SEMANTIC ENHANCEMENT ENCODER FOR AUDIO CAPTIONING AND SPECTROGRAM-BASED DATA AUGMENTATION

Qianhang Feng[1], *Qiuqiang Kong*[1],

[1] The Chinese University of Hong Kong, Electronic Engineering Dept., Hong Kong, China

ABSTRACT

Automatic Audio Captioning (AAC) is a process that transforms audio signals into descriptive narratives. This paper introduces an innovative automated audio captioning model developed for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2024 Challenge Task 6A. The model architecture presented here is meticulously designed to adeptly manage the intricacies of AAC tasks. Additionally, this project introduces a novel data enhancement technique, which, with minimal model adjustments, significantly boosts performance. Exclusively trained and fine-tuned on the Clotho dataset, this project achieved a final SPIDEr-FL score of 0.3318, demonstrating its effectiveness.

Index Terms— Automatic Audio Captioning, Semantics, Data augmentation

1. INTRODUCTION

Automated Audio Captioning (AAC) is a technology that aims to convert the audio content into textual descriptions [1]. It is used to automatically generate captions or descriptions for audio clips. One of the most important applications of AAC is to provide an accessible environment for hearing impaired people or deaf-mute people. With AAC, audio content (e.g. ambient sounds, noise, music) can be converted into a textual description so that those who cannot hear audio elements can better understand and perceive their surroundings. This provides a more inclusive experience for everyone.

The complexity of AAC stems from the temporal dynamics and spectral variations inherent in audio data, necessitating advanced feature extraction methodologies. Early AAC systems employed Mel-frequency cepstral coefficients (MFCCs) [2] and Recurrent Neural Networks (RNNs) [3], yet they fell short in capturing the full semantic spectrum of audio. The current landscape of AAC is marked by end-to-end models that seamlessly integrate audio feature extraction with sequence generation mechanisms. These models adeptly capture both the semantic depth and temporal nuances of audio, culminating in captions that are not only accurate but also descriptively rich. In recent years, there has been a surge of innovations related to Audio-Aided Captioning (AAC). Wu et al innovatively leveraged CHATGPT to assist in generating training samples [4]. Komatsu et al employed audio difference learning to enable the model to better understand differences between audio clips [5]. Ghosh et al proposed Retrieval-Augmented Audio Captioning, which utilizes captions retrieved from a database similar to the input audio to enhance performance [6]. Deshmukh et al introduced an approach to train AAC systems solely using text [7]. Sridhar et al addressed the issues of hallucination and large memory footprint [8]. In terms of model architectures, Xiao et al. presented GraphAC, a graph attention module integrated into the encoder for feature representation [9]. Eren et al utilized bi-directional Gated Recurrent Units (BiGRU) to extract subjects and verbs from captions in the dataset to obtain semantic embeddings and improve model performance [10]. Moon et al combined LLaMAv2 to introduce AnyMAL - a multi-modal language model [11]. Moreover, Kim et al and Mei et al contributed a large-scale dataset AudioCaps [12] with 49K audio clips and a large-scale dataset WavCaps [13] with 403K audio clips, respectively.

In current mainstream audio feature extractors such as PANNs [14], EfficientAT [15], and Wav2Vec [16], they primarily obtain audio features solely from the audio signal. Since audio feature extractors tend to focus on capturing low-level acoustic characteristics like spectral content, temporal dynamics, and rhythm, these features may not directly encode the semantic meaning or context of the audio, limiting their adaptability to AAC task. In contrast, in the computer vision domain, the Q-Former in BLIP2 [17] aims to bridge the gap between a frozen visual model (e.g., ViT [18]) and a large language model, enabling the extraction of visual representations from images that are most relevant to text and ensuring that these representations can be interpreted by large language models. This provides valuable inspiration for our project. Regarding data augmentation techniques, while the application of Mixup [19] is straightforward, it may not yield significant benefits in certain datasets or tasks that already possess sufficient diversity. Additionally, when mixing two samples, an improper mixing ratio could result in generated samples containing a significant amount of non-informative elements that differ greatly from real samples, potentially negatively impacting model training. On the other hand, SpecAugment [20], which directly operates on spectrograms, may not be directly applicable to other types of audio features or representations.

In response to the issues and limitations raised above, the contributions of this project are as follows:

1. This project introduce the Semantic encoder. A model that are specifically designed to understand and generate semantic representations. It is a 6-layer self-attentional Transformer with an additional two-dimensional learnable matrix. For more accurate semantic representation, features are positional enhanced at each layer. After that, the output of the Audio Encoder and the Semantic encoder will be concatenated together at the last dimension. Finally, the new features are transformed by the projection layer and learned by the decoder.
2. This paper also introduces an innovative data augmentation approach that integrates a novel type of noise – spectrographic noise – to significantly enhance performance in AAC tasks. This technique not only accounts for the temporal and spectral characteristics of audio but also bolsters the language model's understanding of semantic content by combining spectrograms with the encoder's output. This integration allows the model to more accurately capture critical

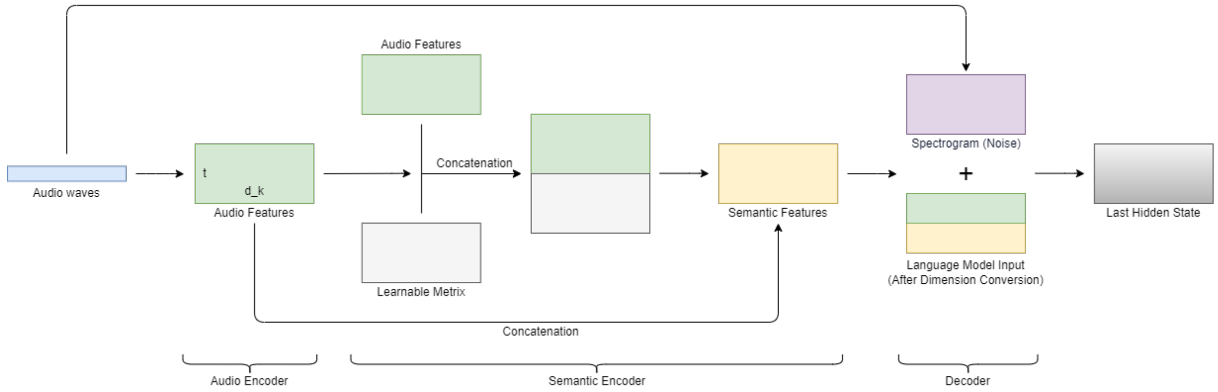

Figure 1: Model Flow

events and scene transitions within the audio, leading to the generation of more precise and descriptive captions.

This paper is organized as follows. In the first section, the system architecture of the model framework is presented, which is divided into three parts: audio encoder, semantic encoder, and decoder. The second part of the paper introduces the new approach to data augmentation. Section Three (Experiments) describes the experimental setup, datasets used, and evaluation metrics in detail. Finally, in the conclusion section, the main findings and contributions of this paper are summarized.

2. SYSTEM STRUCTURE

In this project, an frozen audio feature extractor is applied to filter out useful information, while a semantic enhancement encoder is used introduced for extracting semantic information from audio features. Finally, a language model is used to predict and optimize the probability distribution of word sequences.

$$h = AFE_{\theta_e}(x) \tag{1}$$

Where θ_e are the model parameters of the audio feature extractor (AFE). After getting audio feature h, the semantic enhancement encoder (SEE) extracts semantic information from the concatenation of h and the learnable matrix l at the last dimension.

$$s = SEE_{\theta_s}([h, l]) \tag{2}$$

Where θ_s are the model parameters of the semantic enhancement encoder (SEE). Finally, a language model (decoder) generates sentences with the combination of h and s.

$$p(w_t|[h, s], w_0, ..., w_{t-1}) = DEC_{\theta_d}([h, s], w_0, ..., w_{t-1}) \tag{3}$$

Where θ_d are the model parameters of the decoder (DEC). $[h, s]$ is the concatenation of h and the semantic matrix s.

Training steps of the whole model can be divided as two parts. First step is to training the semantic enhancement encoder without a deocder. This step only facilitates it to analyze and understand the semantic information in the audio. Then both the audio feature extractor and the semantic enhancement encoder are frozen to train the decoder. This step is to make the output sentences more fluent and stable.

2.1. Audio feature extractor

In this project, the computer vision architecture ConvNeXt (Tiny) [21] trained on AudioSet and finetuned on AudioCaps is adopted to perform audio feature extraction. ConvNeXt [22] is a series of pure convolutional neural network models that, through modernized design improvements, demonstrate accuracy and scalability comparable to Transformer models. ConvNeXt (Tiny) is a compact variant of the ConvNeXt architecture designed to offer a balance between performance and computational efficiency. It features a streamlined version of the hierarchical vision transformer structure, with a reduced number of layers and channels compared to larger ConvNeXt models. The Tiny model maintains the core design principles, such as depth-wise separable convolutions for feature extraction and cross-attention mechanisms that allow for efficient modeling of relationships between different parts of the input data. Despite its smaller size, ConvNeXt (Tiny) retains the ability to capture complex patterns and produce high-quality feature representations, making it suitable for applications where computational resources are limited but high accuracy is still required.

2.2. Semantic Enhancement Encoder

The semantic enhancement encoder plays a crucial role in combining the outputs of the Audio Encoder with additional learnable features, resulting in a matrix that captures both the semantic and temporal aspects of the audio. This component is essential for representing the semantic information of the audio, which is not fully captured by the Audio Encoder alone. The Semantic encoder is based on a self-attention Transformer [23] but incorporates one significant improvement: the addition of learnable embeddings for each batch. This learnable matrix is specifically crafted to abstract and capture the salient features of ambient noise, aiming to facilitate the encoder in deriving audio representations that are inherently pertinent to the textual context. Additionally, this representation is structured in a manner that renders it comprehensible and interpretable by large-scale language models, thus enhancing the overall intelligibility and explainability of the system.

This design is inspired by Querying Former in BLIP-2. But different from Querying Former, Querying Former uses the same query set for the extraction of visual features, which Bridges the information bottleneck between feature extractor and LLM. In this project, the audio features are transformed into the feature space with the help of the learnable matrix, which makes up for the difference between the audio space and the text space, so that the model can learn the audio representation most relevant to the text.

Additionally, inspiring by the work from P-Transformer [24], the hidden state of Semantic encoder is added with position embedding at each layer. This approach aims to solve the problem of position information weakening or vanishing as it reaches the bottom layers of the encoder. By enhancing the location information, the model is able to better understand the structure and order of the source audio, thus generating more grammatical and semantic text features.

$$Attention = softmax(\frac{(Q' + P)(K' + P)^T}{\sqrt{d_k}})(V') \tag{4}$$

Where $P \in R^{1\times d_k}$ is the learnable absolute position embeddings. Q' is the concatenation of $[Q, l]$, which means the concatenation of the query $Q \in R^{b\times t\times d_k}$ and the learnable matrix $l \in R^{1\times t\times d_k}$ along the last dimension. b represents the batch size, t is the sequence length of audio features, d_K is the hidden size. Similarly, K' is the concatenation of $[K, l]$, and V' is the concatenation of $[V, l]$. K and V represent the key and value in the attention mechanism, respectively.

Before learning semantic features, the model needs to interpolate the audio features into a specific shape in order to concatenate it with the learnable matrix. In this project, a shape of (94, 768) is chosen for both audio features and the learnable matrix, where 94 is the sequence length and 768 is the hidden size, as it is much closer to the average shape of audio encoder outputs. The semantic enhancement encoder is then trained with the help of a strong and powerful word embedding model - BGE [25]. By utilizing infoNCE [26] loss, the semantic enhancement encoder can learn as much excellent embedding as possible form BGE.

The InfoNCE loss for a single instance can be formulated as follows:

$$L = -\sum_{i=1}^{N} \log(\frac{exp(\frac{q_i \cdot y_{i+}}{\tau})}{\sum_{j=1}^{N} exp(\frac{q_i \cdot y_{j-}}{\tau})}) \tag{5}$$

where N is the batch size, q (query) refers to the outputs of semantic enhancement encoder, y_{i+} is the positive example (target) associated with the $i - th$ input data, and y_{j-} the representations of all the negative examples, τ is the temperature parameter. For Clotho [27] dataset an audio has five corresponding captions, so the loss function can be written as:

$$L_t = \frac{1}{T} \sum_{i=1}^{T} L(q, y_{i+}, y_-) \tag{6}$$

where T is 5 in Clotho dataset.

Also, the output of Semantic encoder are gather with corresponding captions in cross entropy loss as below. This helps the Semantic encoder represent the semantics of the audio itself as much as possible.

$$L_{ce} = -\sum_{c=1}^{M} y log(p) \tag{7}$$

where M is the total number of classes (tokens), y is a binary indicator (0 or 1) of whether the class is the correct classification for observation. p is the predicted probability that observation is of the class.

As a result, the total loss is a simple combination of these two losses.

$$L_{final} = L_t + L_{ce} \tag{8}$$

2.3. Decoder

The Decoder (language model) is a simple 6 layers GPT [28]. This project has conducted a comparative analysis and concluded that the significance of the language model is somewhat diminished in comparison to the Audio Encoder within the given framework. It is suggested that the framework design should commence with a modest number of parameters, which can then be incrementally scaled to an optimal range. The project has also evaluated and ranked the impact of several pivotal elements within a transformer-based decoder architecture.

The first and foremost influential factor is the decoder's hidden size. The magnitude of the hidden size is directly proportional to the model's complexity. An increased hidden size equates to a larger number of parameters, which in turn raises the model's complexity. This allows the model to capture more intricate patterns and relationships. However, it also demands more computational resources and extends the training duration. Within an optimal range, augmenting the hidden size can enhance the model's performance, as it enables the model to learn a more nuanced representation that better encapsulates the input sequence's information. Nonetheless, the performance gain is not linear, and the improvement may plateau or even diminish as the hidden size continues to grow. In this project, simply doubling the baseline decoder's hidden size from 256 to 512 resulted in a 3.5% performance increase, and further scaling to 768 yielded an additional 2% improvement. However, extending the model to a hidden size of 1024 began to exhibit diminishing returns. This is attributed to the model's increased sensitivity to the specific nuances of the training data, which can impede its ability to generalize to new, unseen data.

The second critical element is the number of attention heads within the multi-head attention mechanism. This parameter dictates the number of parallel attention distributions the model utilizes when processing the input sequence. An insufficient number of heads may fail to adequately capture the diverse features and patterns present in the input. Given that AAC task and evaluation metrics often emphasize the diversity and originality of text, it is advisable to moderately increase the number of heads. However, an excessive number of heads could escalate computational complexity and potentially lead to overfitting, a risk that is particularly pronounced with limited datasets.

Lastly, the dictionary size of the decoder is another determinant of the model's capability. A larger dictionary enables the decoder to generate a wider array of outputs, enhancing the diversity and specificity of the generated text. Nevertheless, a vast dictionary with many infrequent words may result in a sparse output distribution, which could lead to the model assigning disproportionately low probabilities to some words and excessively high probabilities to others during the decoding phase. This, in turn, could adversely affect the fluency and readability of the generated text.

Additionally, experiments conducted with a pre-trained BART [29] decoder revealed that the performance of a pre-trained language model may not surpass that of a model trained from scratch. This suggests that the new task at hand is significantly divergent from the pre-trained task, and the pre-trained model may struggle to adapt effectively, leading to challenges in knowledge fusion.

3. NEW METHOD FOR DATA AUGMENTATION

Apart from the model presented, this project also finds a novel way to increasing the performance of the model. This project introduces a novel noise - spectrogram. By adding the corresponding specgram with the output of encoders, the model can reach an upper level.

The spectrogram provides valuable information about the frequency distribution of the signal, which can be more abundant than the time domain information in the original audio signal. Incorporating this information into the model as noise can enhance its ability to learn and differentiate various audio features. Additionally, combining spectrogram and audio features facilitates feature fusion, enabling the model to comprehend audio data from diverse perspectives. This multi-spacial feature fusion contributes to an improved understanding of audio content by the model. Furthermore, this representation closely aligns with how sound is perceived by the human auditory system. Therefore, utilizing spectrograms as a source of noise can simulate complex sound environments that more closely resemble real-world scenarios encountered by human hearing, thereby enhancing the model's robustness in practical applications.

In addition, the spectrogram is also used as a residual [30] to compensate for the information ignored by the encoder. Sound events, such as speech and music, usually contain more complex information, such as semantic functions, different sounds represent different meanings, and can also convey emotions and express intentions. Although environmental noise also contains the physical attributes of sound, compared with speech and music, its information content may be relatively simple, mainly describing the state or change of the environment, such as traffic noise, machine running sound, etc.

The depth of the encoder network poses a significant challenge; an overly deep architecture can lead to a desensitization to subtle differences among sound categories, resulting in a homogenization of stimuli. This, in turn, can diminish the effectiveness of the model's decoding capabilities, as a reduction in mean square error does not necessarily correlate with improved perceptual quality. To counteract this, the project introduces a novel approach that leverages the inherent discrepancies in the stimuli—specifically, the original audio signals—to compensate for information loss due to neural network filtering and compression. This strategy is designed to guide the decoder along the desired convergence trajectory, thereby enhancing the fidelity of the output.

However, the integration of spectrograms as an intrinsic component of the model, particularly during the generative phase, has not yielded the anticipated results in experiments, sometimes even introducing deleterious effects. This project hypothesizes that the raw audio data, laden with noise and interference, may introduce additional errors and uncertainties, potentially distorting the model's performance during evaluation.

Empirically, this project has identified several viable methodologies. Preliminary experiments suggest that the efficacy of these methods is context-dependent, and there is no one-size-fits-all solution. Due to time constraints, the project has not conducted extensive comparative experiments to further validate and categorize these approaches. Nevertheless, based on current findings, following conclusions can be drawn:

1. Firstly, the spectrogram of an audio signal encompasses two primary dimensions: the frequency domain (F) and the time domain (T), denoted as F-T. And the feature representation within a model is typically conceptualized as time (T) and feature dimensions (H), denoted as T-H.

   When considering the integration of these two tensors, two approaches can be particularly effective. One approach involves directly adding the tensors in a residual fashion, where the F-T and T-H dimensions are combined. The other approach involves transposing the frequency dimension of the spectrogram to align it with the feature dimension, thus achieving a T-F to T-H superposition. Both of these methods, or a hybrid combination of the two, can yield promising results and are recommended to be flexibly employed during training.

2. The optimal timing for integrating the spectrogram into the model requires further investigation. The project explored two primary strategies: pre-superposing before the signal passes through the decoder's projection layer, and post-superposing after the fully connected layer. The former introduces a projection layer and a RELU activation function to selectively enhance information, while the latter ensures the integrity of the spectral data, and since the model has done most of the feature extraction and abstraction at this point, it provides more flexibility. Both methods have their merits and are effective, though their performance may vary across different models.

Given that a spectrogram serves as a visual portrayal of how a signal's frequency varies across time, and considering the past successes in applying visual models to the realm of acoustics (AST [31], ACT-DeiT [32]), this project postulates that the same methodology holds promise for application in traditional visual models. Nonetheless, the lack of comprehensive experimental data necessitates further exploration and validation of this hypothesis.

4. EXPERIMENTS

In this project, three datasets are used for training and testing. In order to verify the feasibility of the theory, 4 models were exhibited for comparison. Model_Full is the full model trained with additional AudioCaps and WavCaps dataset. Model_Clotho is the full model trained only on Clotho v2 dataset. Model_Ref is the model trained only on Clotho v2 dataset but without the sepcs-noise data augmentation. Model_Simple is the model trained without the semantic encoder and the sepcs-noise data augmentation on Clotho v2 dataset. The results are shown as below:

Model ID	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL
Model_Full	0.1959	0.5291	0.1382	0.3337	0.3318
Model_Clotho	0.1925	0.4949	0.1396	0.3173	0.3142
Model_Ref	0.1880	0.4808	0.1334	0.3069	0.3054
Model_Simple	0.1783	0.4471	0.1259	0.2865	0.2781

Table 1: Results of models with different structures and data enhancement strategies on Clotho Development-Test subset.

The salient feature of this project is its robust performance, even when trained on a limited dataset. It is worth noting that the model can still ensure convergence when the batch size is small, achieving good performance.

When utilizing only the Clotho dataset and assuming a single GPU training environment equipped with a GTX 4090, the expected duration of the training process is about 200 minutes. In this time frame, the semantic encoder is expected to achieve convergence at the 70th iteration, while the decoder needs 30 iterations to achieve the best performance. The training scheme employs an AdamW optimizer that utilizes a learning rate of 3e-5, which can be sensibly reduced to 1e-5 to potentially improve the effectiveness of the model. At a batch size of 64, the training of the model requires a large memory footprint of 20,000 MiB.

In addition, the project observed a correlation between metrics such as SPIDEr [33] score and the number of GPUs in use when deploying the model for evaluation across multiple GPUs, although the output of the model did not change. This phenomenon suggests that the scoring process may induce some degree of variation even when there is no data overlap between GPUs. Specifically, the estimated observation error is +3.6% when using 5 GPUs to ensure non-overlapping processing of 1045 test samples. In light of these findings, the project advocates the use of a single GPU during testing to obtain a more accurate and unbiased assessment of the model's capabilities.

5. CONCLUSION

The model presented in this paper demonstrates a robust approach to Automatic Audio Captioning, leveraging a combination of advanced model architecture and innovative training strategies. The results indicate that this project not only captures the audio's semantic and temporal features but also enhances the model's ability to generalize and perform well on unseen data.

6. REFERENCES

[1] M. Xinhao, L. Xubo, M. D. Plumbley and W. Wenwu, “Automated Audio Captioning: An Overview of Recent Progress and New Challenges,” arXiv preprint arXiv: 2205.05949, 2022.

[2] S. Davis and P. Mermelstein. ”Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” IEEE Transactions on Acoustics, Speech and Signal Processing, 1980, 28, 357-366.

[3] J.L. Elman, ”Finding structure in time,” Cogn Sci, 14 (2), 1990, pp. 179-211.

[4] W. Shih-Lun, C. Xuankai, W. Gordon, et al, ”BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up,” DCASE2023 Challenge, 2023.

[5] K. Tatsuya, F. Yusuke, T. Kazuya and T. Tomoki, ”Audio Difference Learning for Audio Captioning,” arXiv preprint arXiv:2309.08141, 2023.

[6] G. Sreyan, K. Sonal, et al., ”RECAP: Retrieval-Augmented Audio Captioning” arXiv preprint arXiv:2309.09836, 2024.

[7] D. Soham, E. Benjamin, E. Dimitra, et al., ”Training Audio Captioning Models without Audio,” Microsoft Research, 2024.

[8] S.A. Krishna, G. Yinyi, V. Erik and M. Rehana, ”Parameter Efficient Audio Captioning With Faithful Guidance Using Audio-text Shared Latent Representation” arXiv preprint arXiv:2309.03340, 2023.

[9] X. Feiyang, G. Jian, Z. Qiaoxi and W. Wenwu, ”Graph Attention for Automated Audio Captioning,” arXiv preprint arXiv:2304.03586, 2023.

[10] E. A. Özkaya, S. Mustafa, ”Audio Captioning with Composition of Acoustic and Semantic Information” arXiv preprint arXiv:2105.06355, 2021.

[11] M. Seungwhan, M. Andrea, L. Zhaojiang, et al., ”AnyMAL: An Efficient and Scalable Any-Modality Augmented Language Model” arXiv preprint arXiv:2309.16058, 2023.

[12] C. D. Kim, B. C. Kim, H. M. Lee, and G. H. Kim, ”Audiocaps: Generating captions for audios in the wild,” in Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 119–132.

[13] M. Xinhao, M. Chutong, and L. Haohe, “WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research,” arXiv preprint arXiv:2303.17395, 2023.

[14] K. Qiuqiang, C. Yin, I. Turab, et al., ”PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition” arXiv preprint arXiv:1912.10211, 2020.

[15] S. Florian, K. Khaled, W. Gerhard, ”Efficient Large-Scale Audio Tagging Via Transformer-To-CNN Knowledge Distillation,” ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5.

[16] B. Alexei, Z. Henry, M. Abdelrahman and A. Michael, ”wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations” arXiv preprint arXiv:2006.11477, 2020.

[17] L. Junnan, L. Dongxu, S. Silvio and H. Steven, ”BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” arXiv preprint arXiv:2301.12597, 2023.

[18] D. Alexey, B. Lucas, K. Alexander, et al., ”An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” arXiv preprint arXiv:2010.11929, 2021.

[19] Z. Hongyi, C. Moustapha, D. N. Yann and L. David, ”mixup: Beyond Empirical Risk Minimization” arXiv preprint arXiv:1710.09412, 2017.

[20] D. S. Park, C. William, Z. Yu, et al., ”SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.” arXiv preprint arXiv: 1904.08779, 2019.

[21] P Thomas, Khalfaoui-Hassani I, Labbé E, et al., “Adapting a ConvNeXt model to audio classification on AudioSet,” arXiv preprint arXiv: 2306.00830, 2023.

[22] Z. Liu, H. Mao, C.-Y. Wu, et al., “A convnet for the 2020s,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11 976–11 986.

[23] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” In Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017.

[24] Y. Li, J. Li, Jing. J, et al., “P-Transformer: Towards Better Document-to-Document Neural Machine Translation,” arXiv preprint arXiv:2212.05830, 2022.

[25] J. Chen, S. Xiao, P. Zhang, ”BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation,” arXiv preprint arXiv:2402.03216, 2024.

[26] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” arXiv preprint arXiv:1807.03748, 2018.

[27] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: an audio captioning dataset,” in Proc. ICASSP, 2020, pp. 736–740.

[28] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever. ”Improving language understanding by generative pre-training,” 2018.

[29] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in Proc. ACL, 2020.

[30] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, ”Deep Residual Learning for Image Recognition,” arXiv preprint arXiv:1512.03385, 2015.

[31] G. Yuan, C. Yu-An, and G. James, “Ast: Audio spectrogram transformer,” in Interspeech 2021.

[32] M. Xinhao, L. Xubo, M. D. Plumbley and W. Wenwu, ”Audio Captioning Transformer.” arXiv preprint arXiv:2107.09817, 2021.

[33] Z. Zelin, Z. Zhiling, X. Xuenan, et al., ”Can Audio Captions Be Evaluated with Image Caption Metrics?” arXiv preprint arXiv: 2110.04684, 2022.