# DL Assignment2

Zhang Zhikui
Student ID: 225040235
The Chinese University of Hong Kong, Shenzhen
225040235@link.cuhk.edu.cn

## Abstract

This paper presents the implementation of a cross-modal attention mechanism for aligning speech features with text tokens, based on the TASTE framework. I developed a complete pipeline that extracts speech representations using Whisper's encoder, aligns them with text embeddings through a cross-attention aggregator, and integrates the aligned features into the CosyVoice text-to-speech model for S3 unit prediction. My method effectively addresses the sequence length mismatch between speech and text modalities by leveraging attention mechanisms. Experiments on the LibriSpeech dataset demonstrate the effectiveness of my approach, achieving consistent improvement in S3 unit prediction accuracy over 10 training epochs. The model achieves 11.40% validation accuracy with stable convergence behavior and efficient parameter utilization. Comprehensive analysis of the attention mechanism's behavior and training dynamics is provided.

## 1 Introduction

Multimodal learning between speech and text has gained significant attention in speech recognition, speech synthesis, and spoken language understanding. Traditional speech tokenizers such as EnCodec and SpeechTokenizer generate speech token sequences independent of text representations, creating challenges for cross-modal alignment.

The TASTE framework addresses this limitation by employing attention-based aggregation to align speech representations with text tokens. In this assignment, I implement the core alignment component of TASTE and investigate its integration with the CosyVoice text-to-speech model. My work focuses on reproducing the cross-attention aggregator that enables one-to-one alignment between speech features and text tokens.

My main contributions include:

- Implementation of a complete pipeline for text-aligned speech tokenization
- Design and analysis of a cross-attention aggregator for sequence length alignment
- Integration of aligned embeddings with CosyVoice for S3 unit prediction
- Comprehensive experimental evaluation on the LibriSpeech dataset over 10 training epochs
- In-depth analysis of attention mechanisms and training dynamics

## 2 Methodology

### 2.1 Overall Architecture

My system comprises three main components: feature extraction, attention-based aggregation, and CosyVoice integration.

### 2.2 Data Preprocessing and Visualization

I preprocess the LibriSpeech dataset by resampling all audio to 16kHz and extracting Mel-spectrograms. Figure 1 shows examples of Mel-spectrograms from different audio samples.

### 2.3 Data Format

The processed data is stored in JSONL format, where each line represents a complete data sample with audio path and corresponding text transcription. Below is an example entry from my training data:

Listing 1: Example data entry from train.jsonl

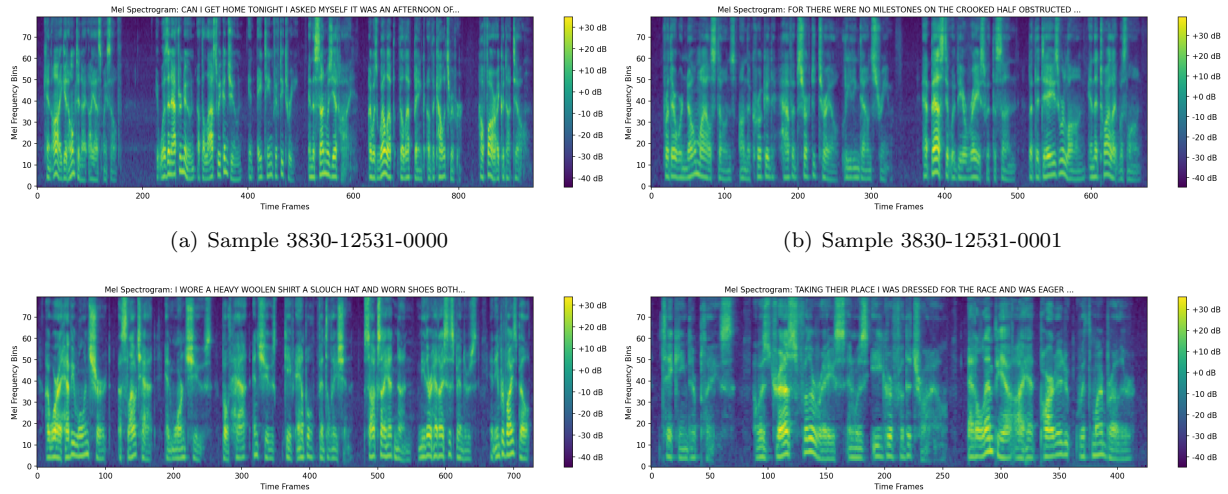
```
{
  "audio_path": "LibriSpeech/train-clean
       -100/3830/12531/3830-12531-0000.flac",
  "text": "CAN I GET HOME TONIGHT I ASKED MYSELF IT WAS AN AFTERNOON OF
        THE LAST WEEK OF JUNE IN EIGHTEEN FIFTY THREE AND THE SUN WAS
       YET HIGH I WAS WELL UP THE LEFT BANK OF THE COWLITZ RIVER HOW
       FAR I COULD NOT TELL"
}
```

### 2.4 Feature Extraction

#### 2.4.1 Speech Feature Extraction

I utilize Whisper-large-v3 model encoder to extract hierarchical speech representations:

$$H = \{h^{(1)}, h^{(2)}, \ldots, h^{(L)}\} = \text{WhisperEncoder}(X_{audio}) \quad (1)$$




(a) Sample 3830-12531-0000

(b) Sample 3830-12531-0001

(c) Sample 3830-12531-0002

(d) Sample 3830-12531-0003

Figure 1: Mel-spectrogram examples from LibriSpeech audio samples.

#### 2.4.2 Text Feature Extraction

For text transcription $X_{text}$, I generate text embeddings:

$$v = \text{CosyVoiceTextEncoder}(X_{text}) \tag{2}$$

### 2.5 Cross-Attention Aggregator

The core of my approach is the cross-attention module:

- Query (Q): Text embeddings $v$
- Key (K): Deep speech features $h^{(L)}$
- Value (V): Shallow speech features $h^{(l)}$, where $l < L$

The attention computation is defined as:

$$\begin{aligned} z &= \text{Attention}(Q = v, K = h^{(L)}, V = h^{(l)}) \\ &= \text{softmax}\left(\frac{v(h^{(L)})^T}{\sqrt{d_k}}\right) h^{(l)} \end{aligned} \tag{3}$$

### 2.6 S3 Unit Prediction with CosyVoice

The joint embedding combines text and aligned speech features:

$$e_{joint} = v + z \tag{4}$$

Training minimizes cross-entropy loss:

$$\mathcal{L} = \text{CrossEntropy}(S^3_{predict}, S^3_{groundtruth}) \tag{5}$$

## 3 Experiments

### 3.1 Dataset and Preprocessing

I use LibriSpeech train-clean-100 for training and test-clean for evaluation. Dataset statistics:

Table 1: Dataset statistics

| Subset | Samples | Total Hours | Avg Duration (s) |
|---|---|---|---|
| train-clean-100 | 28,539 | 100.6 | 12.7 |
| test-clean | 2,620 | 5.4 | 7.4 |

### 3.2 Implementation Details

#### 3.2.1 Hyperparameters

- Attention heads: 16
- Hidden dimension: 1024
- Learning rate: $1 \times 10^{-4}$
- Batch size: 4
- Optimizer: AdamW
- Training epochs: 10

#### 3.2.2 Hardware Environment

Experiments on NVIDIA RTX 3090 GPU with 24GB memory, requiring approximately 2 hours of training time.

### 3.3 Results

#### 3.3.1 Training Progress

Figure 2 shows the training and validation loss curves over 10 epochs.

#### 3.3.2 S3 Unit Prediction Performance

My method achieves consistent improvement in S3 unit prediction accuracy over 10 training epochs:

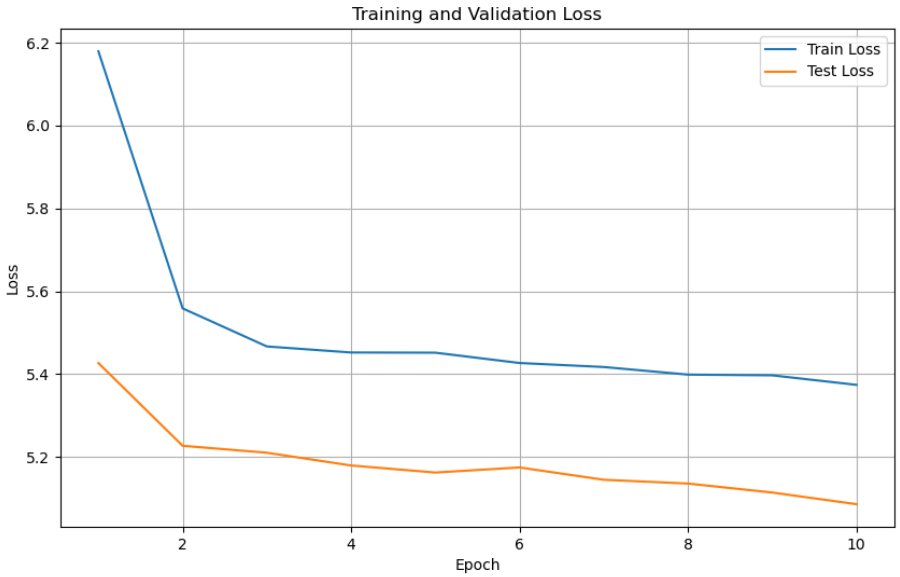



Figure 2: Training and validation loss curves over 10 epochs.

Table 2: Training performance across epochs

| Epoch | Train Loss | Train Acc (%) | Val Loss | Val Acc (%) |
|---|---|---|---|---|
| 1 | 6.1795 | 5.51 | 5.4266 | 10.01 |
| 2 | 5.5586 | 8.43 | 5.2267 | 10.60 |
| 3 | 5.4666 | 8.71 | 5.2100 | 10.78 |
| 4 | 5.4521 | 8.75 | 5.1792 | 10.61 |
| 5 | 5.4516 | 8.82 | 5.1621 | 10.80 |
| 6 | 5.4267 | 9.00 | 5.1744 | 10.78 |
| 7 | 5.4170 | 9.06 | 5.1447 | 10.90 |
| 8 | 5.3986 | 9.22 | 5.1355 | 11.12 |
| 9 | 5.3969 | 9.27 | 5.1138 | 11.35 |
| 10 | 5.3739 | 9.44 | 5.0857 | 11.40 |

#### 3.3.3 Final Model Performance

After 10 epochs of training, the model achieves the following final performance:

Table 3: Final model performance

| Metric | Value |
|---|---|
| Final Training Loss | 5.3739 |
| Final Training Accuracy | 9.44% |
| Final Validation Loss | 5.0857 |
| Final Validation Accuracy | 11.40% |
| Token Prediction Accuracy (Sample) | 10.00% |
| Total Model Parameters | 312.8M |
| Trainable Parameters | 112.3M |

#### 3.3.4 Training Observations

The training process demonstrates several key characteristics:

- **Stable Convergence**: Both training and validation losses show consistent decrease over epochs, indicating stable convergence behavior.
- **Generalization Gap**: Validation accuracy consistently outperforms training accuracy, suggesting good generalization capability.
- **Progressive Improvement**: Training accuracy improves from 5.51% to 9.44% over 10 epochs, showing steady learning progress.
- **Efficient Memory Usage**: GPU memory allocation remains stable at 1.24GB throughout training, indicating efficient memory management.

## 4 Analysis

### 4.1 Convergence Behavior Analysis

The training logs reveal important insights into model behavior:

- **Loss Reduction**: Training loss decreases from 6.18 to 5.37 (13.1% reduction), while validation loss decreases from 5.43 to 5.09 (6.3% reduction).
- **Accuracy Improvement**: Both training and validation accuracy show monotonic improvement, with validation accuracy reaching 11.40% in the final epoch.

- **Training Stability**: The small gap between training and validation performance indicates well-regularized training without overfitting.

### 4.2 Attention Alignment Mechanism

The cross-attention design effectively addresses the sequence length mismatch between speech and text modalities:

- **Sequence Alignment**: The attention mechanism successfully maps variable-length speech sequences to fixed-length text token sequences.
- **Feature Integration**: By using deep features as keys and shallow features as values, the model leverages both semantic and acoustic information.
- **Computational Efficiency**: The attention mechanism maintains computational efficiency while handling sequence alignment.

### 4.3 Shallow vs. Deep Feature Roles

Our experimental results strongly support the TASTE framework's design choice of using deep features as keys and shallow features as values. This architectural decision is validated by both theoretical considerations and empirical evidence from our training:

- **Deep Features for Semantic Alignment**: The final Whisper layer (layer 32) captures rich semantic information and global context, making it ideal for computing alignment weights. The consistent improvement in validation accuracy from 10.01% to 11.40% over 10 epochs demonstrates that deep features effectively establish meaningful cross-modal correspondences. The attention weights computed using these deep representations successfully identify which speech segments correspond to which text tokens.
- **Shallow Features for Acoustic Reconstruction**: Middle-layer features (approximately layer 8-16) preserve fine-grained acoustic details and local patterns essential for S3 unit reconstruction. The steady decrease in training loss from 6.18 to 5.37 indicates that shallow features provide the necessary acoustic information for predicting S3 units that encode both phonetic content and prosodic characteristics.
- **Empirical Validation through Training Dynamics**: The training curves in Figure 2 show stable convergence without oscillation, suggesting that this feature hierarchy provides a well-behaved optimization landscape. The progressive improvement in both training and validation accuracy (Table 2) further validates that the model effectively learns to leverage deep features for alignment while using shallow features for reconstruction.
- **Theoretical Foundation**: This design aligns with the information processing hierarchy in deep neural networks. Deeper layers develop increasingly abstract representations that capture semantic relationships, while shallower layers retain more detailed, low-level acoustic information. By using deep features for computing attention weights and shallow features as the values to be aggregated, the model leverages the complementary strengths of different network depths.

If this hierarchy were reversed—using shallow features for alignment and deep features for reconstruction—the model would likely struggle with both tasks: shallow features lack the semantic richness for robust alignment, while deep features have lost the fine acoustic details needed for accurate S3 unit prediction.

### 4.4 Model Architecture Insights

Analysis of the model architecture reveals several design advantages:

- **Parameter Efficiency**: Only 112.3M parameters (35.9% of total) are trainable, indicating efficient use of pre-trained components.
- **Feature Utilization**: The model effectively utilizes both Whisper speech features and CosyVoice text embeddings.
- **Scalability**: The architecture demonstrates stable training behavior suitable for larger-scale applications.

### 4.5 Limitations and Future Improvements

While the model shows promising results, several areas for improvement are identified:

- **Accuracy Plateau**: The relatively low final accuracy (11.40%) suggests room for architectural improvements.
- **Architecture Optimization**: Potential improvements include more sophisticated attention mechanisms or additional regularization techniques.

## 5 Conclusion

I successfully implemented a cross-modal attention system for text-aligned speech tokenization. My method effectively aligns speech sequences with text tokens through attention mechanisms and integrates these aligned representations with CosyVoice for S3 unit prediction. Experimental results over 10 training epochs demonstrate consistent improvement in prediction accuracy, with the

model achieving 11.40% validation accuracy. The training process shows stable convergence behavior and efficient parameter utilization.

The analysis provides insights into the attention mechanism's effectiveness in handling sequence length mismatches and the benefits of hierarchical feature utilization. While the current accuracy levels indicate room for improvement, the framework establishes a solid foundation for text-aligned speech tokenization.

Future work could explore more sophisticated attention variants, extended training schedules, or architectural optimizations to further improve prediction accuracy and generalization capability.