section 4.

2. SYSTEM DESCRIPTION

This system comprises an audio encoderand a Qformer. Figure 1 shows the overall structure of the retrieval system. The number of total trainable parameters is about 189M.

2.1. Audio Encoder

The audio feature extractor is the BEATs model, which consists of a 12-layer transformer encoder. This feature extractor accepts a monophonic audio signal input s, with a duration of T and a sample rate of 16000Hz. It outputs T' frames of features, where $T' = \frac{T}{0.16}$, and each frame feature has a size of 8×768. We use a pre-trained model[1] with frozen weights, which has been trained on the AudioSet-2M [8] dataset.

2.2. Qformer

Qformer uses a network structure similar to bert-base and is also initialized with the latter's weights. The input of Qformer are fixed-length learnable query tokens. Audio features enter the network through cross attention, while text features enter the network from the input layer. We follow the pre-training method in BLIP-2 to train Qformer, simultaneously optimizing three training criteria. These are 1) Audio Text Contrastive Learning(ATC), where we input audio features and text features separately, obtain their respective embeddings, and then optimize them using InfoNCE loss [9] to align the audio features with the text features. 2) Audio Text Matching(ATM), where a binary classification task is used for training, and the network is asked to determine if the input audio-text pair matches. 3) Audio-grounded Text Generation(ATG), where the network is trained to generate text descriptions based on the input audio.

During training, paired audio and text are first used to train under ATC criterion. Then, based on the similarity of the audio and text embeddings, a certain number of hard negative samples are sampled separately for the audio and text in each pair of data

[1] https://valle.blob.core.windows.net/share/BEATs/BEATs_iter3_plus_AS2M.pt?sv=2020-08-04&st=2023-03-01T07%3A51%3A05Z&se=2033-03-02T07%3A51%3A00Z&sr=c&sp=rl&sig=QJXmSJG9DbMKf48UDIU1MfzIro8HQOf3sqlNXiflY1I%3D

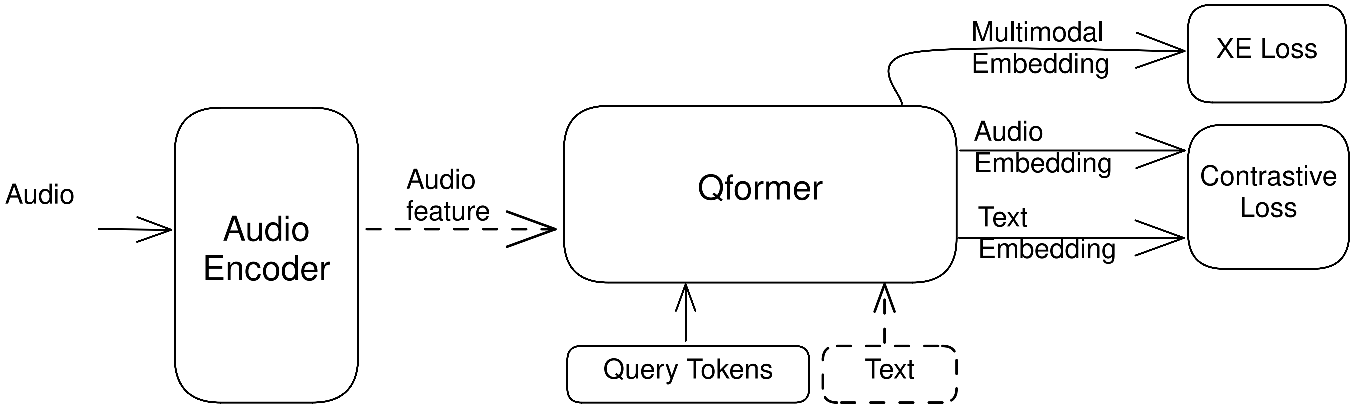

Figure 1: An overview of the whole system

within the same batch. These samples form the pairs used for ATM training. Finally, ATG training is performed.

During inference, we first calculate the ATC embeddings $\boldsymbol{ai}$ for all candidate audios. For each natural language query, we calculate the query's ATC embedding $\boldsymbol{tj}$, and then calculate its distance s from all audio embeddings.

$$s = \frac{\boldsymbol{a}_i \cdot \boldsymbol{t}_j^T}{||\boldsymbol{a}_i|| \cdot ||\boldsymbol{t}_j||}$$

The audio corresponding to the representation with the shortest distance is taken as the retrieval result. Optionally, we can use Qformer's ATM head to re-rank the k audios most similar to the query text.

3. EXPERIMENTS

AudioCaps[10] and ClothoV2[11] datasets are utilized to train the model. However, their original splits are not used. Instead, we combined the training and validation sets of AudioCaps and ClothoV2, randomly drawing 1000 pieces of data as the validation set, with the remaining parts serving as the training set. The evaluation set of ClothoV2 is also used as a validation set.

Audio feature extraction is run separately before the model training begins to avoid too much GPU memory being consumed during training. For audio longer than 10 seconds, its features are randomly truncated to 512 (approximately 10.24 seconds) during training.

The model is trained on the AudioCaps and ClothoV2 datasets. Experiments are conducted using Adam optimizer with a learning rate set at 0.0001. The learning rate scheduling strategy is cosine annealing with a period of 1800 steps, and there is a 1000-step linear warmup at the start of training. We train the model on a machine with RTX2080Ti x4, with a batch size of 16 per card, and achieve a batch size of 256 through gradient accumulation. Subsequently, the model is fintuned on the ClothoV2 dataset with a learning rate set at 0.00001.

4. RESULTS

We evaluate the performance of four systems on the ClothoV2 test set, the results are shown in Table 1. All of our four systems performs significantly better than the baseline system.

- System 1: Not fine-tuned on the Clotho dataset, no reranking performed
- System 2: Not fine-tuned on the Clotho dataset, reranking performed on the top-16 results
- System 3: Fine-tuned on the Clotho dataset, no reranking performed
- System 4: Fine-tuned on the Clotho dataset, reranking performed on the top-16 results

Name	mAP@10	R@1	R@5	R@10
Baseline	22.2	13.0	34.3	48.0
System 1	25.30	14.68	40.08	54.22
System 2	26.18	15.44	**40.61**	**55.14**
System 3	**26.47**	**16.02**	40.19	54.39
System 4	25.08	14.56	38.97	53.26

Table 1: Results on Clotho evaluation set

5. CONCLUSIONS

This paper presents the system we submitted for DCASE2023 Task 6B. This system utilizes BEATs and Qformer to implement a text-to-audio retrieval system. By training on AudioCaps and ClothoV2 and fine-tuning on ClothoV2, the system achieved competitive performance.

6. REFERENCES

[1] A.-M. Oncescu, A. S. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio Retrieval with Natural Language Queries," July 2021.

[2] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers," Dec. 2022.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," https://arxiv.org/abs/2103.00020v1, Feb. 2021.

[4] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” Jan. 2023.

[5] A. Guzhov, F. Raue, J. Hees, and A. Dengel, “AudioCLIP: Extending CLIP to Image, Text and Audio,” https://arxiv.org/abs/2106.13043v1, June 2021.

[6] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, “Wav2CLIP: Learning Robust Audio Representations From CLIP,” https://arxiv.org/abs/2110.11499v2, Oct. 2021.

[7] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, “CLAP: Learning Audio Concepts From Natural Language Supervision,” https://arxiv.org/abs/2206.04769v1, June 2022.

[8] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[9] A. van den Oord, Y. Li, and O. Vinyals, “Representation Learning with Contrastive Predictive Coding,” https://arxiv.org/abs/1807.03748v2, July 2018.

[10] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating Captions for Audios in The Wild,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132.

[11] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An Audio Captioning Dataset,” https://arxiv.org/abs/1910.09387v1, Oct. 2019.