SCUT SUBMISSION FOR AUTOMATED AUDIO CAPTIONING USING GRAPH ATTENTION AND CROSS-ATTENTION MECHANISMS

Technical Report

Qianqian Li

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou,
China
leepp199@gmail.com

Yanxiong Li[*]

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou,
China
leeyxli@scut.edu.cn

ABSTRACT

This report presents our work for automated audio captioning which is the Task 6A of DCASE 2024. Our system is an encoder-decoder framework. The encoder uses a pre-trained ConvNeXt network and the decoder employs a standard Transformer structure. Among the encoders, we include a graph attention module to enhance the module's ability to extract audio features. In the decoder, in addition to the Transformer's multi-head self-attention mechanism, a cross-attention mechanism is added to improve the association between output subtitles and audio features. Finally, our system achieves FENSE score of 0.5131 which is higher than the baseline system's FENSE score of 0.5040.

Index Terms— Graph attention, cross-attention, encoder, decoder

1. INTRODUCTION

The Automated audio captioning (AAC) system implements a cross-modal translation task, using free-form text to describe regular audio content, rather than simply performing speech-to-text conversion. Unlike the audio event detection and audio classification tasks [1]-[4], AAC aims to capture spatio-temporal relationships in audio clips and perform advanced audio interpretation of the audio. The challenge on Detection and Classification of Acoustic Scene and Event (DCASE) plays an important role in promoting the AAC research.

A number of research teams recently conducted studies on the AAC. All the latest systems employ a pre-trained network as an audio encoder combined with a Transformer-like structure as a word decoder [5]. One researcher used a data enhancement technique where Gaussian noise and SpecAugment [6] were used to generate variants. Mixed audio waveforms and spectrograms were enhanced using the data augmentation technique of Mixup [7] during training and linked to corresponding caption labels. As with most models, they used a complete Transformer architecture and their system was currently leading on the Audiocaps dataset. Some studies have suggested the use of pre-trained decoders to improve the speech production part [8], where the authors used a decoder called BART to improve the quality of captions. They combined 2 audio encoders. The first audio encoder generates the sound timestamps detected in the audio file, and gives them to the BART input embedding layer and adds them to the audio embedding extracted from another audio encoder. This approach is expected to make the BART input closer to the input expected from the pre-trained weights. Some authors have used ConvNeXt [9], a network from the vision domain, in AAC technology and trained it with multiple datasets. Their studies have found that ConvNeXt works very well as an audio encoder for extracting audio features, and that in combination with a normal Transformer decoder, the quality of the generated captions is promising.

We use the ConvNeXt-transformer framework mentioned above. The first component is the audio encoder module, the module is initialized with parameters that are weights after pre-training on AudioSet, and this audio encoder extracts the features of the input audio as input to the model. The next component is a transformer language model, to which we add a graph attention mechanism that allows the downstream text decoder to produce more accurate caption output. The transformer itself is structured with a multi-head self-attention mechanism module, which helps the system to merge multiple levels of features so that the model is able to understand both audio and text. In addition, we have added a cross-attention module, which allows the language model to focus not only on the text decoder itself, but also on the audio feature inputs at the same moment, thus improving the smoothness and accuracy of the caption output. In addition to this, some common data enhancement techniques such as Mixup, Speed Perturbation [10]. Spec-Augment is also used to improve the generalization and robustness of the system.

2. SYSTEM DESCRIPTION

Our system is an encoder-decoder architecture, where the encoder is an audio feature extractor and the decoder is a language model. To improve the generalization of the system, we use data augmentation to increase diversity of the training data.

[*] Corresponding author (eeyxli@scut.edu.cn)

This work was partly supported by the national natural science foundation of China (62371195, 62111530145, 61771200), international scientific research collaboration project of Guangdong (2023A0505050116), Guangdong basic and applied basic research foundation (2022A1515011687), and Guangdong provincial key laboratory of human digital twin (2022B1212010004).

2.1. Audio feature extractor

The ConvNeXt model is based on Depthwise Separable Convolutions (DSC) [11] with an Inverted Bottleneck (IB) layer [12]. DSC consists of a sequence of depth convolutions that process the feature channels separately, followed by a pointwise convolution to mix them. This technique aims to produce similar results to standard convolutional layers while reducing the number of operations to speed up training and reduce overfitting. The IB layer is a sequence of pointwise convolutional layers that increase the number of channels, followed by a GELU [13] function activation. Finally, another pointwise convolutional layer restores the number of channels to the value they had at the bottleneck input. The output is then added to the original input to create residual connections, thus avoiding the problem of vanishing gradients and reducing the number of parameters compared to the standard residual block. It has been shown in previous work that when used as an encoder it can produce excellent audio features, which are essential for the decoder to generate accurate captions. On the basis of the original structure, we added the graph attention mechanism module [14], which can learn the correlation between audio feature nodes, and thus modelling the long-time dependence of audio signals and highlighting the important semantic information related to the sound field scenes and events. With the help of residual connections, the audio features learned by the graph attention mechanism can contain the local time-frequency pattern information extracted from the convolutional network at the same time, which can provide the downstream text decoder with effective features for the generation of more accurate captions. Figure 1 shows our application of graph attention in the encoder.

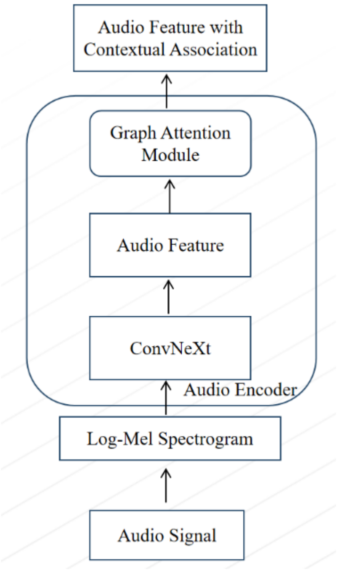

Figure 1: Applications of graph attention.

2.2. Language model

The encoder of the Transformer processes the input sequence, and extracts the features and semantic information from it. A multi-head self-attention mechanism and a feed-forward neural network are used to encode the input sequence and generate a set of context vectors. These context vectors capture the relevant information in the input sequence and provide the basis for subsequent output generation by the decoder. The decoder generates the output sequence step-by-step through a multi-head auto-regressive mechanism. In the decoder, the previous moment-generated output is used as the input of the current moment, which is combined with the attention mechanism to model the contextual information for generating the next output text character. In this report, based on the basic Transformer architecture, a cross-attention mechanism is added between the feed forward layer module and the multi-head auto-regressive module. The cross-attention mechanism can effectively integrate audio features and textual information deeply, so that the decoder can not only focus on the previously generated sequences, but also focus on the relevant audio features when generating each output character. It enables the decoder to focus not only on the previously generated sequence, but also on the relevant audio features when generating each output character. Hence, a text description is generated, which is more relevant to the content. Figure 2 shows our application of cross-attention in language model.

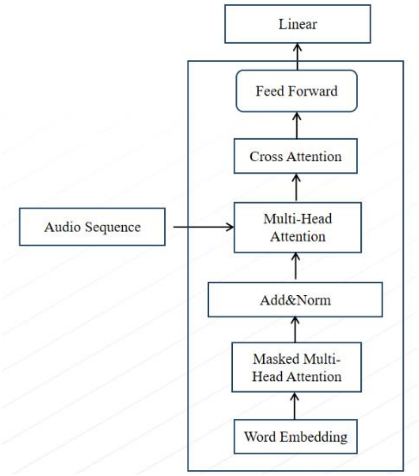

Figure 2: Applications of cross attention.

2.3. Data augmentation

In order to improve the generalization of the model and prevent the model from serious overfitting during the training process, we use some data enhancement methods on the training data, such as Mixup, Speed Perturbation, SpecAugment, etc. These techniques of data augmentation are introduced as follows.

2.3.1. Mixup

Firstly, this report uses the Mixup technique on the decoder inputs (i.e., audio embeddings and previously labelled embeddings) to improve the generalization and robustness of the model. Equation (1) presents a specific audio embedding Mixup method.

$$\begin{cases} \lambda \sim Beta(\alpha,\alpha) \\ \lambda = \max(\lambda, 1-\lambda) \\ x_{mix} = \lambda x_1 + (1-\lambda) x_2 \\ w_1 = W(y_1, prev) \\ w_2 = W(y_2, prev) \\ w_{mix} = \lambda w_1 + (1-\lambda) w_2 \\ z_{mix} = f(x_{mix}, w_{mix}) \\ L = CE(z_{mix}, y_1, next) \end{cases} \tag{1}$$

where λ denotes the mixing coefficients obtained by sampling using the Beta distribution; α denotes the parameters of the Beta distribution; x_1 and x_2 denote the two audio embeddings in the current batch; y_1 and y_2 denote the sequence of labels corresponding to x_1 and x_2, respectively; w_1 and w_2 denote the word embedding vectors corresponding to y_1 and y_2; W denotes the input word embedding layer; f denotes the rest of the AAC system decoder network; z_{mix} denotes the output of the mixed decoder; L denotes the cross-entropy loss function.

2.3.2. *Speed Perturbation*

This report also employs the technique of Speed Perturbation, which is used to simulate the speech signal by resampling the speech signal and changing its duration. Different speaking rates are simulated, which helps to enhance the diversity of training data and make the AAC system better adapt to various speaking rates without being limited to a certain speech rate. This helps to enhance the diversity of the training data, so that the AAC system can better adapt to a variety of speech rates instead of being limited to a specific speed range.

2.3.3. *SpecAugment*

The SpecAugment technique is applied to audio frame embedding in the output of an audio encoder. By using this technique, it is possible to mask the spectrogram or audio embedding so that a certain percentage of the time and feature axes are masked, rather than using an absolute mask size. Each axis is masked twice and the mask size is sampled at 0-10% of the total axis size (time step or embedding size).

3. EXPERIMENTS

3.1. Training

During the training process, we use the teacher coercion method to train the model. This method means that while training the model, we always take the correct previous word as input instead of using the previous word predicted by the model. This is different from the planned sampling method approach, which would use the previous word predicted by the model as input. In this way, the model can better learn how to generate the next word based on the previous word. The output of the model is the probability distribution of the next word. We compare this probability distribution with the correct next word to calculate the cross-entropy loss, a loss function that guides the model to better predict the next word. In the inference process, the model adds the most likely next word to the previous word, generating each word in the sentence in turn until an end marker occurs or the maximum number of words is reached.

3.2. Dataset

Clotho v2.1 [15] consists of three subsets of the released development sets: Dev-Training, Dev-Verify, and Dev-Test. The Dev-Training subset consists of 3,839 audio clips, while the Dev-Verify and Dev-Test subsets consist of 1,045 audio clips. The Dev-Test subset consists of 1,045 audio clips each. Each audio file in the dataset has a duration of 15 to 30 seconds. Five subtitles are provided for each file, ranging from 8 to 20 words in length.

3.3. Experiment setup

In the training process, we set the maximum number of training rounds to 300, the gradient shear threshold to 1, the number of training samples per unit batch to 64, and the initial learning rate to 5e-5 and decayed according to the cosine schedule. In the inference process, we use beam search method and set the beam size to 4.

4. RESULTS

The experimental results of the submission are shown in Table 1. The details of the submission methods are follows.

System1: ConvNeXt-trans model with graph attention.
System2: ConvNeXt-trans model with cross attention.
System3: ConvNeXt-trans model with graph attention and cross attention, beam size to be 3.
System4: ConvNeXt-trans model with graph attention and cross attention, beam size to be 4.

Table 1: The performance of the submission and baseline systems on Clotho. In all metrics, higher values indicate better performance.

Model	Baseline	System1	System2	System3	System4
METERE	0.1897	0.1870	0.1864	0.1874	0.1887
CIDEr	0.4619	0.4701	0.4601	0.4666	0.4690
SPICE	0.1335	0.1312	0.1328	0.1337	0.1339
SPIDEr	0.2977	0.3007	0.2965	0.3001	0.3015
SPIDEr-FL	0.2962	0.3002	0.2950	0.3001	0.3011
FENSE	0.5040	0.5073	0.5052	0.5121	0.5131
Vocabulary	551	487	593	482	477

4.1. System Output Captioning

Our system was tested on the evaluation set in the Clotho dataset, where each audio data is accompanied by a corresponding reference caption. Table 2 shows the comparison between the generated results and the reference captions for ten typical audio captions in the dataset. As shown in Table 2, The AAC system performs well in these audio content types and is able to generate semantics that match human characteristics by converting synonyms and so on.

Table 2 Comparison of system output captions with reference captions.

Reference caption	System Generated Captions
someone is trimming the bushes with electric clippers.	someone is using a tool to cut a piece of wood.
a person is attempting to mimic an angry dog.	a man is grunting and growling at a consistent rate.
a person breathing heavily and deeply while groaning.	a man is breathing heavily, then coughs again and again.
a large gathering of people are talking loudly with each other.	a crowd of people are talking in a crowded restaurant.

4.2. System Error Types and Probability

We also count the probability that the AAC system produces various types of errors in generating captions, including adding extra words at the end of a sentence, describing repeated events, repeating adverbs, missing conjunctions, missing verbs, and overall fluency errors, as shown in Table 3.

Table 3 Types and probability of errors obtained by system subtitles

Error Types	Probability
adding extra words: the man is running and breathing hard(...)	4.6%
repeated events: music plays by (music playing)	7.3%
repeating adverbs: sheep bleats nearby several times (nearby)	0.2%
missing conjunction: people speaking (and) a train horn blows	12.6%
missing verbs: food sizzles and a pan (verb)	22.3%
Overall fluency error	34.5%

5. CONCLUSIONS

This report introduced various techniques used in our system which is submitted to the Task 6A of DCASE 2024 challenge. We discuss effective methods for optimizing AAC models based on attentional mechanisms. Additionally, we demonstrate the effectiveness of our approach by achieving a FENSE score of 0.5131.

6. REFERENCES

[1] Y. Li, Q. Wang, X. Li, X. Zhang, Y. Zhang, A. Chen, Q. He, and Q. Huang, “Unsupervised detection of acoustic events using information bottleneck principle,” *Digital Signal Processing*, vol. 63, pp. 123-134, 2017.

[2] Z. Lin, Y. Li, Z. Huang, W. Zhang, Y. Tan, Y. Chen, and Q. He, “Domestic activities clustering from audio recordings using convolutional capsule autoencoder network,” in *Proc. of IEEE ICASSP*, 2021, pp. 835-839.

[3] Y. Li, X. Li, Y. Zhang, M. Liu, and W. Wang, “Anomalous sound detection using deep audio representation and a BLSTM network for audio surveillance of roads,” *IEEE Access*, vol. 6, pp. 58043-58055, 2018.

[4] Y. Li, M. Liu, K. Drossos, and T. Virtanen, “Sound event detection via dilated convolutional recurrent neural networks,” in *Proc. of IEEE ICASSP*, 2020, pp. 286-290.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[6] Park D S, Chan W, Zhang Y, et al. Specaugment: A simple data augmentation method for automatic speech recognition[J]. arXiv preprint arXiv:1904.08779, 2019.

[7] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.

[8] Gontier F, Serizel R, Cerisara C. Automated audio captioning by fine-tuning bart with audioset tags[C]//DCASE 2021-6th Workshop on Detection and Classification of Acoustic Scenes and Events. 2021.

[9] Etienne Labbé, Thomas Pellegrini, and Julien Pinquier. CoNeTTE: An efficient Audio Captioning system leveraging multiple datasets with Task Embedding. 2023.

[10] Y. Li, W. Cao, W. Xin Xie, Q. Huang, W. Pang, and Q. He, "Low-complexity acoustic scene classification using data augmentation and lightweight ResNet." in Proc. of IEEE International Conference on Signal Processing (ICSP), vol. 1, pp. 41-45, 2022.

[11] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.

[12] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[13] Zhang Q, Wang C, Wu H, et al. GELU-Net: A Globally Encrypted, Locally Unencrypted Deep Neural Network for Privacy-Preserved Learning[C]//IJCAI. 2018: 3933-3939.

[14] Xiao F, Guan J, Zhu Q, et al. Graph attention for automated audio captioning[J]. IEEE signal processing letters, 2023.

[15] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. In Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP), 736-740. 2020.