

# 实验二：基于HMM的语音识别与强制对齐实验

## 基本任务：查找阅读文献资料并回答问题

**1.1 请查找并阅读至少 3 篇有关 HMM 模型的参考文献资料，其中：至少 1 篇为经过同行评审的正式发表的英文论文（会议或期刊均可）；不得多于 1 篇为知乎、博客、百科等非正式发表的科普性阅读资料或讲义。（10 分）**

解：查找并阅读的3篇参考文献资料所在网址如下：

- (1) <https://ieeexplore.ieee.org/document/18626>
- (2) <https://www.jmlr.org/papers/v18/16-093.html>
- (3) <https://ieeexplore.ieee.org/abstract/document/9746686>

**1.2 以标准参考文献的格式列出所阅读的参考文献列表，并在回答以下问题时给出明确的引用。（5 分）**

解：所阅读的参考文献列表如下：

- [1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.
- [2] F. Yang, S. Balakrishnan, and M. J. Wainwright, "Statistical and computational guarantees for the baum-welch algorithm," Journal of Machine Learning Research, vol. 18, no. 125, pp. 1–53, 2017.
- [3] S. Mehta, E. Szekely, J. Beskow, and G. E. Henter, "Neural hmms are all you need (for high-quality attention-free tts)," in ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7457–7461, 2021.

**1.3 什么是马尔可夫过程？一阶马尔可夫过程的特点是什么？（可结合图）对一阶马尔可夫过程及一阶马尔可夫系统进行明确的形式化定义。（3 分）**

解：

(1)

马尔可夫过程<sup>[1]</sup>是数学中的一个随机过程概念，描述了一个系统如何在不同状态之间转移。其核心特性是无后效性，即在某一时刻的系统状态，只取决于系统当前的状态，而与系统过去的状态无关。这意味着系统的未来演变只依赖于当前状态，而不依赖于到达该状态的历史路径。

(2)

一阶马尔可夫过程的特点是系统的未来状态只依赖于当前状态，而与之前的状态无关。换句话说，它满足所谓的马尔可夫性，即当前状态是未来状态的充分条件，而过去状态对未来状态的影响完全通过当前

状态体现。

具体特点如下：

1. **无后效性**：在任何时刻，未来状态的概率分布只依赖于当前状态，而不依赖于系统的历史状态。这种特性是马尔可夫过程最显著的特点。

$$P(X_{t+1}|X_t, X_{t-1}, \dots, X_0) = P(X_{t+1}|X_t)$$

其中  $X_t$  表示系统在时刻  $t$  的状态， $P$  是条件概率。

2. **状态转移概率**：系统从一个状态转移到另一个状态是通过一个概率分布来描述的，这个分布在一阶马尔可夫过程中只取决于当前状态。每个状态都有可能转移到另一个状态，且不同状态之间的转移可能具有不同的概率。
3. **记忆限制**：由于未来状态仅依赖于当前状态，系统的“记忆”仅限于当前的状态，这大大简化了对系统演变分析和计算。
4. **简化性**：一阶马尔可夫过程是最简单的马尔可夫过程形式，因此在许多应用中得到广泛使用，例如隐马尔可夫模型（HMM）和随机游走等。

(3)

一阶马尔可夫系统是描述一阶马尔可夫过程的完整结构。它由以下几部分组成：

1. **状态空间  $S$** ：这是系统可能取值的所有状态的集合。状态空间可以是有限集，也可以是可数集，记为：

$$S = \{s_1, s_2, \dots, s_n\}$$

其中， $s_i$  代表第  $i$  个状态。

2. **转移概率矩阵  $P$** ：这是描述系统从一个状态转移到另一个状态的概率矩阵。对于所有状态  $s_i, s_j \in S$ ，转移概率  $P_{ij}$  定义为在时刻  $t$  系统从状态  $s_i$  转移到状态  $s_j$  的概率：

$$P_{ij} = P(X_{t+1} = s_j | X_t = s_i)$$

3. **初始状态分布  $\pi_0$** ：这是系统在时间  $t = 0$  时的状态概率分布，即描述系统最初时刻处于每个状态的概率。对于每个状态  $s_i \in S$ ，初始状态概率定义为：

$$\pi_0(s_i) = P(X_0 = s_i), \quad \sum_{i=1}^n \pi_0(s_i) = 1$$

这表示在  $t = 0$  时，系统处于每个状态  $s_i$  的概率，总和为 1。

综上，一个一阶马尔可夫系统  $M$  可以表示为三元组：

$$M = (S, P, \pi_0)$$

其中：

- $S$  是状态空间，表示系统可能的所有状态；
- $P$  是转移概率矩阵，描述系统从一个状态转移到另一个状态的概率；
- $\pi_0$  是初始状态分布，表示系统在初始时刻的状态概率。

## 1.4 什么是隐马尔可夫模型？（可结合图）对隐马尔可夫模型进行明确的形式化定义。（3 分）

解：

(1)

隐马尔可夫模型是一种用来描述一个含有隐含未知参数的马尔可夫过程的统计模型，其难点是从可观察的参数中确定该过程的隐含参数，然后利用这些参数来作进一步的分析。

在正常的马尔可夫模型中，状态对于观察者来说是直接可见的。这样状态的转换概率便是全部的参数。而在隐马尔可夫模型中，状态并不是直接可见的，但受状态影响的某些变量则是可见的。每一个状态在可能输出的符号上都有一概率分布。因此输出符号的序列能够透露出状态序列的一些信息。

(2)

一个隐马尔可夫模型  $\lambda$  由以下五个要素组成：

### 1. 隐含状态集合 $S$ ：

隐含状态的有限集合，表示系统可能处于的状态：

$$S = \{s_1, s_2, \dots, s_N\}$$

其中  $N$  是隐含状态的数量。

### 2. 观测集合 $O$ ：

观测结果的有限集合，表示可观测的输出：

$$O = \{o_1, o_2, \dots, o_M\}$$

其中  $M$  是观测符号的数量。

### 3. 初始状态分布 $\pi$ ：

系统在初始时刻  $t = 0$  时处于各隐含状态的概率分布：

$$\pi = \{\pi_i\}, \quad \pi_i = P(X_0 = s_i), \quad \sum_{i=1}^N \pi_i = 1$$

### 4. 状态转移概率矩阵 $A$ ：

隐含状态之间的转移概率矩阵，表示在时间  $t$  从隐含状态  $s_i$  转移到状态  $s_j$  的概率：

$$A = \{a_{ij}\}, \quad a_{ij} = P(X_{t+1} = s_j | X_t = s_i)$$

#### 5. 观测概率矩阵 $B$ :

隐含状态生成观测符号的概率矩阵, 表示在隐含状态  $s_i$  下生成观测符号  $o_k$  的概率:

$$B = \{b_i(k)\}, \quad b_i(k) = P(O_t = o_k | X_t = s_i)$$

综上, 一个完整的隐马尔可夫模型可以表示为三元组:

$$\lambda = (A, B, \pi)$$

其中:

- $A$  是状态转移概率矩阵;
- $B$  是观测概率矩阵;
- $\pi$  是初始状态分布。

HMM 的目标是在给定观测序列  $O$  时, 推断出最有可能的隐含状态序列  $X$ , 或者计算观测序列的概率  $P(O|\lambda)$ 。

### 1.5 给出隐马尔可夫模型的 3 个典型问题的形式化描述。 (4 分)

解:

#### (1) 评估问题

对于一个给定的 HMM  $\lambda = (A, B, \pi)$  和观测序列  $O = \{O_1, O_2, \dots, O_T\}$ , 目标是计算观测序列  $O$  的出现概率:

$$P(O|\lambda) = \sum_X P(O|X, \lambda)P(X|\lambda)$$

其中,  $X = \{X_1, X_2, \dots, X_T\}$  是可能的隐含状态序列。

该问题通常通过前向算法或后向算法来高效求解。

#### (2) 解码问题

目标是找到最优隐含状态序列  $X = \{X_1, X_2, \dots, X_T\}$ , 使得:

$$X^* = \arg \max_X P(X|O, \lambda)$$

或者等价于:

$$X^* = \arg \max_X P(O, X|\lambda)$$

因为  $P(X|O, \lambda) \propto P(O, X|\lambda)$ 。

这个问题通常通过Viterbi算法来求解。

### (3) 学习问题

目标是找到模型参数  $\lambda = (A, B, \pi)$ ，使得：

$$\lambda^* = \arg \max_{\lambda} P(O|\lambda)$$

该问题通常通过Baum-Welch算法（即期望最大化算法，EM Algorithm）来求解。

## 1.6 给定隐马尔可夫模型 $\lambda$ ，如何计算该模型生成某个观察序列 $O$ 的概率？请给出数学公式及对应的推导。（5分）

解：

给定隐马尔可夫模型  $\lambda = (A, B, \pi)$ ，以及观测序列  $O = \{O_1, O_2, \dots, O_T\}$ ，我们希望计算观测序列  $O$  出现的概率  $P(O|\lambda)$ 。这个问题称为评估问题，计算方法通常是通过前向算法来高效解决。

前向算法是一种动态规划算法，可以有效地计算  $P(O|\lambda)$ 。通过引入前向概率，我们将计算分为逐步累积。

定义  $\alpha_t(i)$  表示在时刻  $t$  隐含状态为  $s_i$  且已生成部分观测序列  $O_1, O_2, \dots, O_t$  的概率为前向概率：

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, X_t = s_i | \lambda)$$

即：

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t \text{ and } X_t = s_i | \lambda)$$

前向概率可以通过以下递推公式计算：

1. 在初始时刻  $t = 1$ ，计算前向概率：

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

其中  $\pi_i$  是初始状态  $s_i$  的概率， $b_i(O_1)$  是在状态  $s_i$  下观测到  $O_1$  的概率。

2. 从时刻  $t$  到  $t + 1$ ，使用递推公式依次计算每个时刻  $t$  的前向概率：

$$\alpha_{t+1}(j) = \left( \sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1}), \quad 1 \leq j \leq N$$

其中  $a_{ij}$  是从状态  $s_i$  转移到状态  $s_j$  的转移概率， $b_j(O_{t+1})$  是在状态  $s_j$  下生成观测值  $O_{t+1}$  的概率。

3. 完成所有观测序列时，计算观测序列  $O$  的总概率，即所有可能终止状态的前向概率的和：

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

其中  $\alpha_T(i)$  是在时刻  $T$  状态为  $s_i$  时生成整个观测序列的概率。

## 1.7 给定隐马尔可夫模型 $\lambda$ 、以及对应的观察序列 $O$ ，如何得到该模型产生该观察序列的最优隐含状态的序列 $Q$ ？请给出数学公式及对应的推导。（5 分）

解：

### (1)问题定义

要从给定的隐马尔可夫模型  $\lambda = (A, B, \pi)$  及观测序列  $O = \{O_1, O_2, \dots, O_T\}$  中找到最优的隐含状态序列  $Q = \{q_1, q_2, \dots, q_T\}$ ，这个问题称为解码问题，可以使用Viterbi<sup>[2]</sup>算法。该算法通过动态规划的方法，找到使得观测序列概率  $P(O, Q|\lambda)$  最大的隐含状态序列。

我们希望找到最优的隐含状态序列  $Q = \{q_1, q_2, \dots, q_T\}$ ，即：

$$Q^* = \arg \max_Q P(Q|O, \lambda)$$

根据贝叶斯定理，等价于最大化联合概率：

$$Q^* = \arg \max_Q P(O, Q|\lambda)$$

这意味着我们要找到使得观测序列  $O$  和隐含状态序列  $Q$  联合概率最大的隐含状态序列。

### (2)变量引入

Viterbi算法通过递归计算来找到最优隐含状态序列。它定义了一个变量  $\delta_t(i)$ ，表示在时刻  $t$  时处于状态  $s_i$  并且生成部分观测序列  $O_1, O_2, \dots, O_t$  的最大概率：

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, X_t = s_i, O_1, O_2, \dots, O_t | \lambda)$$

即  $\delta_t(i)$  是在时刻  $t$  到达状态  $s_i$  的最优路径的最大概率。

此外，为了找到完整的最优隐含状态序列  $Q^*$ ，我们在递推过程中还需要记录每一步的最优路径。引入一个辅助变量  $\psi_t(j)$ ，表示在时刻  $t$  处于状态  $s_j$  时，时刻  $t-1$  的最优隐含状态：

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

根据这个辅助变量，可以通过回溯来得到最优隐含状态序列  $Q^*$ 。

### (3)算法步骤

Viterbi算法的核心是利用动态规划来递推计算最优路径的概率。递推步骤包括以下几部分：

1. **初始化** (时刻  $t = 1$ ) :

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

其中  $\pi_i$  是初始时刻处于状态  $s_i$  的概率,  $b_i(O_1)$  是在状态  $s_i$  生成观测值  $O_1$  的概率。

对所有  $i$ , 设  $\psi_1(i) = 0$ 。

2. **递推** (时刻  $t \geq 2$ ) :

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(O_t), \quad 1 \leq j \leq N$$

其中  $a_{ij}$  是从状态  $s_i$  转移到状态  $s_j$  的概率,  $b_j(O_t)$  是在状态  $s_j$  生成观测值  $O_t$  的概率。

并记录：

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$$

3. **终止** (时刻  $t = T$ ) :

计算终止时刻  $T$  的最优隐含状态：

$$P^* = \max_{1 \leq i \leq N} \delta_T(i)$$

并记录下对应的最优终止状态  $q_T^*$ ：

$$q_T^* = \arg \max_{1 \leq i \leq N} \delta_T(i)$$

4. **回溯** (从  $t = T - 1$  到  $t = 1$ ) :

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

**1.8 给你一堆观察数据  $O_1, O_2, \dots, O_N$ , 并假设这些观察数据符合隐马尔可夫过程的假设, 如何估计出一个最优的隐马尔可夫模型  $\lambda$ , 该模型产生这些观察数据的概率最优。 (5 分)**

**解：**

这个问题是隐马尔可夫模型的学习问题, 即在给定观测数据  $O_1, O_2, \dots, O_N$  的情况下, 如何估计出最优的 HMM 参数  $\lambda = (A, B, \pi)$ , 使得该模型产生这些观测数据的概率最大, 即最大化  $P(O|\lambda)$ 。该问题通常通过 Baum-Welch 算法 (即期望最大化算法, EM Algorithm) 来解决。

Baum-Welch 算法依赖于前向-后向算法, 因此在进入 Baum-Welch 算法之前, 需要定义前向概率和后向概率。

- **前向概率**  $\alpha_t(i)$ : 表示在时刻  $t$  处于隐含状态  $s_i$  且已经观测到部分序列  $O_1, O_2, \dots, O_t$  的概率:

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, X_t = s_i | \lambda)$$

- **后向概率**  $\beta_t(i)$ : 表示在时刻  $t$  处于隐含状态  $s_i$  且生成余下观测序列  $O_{t+1}, O_{t+2}, \dots, O_T$  的概率:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | X_t = s_i, \lambda)$$

Baum-Welch算法的步骤如下:

1. **初始化**: 初始化 HMM 的参数  $\lambda = (A, B, \pi)$ , 通常是随机初始化或基于某些经验的初始化。

2. **E步 (Expectation Step)**: 根据当前模型参数  $\lambda = (A, B, \pi)$ , 计算隐含状态的期望。

- **计算前向概率**  $\alpha_t(i)$ :

通过递推方式计算  $\alpha_t(i)$ , 即从时刻  $t = 1$  开始计算, 直到  $t = T$ :

$$\alpha_1(i) = \pi_i b_i(O_1), \quad \alpha_{t+1}(j) = \left( \sum_{i=1}^N \alpha_t(i) a_{ij} \right) b_j(O_{t+1})$$

- **计算后向概率**  $\beta_t(i)$ :

从时刻  $t = T$  开始递推计算, 直到  $t = 1$ :

$$\beta_T(i) = 1, \quad \beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

- **计算**  $\gamma_t(i)$ :

$\gamma_t(i)$  表示在时刻  $t$  系统处于状态  $s_i$  的概率:

$$\gamma_t(i) = P(X_t = s_i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{P(O | \lambda)}$$

- **计算**  $\xi_t(i, j)$ :

$\xi_t(i, j)$  表示在时刻  $t$  系统从状态  $s_i$  转移到状态  $s_j$  的概率:

$$\xi_t(i, j) = P(X_t = s_i, X_{t+1} = s_j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)}$$

3. **M步 (Maximization Step)**: 基于计算的期望值更新模型参数  $A$ 、 $B$  和  $\pi$ , 以最大化观测数据的似然。

- **更新初始状态分布**  $\pi$ :

初始状态的更新由时刻  $t = 1$  处的状态概率  $\gamma_1(i)$  给出:

$$\pi_i = \gamma_1(i)$$



- **更新转移概率矩阵  $A$ :**

状态从  $s_i$  转移到  $s_j$  的概率由所有时刻  $t$  的转移频率给出:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

- **更新观测概率矩阵  $B$ :**

在状态  $s_i$  下生成观测  $O_k$  的概率由所有时刻  $t$  处在状态  $s_i$  并观测到  $O_k$  的次数给出:

$$b_i(k) = \frac{\sum_{t=1}^T \gamma_t(i) \cdot \mathbb{I}(O_t = o_k)}{\sum_{t=1}^T \gamma_t(i)}$$

其中  $\mathbb{I}(O_t = o_k)$  是一个指示函数, 当  $O_t = o_k$  时取值为 1, 否则为 0。

4. **终止条件:** 重复执行 E 步和 M 步, 直到对数似然  $\log P(O|\lambda)$  的变化量小于预定的阈值, 或达到最大迭代次数。

## 1.9 什么是高斯混合模型? 高斯混合模型如何与隐马尔可夫模型结合? (5 分)

解:

(1)

高斯混合模型 (Gaussian Mixture Model, GMM) 是一种用于概率密度估计的统计模型, 它假设数据由多个高斯分布组成。每个高斯分布代表数据的一个潜在子群。GMM 的核心思想是, 复杂的分布可以通过多个简单的高斯分布 (正态分布) 的加权组合来表示。

高斯混合模型的基本组成部分为:

1. **高斯分布:** GMM 假设每个数据点来自一个高斯分布。高斯分布由均值和协方差矩阵确定, 用来描述数据点的中心和扩展范围。
2. **混合系数:** GMM 中的混合系数表示每个高斯分布在整体分布中的权重, 这些权重相加为 1。
3. **潜在变量:** GMM 引入了隐含变量来表示每个数据点属于哪个高斯分布, 这些变量通常通过期望最大化 (Expectation-Maximization, EM) 算法来估计。

高斯混合模型的主要步骤有:

1. **初始化:** 随机初始化每个高斯分布的参数 (均值、协方差和混合系数)。
2. **E 步 (期望步):** 计算每个数据点属于每个高斯分布的概率 (即后验概率), 根据当前的高斯分布参数。
3. **M 步 (最大化步):** 利用 E 步的结果, 重新估计每个高斯分布的参数, 使得模型的似然函数最大化。
4. **迭代:** 不断交替执行 E 步和 M 步, 直到收敛, 即参数不再显著变化。

(2)

高斯混合模型（GMM）与隐马尔可夫模型（Hidden Markov Model, HMM）的结合在许多序列数据的建模任务中有广泛应用，尤其是在语音识别、时间序列分析等领域。这种结合被称为高斯混合隐马尔可夫模型（GMM-HMM），它结合了GMM的多峰分布拟合能力和HMM的时间依赖性结构。下面详细解释它们的结合方式。

在传统的HMM中，观测值通常被假设服从某种简单分布（例如单峰的正态分布），这对于复杂的观测数据可能不够灵活。将GMM引入到HMM的观测模型中，能够更灵活地表示观测数据的概率分布。

1. **GMM作为观测模型**：在GMM-HMM中，每个隐含状态对应的观测值分布不是单一的高斯分布，而是高斯混合模型。这样，每个隐状态下的观测值可以通过多个高斯分布的组合来描述，这大大提高了模型对复杂数据的拟合能力。
  - 在每个隐含状态下，观测值的概率分布被建模为高斯混合模型，即观测值是由多个高斯分布的加权和生成的。
  - 通过这种方式，隐马尔可夫模型的每个隐状态不再只使用一个简单的正态分布表示观测值，而是通过多个高斯分布来捕捉观测数据的复杂性。
2. **隐状态的时间依赖性**：HMM的隐含状态具有时间依赖性，状态之间的转移遵循马尔可夫性质（即当前状态只依赖于前一个状态），并且这种依赖关系通过状态转移矩阵建模。结合GMM后，状态转移的过程保持不变，HMM仍然可以对状态之间的动态变化进行建模。
3. **期望最大化（EM）算法的扩展**：在参数估计方面，GMM-HMM通常通过一种改进的EM算法来进行训练。这个算法结合了HMM的前向-后向算法与GMM中的期望最大化步骤，逐步优化模型参数，包括状态转移概率、混合系数、每个高斯分布的均值和协方差矩阵等。

高斯混合隐马尔可夫模型（GMM-HMM）将GMM灵活的概率密度建模能力与HMM的时间依赖性结合在一起，能够在处理具有复杂分布的时间序列数据时展现出强大的表现力。通过GMM对每个状态下观测值的多峰分布进行建模，同时利用HMM对状态序列进行建模，GMM-HMM能够有效处理多样、复杂的序列数据。例如在语音识别中，声学特征（如MFCC）通常是由GMM-HMM来建模的。GMM描述声音的多峰特性（如噪声和不同发音的变化），而HMM描述语音的时间依赖性（如音素或词语之间的序列关系）。

**1.10 请结合相关文献，介绍一个隐马尔可夫模型在语音及自然语言处理领域的最新应用实例，包括但不限于语音识别、语音合成、文本分词等。注意需要解释清楚模型是如何应用到实际任务中的，以及是否存在方法上的改进。（5分）**

解：

在语音合成领域，隐马尔可夫模型（HMM）过去曾是主流技术之一，主要用于建模语音信号的时间和发音特性。然而，随着神经网络的兴起，基于HMM的语音合成技术逐渐被序列到序列（seq2seq）模型和深度学习技术所取代，例如Tacotron 2。Tacotron 2通过注意力机制实现了语音合成中的对齐，但由于其非单调注意力机制的固有问题（如对齐失败、语音跳过等），生成语音的质量和效率有所限制。

为了克服上述问题，ICASSP 2022的论文《Neural HMMs are all you need (for high-quality attention-free TTS)》<sup>[3]</sup>提出了一个基于 HMM 的神经 TTS 模型，通过将 Tacotron 2 中的注意力机制替换为神经 HMM，实现了以下几个方面的改进：

### 1. 概率性:

- 原有的 Tacotron 2 模型不是概率性的，这使得模型的预测结果难以解释，并且难以进行一些重要的分析，例如解码器状态的概率分布。
- 基于 HMM 的神经 TTS 模型是一个完全概率性的模型，可以计算精确的序列对数似然，并使用反向传播和自动微分来优化模型参数。这使得模型更加可解释，并且可以进行更深入的分析 and 应用。

### 2. 单调对齐:

- 原有的 Tacotron 2 模型使用非单调的注意力机制，这会导致训练时间增加，并可能使合成语音变得不连贯。
- 基于 HMM 的神经 TTS 模型使用神经 HMM 进行序列到序列的建模，并且状态转换概率仅依赖于当前状态和过去观测值。这使得模型能够进行单调对齐，从而避免注意力机制中的跳过和重复问题，使合成语音更加流畅。

### 3. 其他改进:

- **模型大小:** 基于 HMM 的神经 TTS 模型比 Tacotron 2 模型更小，这使得模型更容易部署和使用。
- **学习效率:** 基于 HMM 的神经 TTS 模型学习速度更快，需要更少的训练数据和迭代次数即可达到可理解的水平。
- **鲁棒性:** 基于 HMM 的神经 TTS 模型不会出现 Tacotron 2 模型中常见的注意力失败问题，例如跳过和重复。
- **可控性:** 基于 HMM 的神经 TTS 模型可以轻松地控制合成语音的语速。

总而言之，基于 HMM 的神经 TTS 模型通过将 HMM 和神经网络的优点结合起来，实现了高质量的无注意力语音合成，并取得了显著的性能改进。

# 实验任务一：Viterbi 解码算法实现

## 2.1 前向算法（7 分）

```
def forward(self, ob):
    """HMM Forward Algorithm.

    Args:
        ob (array, with shape(T,)): (o1, o2, ..., oT), observations

    Returns:
        fwd (array, with shape(T, 3)): fwd[t, s] means full-path forward probability towards
            timestep t given the observation ob[0:t+1].
            给定观察ob[0:t+1]情况下t时刻到达状态s的所有可能路径的概率
        prob: the probability of HMM model generating observations.

    """
    T = ob.shape[0]
    fwd = np.zeros((T, self.total_states))

    # Begin Assignment
    # Initialization step
    fwd[0, :] = self.pi * self.B[:, ob[0]]

    # Recursion step
    for t in range(1, T):
        for s in range(self.total_states):
            fwd[t, s] = np.sum(fwd[t - 1, :] * self.A[:, s]) * self.B[s, ob[t]]
    # End Assignment

    prob = fwd[-1, :].sum()

    return fwd, prob
```

## 2.2 后向算法 (7 分)

```
def backward(self, ob):
    """HMM Backward Algorithm.

    Args:
        ob (array, with shape(T,)): (o1, o2, ..., oT), observations

    Returns:
        bwd (array, with shape(T, 3)): bwd[t, s] means full-path backward probability towards
            timestep t given the observation ob[t+1::]
            给定观察ob[t+1::]情况下t时刻到达状态s的所有可能路径的概率
        prob: the probability of HMM model generating observations.

    """
    T = ob.shape[0]
    bwd = np.zeros((T, self.total_states))

    # Begin Assignment
    # Initialization step
    bwd[T - 1, :] = 1

    # Recursion step
    for t in reversed(range(T - 1)):
        for s in range(self.total_states):
            bwd[t, s] = np.sum(bwd[t + 1, :] * self.A[s, :] * self.B[:, ob[t + 1]])
    # End Assignment

    prob = (bwd[0, :] * self.B[:, ob[0]] * self.pi).sum()

    return bwd, prob
```

## 2.3 Viterbi 解码算法 (6 分)

```
def viterbi(self, ob):
    """Viterbi Decoding Algorithm.

    Args:
        ob (array, with shape(T,)): (o1, o2, ..., oT), observations

    Variables:
        delta (array, with shape(T, 3)): delta[t, s] means max probability towards state s
            timestep t given the observation ob[0:t+1]
            给定观察ob[0:t+1]情况下t时刻到达状态s的概率最大的路径
        phi (array, with shape(T, 3)): phi[t, s] means prior state s' for delta[t, s]
            给定观察ob[0:t+1]情况下t时刻到达状态s的概率最大的路径的

    Returns:
        best_prob: the probability of the best state sequence
        best_path: the best state sequence

    """
    T = ob.shape[0]
    delta = np.zeros((T, self.total_states))
    phi = np.zeros((T, self.total_states), np.int32)
    best_prob, best_path = 0.0, np.zeros(T, dtype=np.int32)

    # Begin Assignment
    # Initialization step
    delta[0, :] = self.pi * self.B[:, ob[0]]

    # Recursion step
    for t in range(1, T):
        for s in range(self.total_states):
            delta[t, s] = np.max(delta[t - 1, :] * self.A[:, s]) * self.B[s, ob[t]]
            phi[t, s] = np.argmax(delta[t - 1, :] * self.A[:, s])
    # End Assignment

    best_path[T - 1] = delta[T - 1, :].argmax(0)
    best_prob = delta[T - 1, best_path[T - 1]]
    for t in reversed(range(T - 1)):
        best_path[t] = phi[t + 1, best_path[t + 1]]

    return best_prob, best_path
```

最后运行代码文件 viterbi.py 得到以下结果，可以看到和 ground truth decoding results 相同：

```
PS C:\Users\ysn\Desktop\研究生课内资料\语音处理> python viterbi.py
0.0342796928
[[0.1      0.16      0.28      ]
 [0.127    0.1176   0.027    ]
 [0.03344  0.05616   0.045024 ]
 [0.0258544 0.03428928 0.00772992]
 [0.00914158 0.02052196 0.00461615]]
0.0342796928
[[0.06272704 0.06416424 0.06335968]
 [0.127352   0.124808   0.126984   ]
 [0.2536     0.258     0.2512     ]
 [0.5        0.51      0.5        ]
 [1.         1.         1.         ]]
0.0025930799999999999
[2 0 2 0 1]
```

## 实验任务二：基于 HMM 的强制对齐实验

**3.1（课上检查6分，实验报告6分）** 课上检查实验结果，检查内容:MFA是否安装成功; MFA align生成的中间文件； MFA align生成的textgrid。实验报告中记录实验过程。

下述每一步执行命令均在windows环境的vscode的终端进行操作。

### 1. 安装最新版conda

验证结果如下：

```
(aligner) PS C:\Users\ysn\Documents\MFA> conda --version
conda 24.9.2
```

### 2. 创建新环境并安装MFA

验证结果如下：

```
(aligner) PS C:\Users\ysn\Documents\MFA> MFA version
3.2.0
```

### 3. 下载数据集并放到合适的位置

(1)将mfadir.rar下载到C:\Users\ysn\Documents\MFA并解压（任务二说明README.md里没说要解压，建议补上）

(2)将pinyin-lexicon-r-new.txt下载到C:\Users\ysn\Documents\MFA

(3)将new\_acoustic\_model.zip下载到C:\Users\ysn\Documents\MFA\pretrained\_models\acoustic

#### 4. **验证对齐可行性**

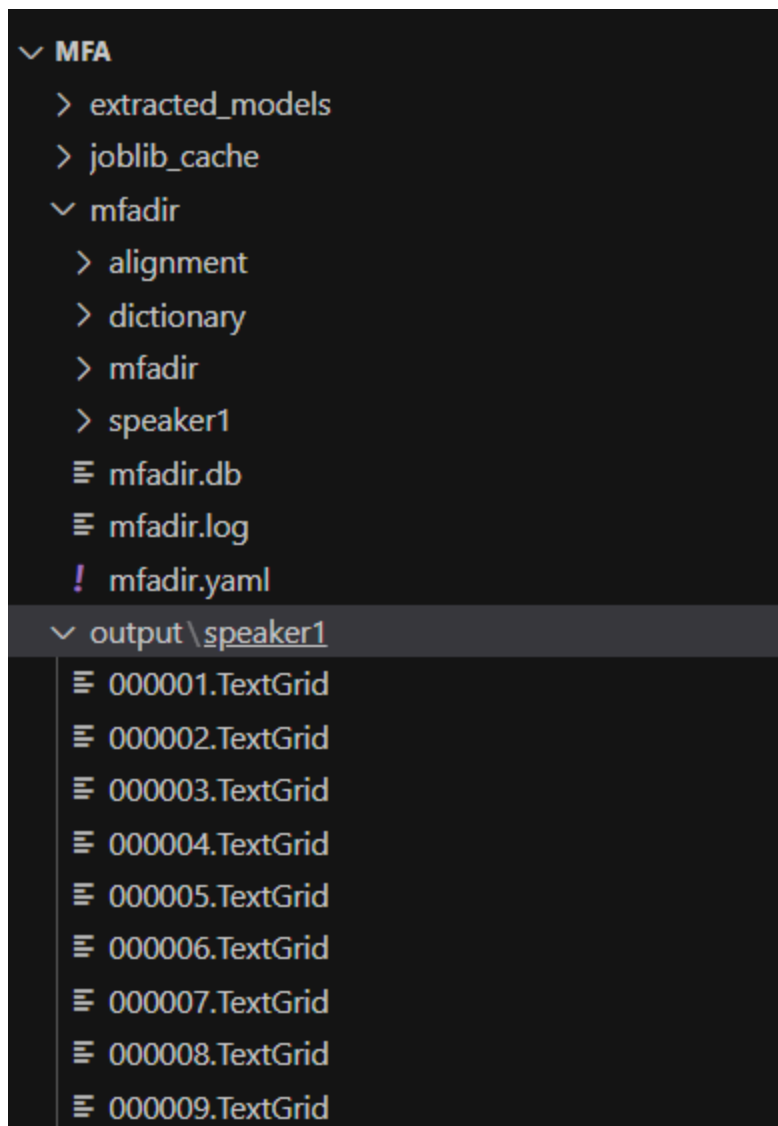
执行命令得到运行结果如下：



```
PS C:\Users\ysn\Documents\MFA> mfa validate mfidir pinyin-lexicon-r-new.txt pretrained_models/acoustic/new_acoustic_model
INFO Setting up corpus information...
INFO Loading corpus from source files...
100% 10,000/100 [ 0:00:00 < 0:00:00 , 1,809 it/s ]
INFO Found 1 speaker across 10000 files, average number of utterances per speaker: 10000.0
INFO Initializing multiprocessing jobs...
WARNING Number of jobs was specified as 3, but due to only having 1 speakers, MFA will only use 1 jobs. Use the --single_speaker flag if you would like to
split utterances across jobs regardless of their speaker.
INFO Normalizing text...
100% 10,000/10,000 [ 0:00:01 < 0:00:00 , ? it/s ]
INFO Generating MFCCs...
100% 10,000/10,000 [ 0:00:47 < 0:00:00 , 261 it/s ]
INFO Calculating CMVN...
INFO Generating final features...
100% 10,000/10,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO Creating corpus split...
100% 10,000/10,000 [ 0:00:01 < 0:00:00 , ? it/s ]
INFO Corpus
INFO 10000 sound files
INFO 10000 text files
INFO 1 speakers
INFO 10000 utterances
INFO 42680.203 seconds total duration
INFO Sound file read errors
INFO There were no issues reading sound files.
INFO Feature generation
INFO There were no utterances missing features.
INFO Files without transcriptions
INFO There were no sound files missing transcriptions.
INFO Transcriptions without sound files
INFO There were no transcription files missing sound files.
INFO Dictionary
INFO Out of vocabulary words
WARNING 7 OOV word types
WARNING 8 total OOV tokens
WARNING For a full list of the word types, please see: C:\Users\ysn\Documents\MFA\mfadir\oovs_found.txt. For a by-utterance breakdown of missing words, see:
C:\Users\ysn\Documents\MFA\mfadir\utterance_oovs.txt
INFO Training
INFO Creating subset directory with 2000 utterances...
100% 2,000/2,000 [ 0:00:01 < 0:00:00 , ? it/s ]
INFO Initializing training for monophone...
INFO Compiling training graphs...
INFO Generating initial alignments...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO Initialization complete!
INFO monophone - Iteration 1 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:01:18 < 0:00:00 , 25 it/s ]
INFO Accumulating statistics...
100% 1,997/2,000 [ 0:00:03 < -:--:-- , ? it/s ]
INFO monophone - Iteration 2 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:12 < 0:00:00 , 96 it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 3 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:08 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 4 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:07 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 5 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:06 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 6 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:06 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 7 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:06 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 8 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:07 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 9 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:08 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 10 of 40
INFO Generating alignments...
100% 2,000/2,000 [ 0:00:07 < 0:00:00 , ? it/s ]
INFO Accumulating statistics...
100% 2,001/2,000 [ 0:00:03 < 0:00:00 , ? it/s ]
INFO monophone - Iteration 11 of 40
INFO Accumulating statistics...
```

[illegible]



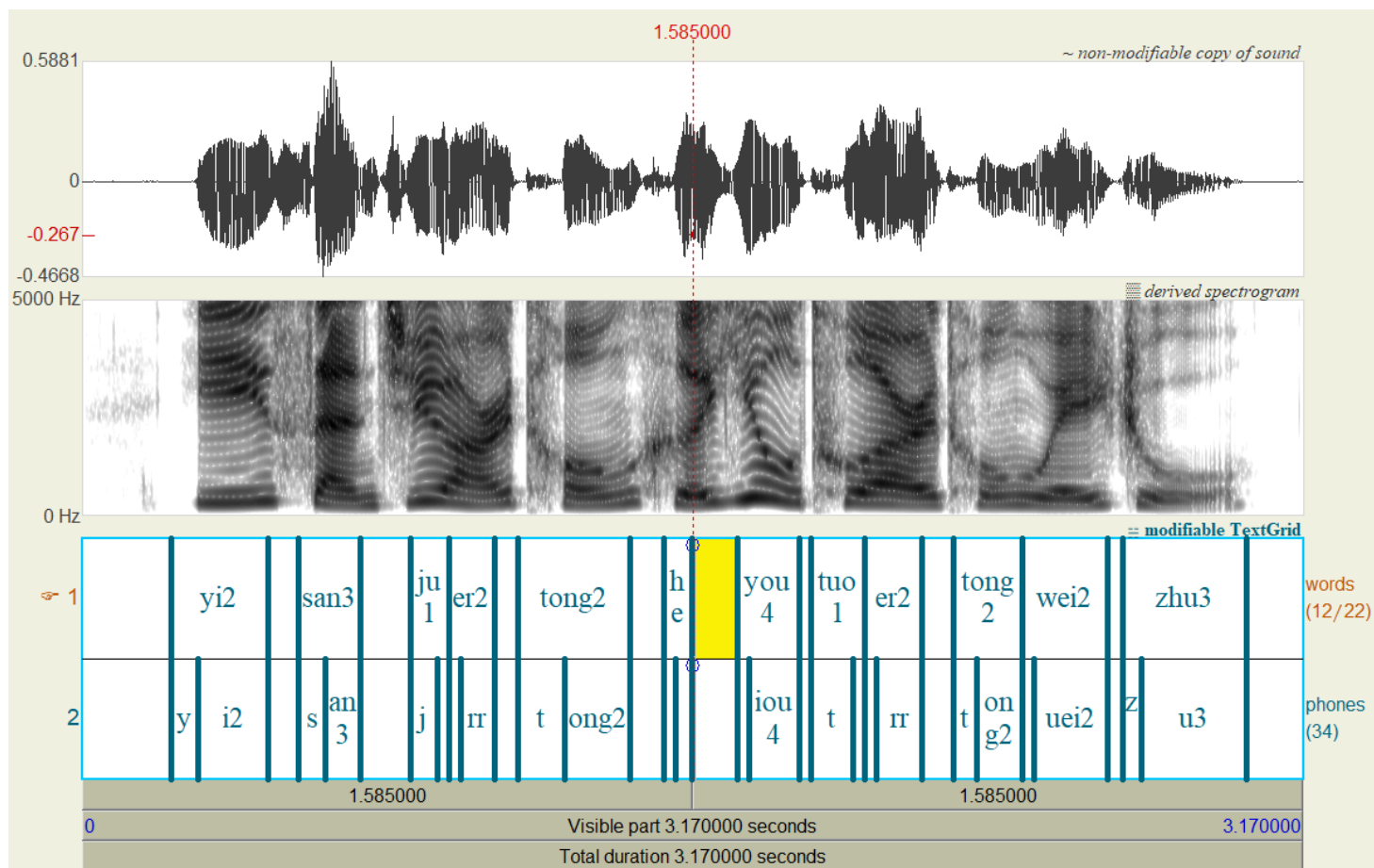


### 3.2 (文本内容分析3分，结合波形、频谱、基频分析6分) 分析最后生成的textgrid的文本内容，并结合音频的波形、频谱、基频进行分析。

使用 Praat，选择 000009.wav 和 000009.TextGrid 进行分析：

(1)

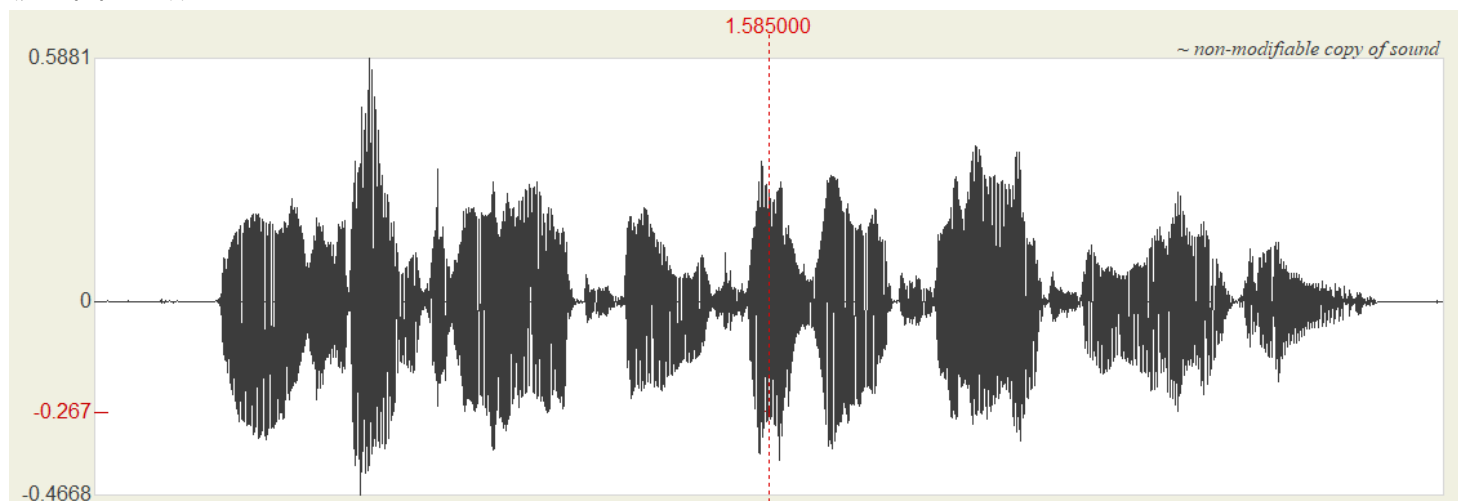
Praat分析结果如下图所示：



根据下方的拼音和播放内容可得：最后生成的 textgrid 的文本内容为“以散居儿童和幼托儿童为主”。

(2)

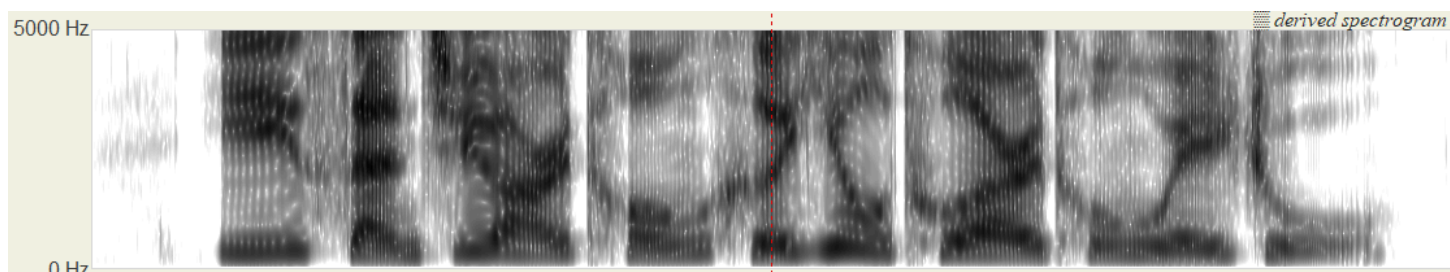
波形图如下所示：



其中峰值较高的地方表示音频中音强较大，峰值较低的地方表示音频中音强较小。可以看出在与下面音节或音素对应的部分，峰值分布都较为清晰密集。

(3)

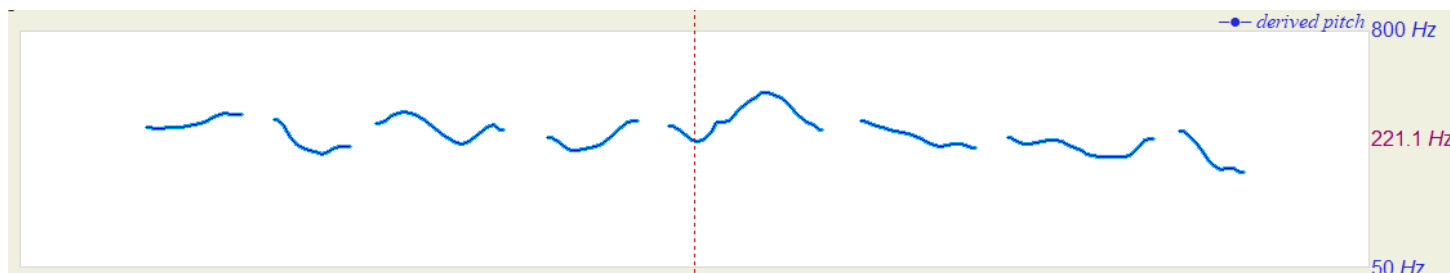
频谱图如下所示：



显示范围为 0-5000Hz。频谱图可以观察到声音的频率分布，可以看到频谱的变化与音节的变化同步。

(4)

基频图如下所示：



显示范围为 50-800Hz。从图中可以看出基频的变化，以及音频中不同部分的音高和说话人的语调相关。

### 3.3 （3分）如何使用HMM（隐马尔可夫模型）在MFA中进行音素对齐？

在MFA中，HMM（隐马尔可夫模型）是用于实现音素对齐的核心技术。具体而言，MFA使用HMM对语音信号和文本之间的关系进行建模，通过状态序列来表示音素序列，使得音频与标注文本实现精确对齐。以下是MFA中使用HMM进行音素对齐的大致步骤：

#### 1. 数据准备：

- 提供语音音频文件（通常为 .wav 格式）以及对应的文本标注文件（通常为 .txt 格式）。文本文件中应包含与音频内容一致的逐词或逐句的标注。

#### 2. 训练声学模型：

- MFA利用HMM来训练声学模型，其中HMM的状态表示音素，状态间的转换概率用来描述音素的时间结构。MFA提供了一些预训练的声学模型，但你也可以使用自己数据来训练自定义声学模型。

#### 3. 初始化对齐：

- 对齐过程首先使用文本的音素表示，通常通过词典将单词转为音素序列。词典定义了每个单词对应的音素列表，确保音素标注的准确性。
- 如果没有现成的词典，MFA会通过G2P（字到音转换）工具生成音素表示。

#### 4. Viterbi算法对齐：

- 使用Viterbi算法找到给定音频和音素序列的最可能的状态路径。具体而言，Viterbi算法会遍历所有可能的HMM状态序列，通过最大化概率找到最合适的音频片段对应的音素标签，从而实

现音素对齐。

- 在这个过程中，HMM的状态转移概率和发射概率用于决定每个音素在时间上的位置以及持续时间。

#### 5. 迭代优化：

- MFA通常会对初始的对齐结果进行多次迭代优化，以提高对齐精度。这些迭代通过不断更新HMM参数（如发射概率和转移概率）来减少对齐错误。
- 这种迭代过程也类似于Baum-Welch算法，用于重新估计HMM参数，进而改善对齐的质量。

#### 6. 输出对齐结果：

- 最终，MFA输出对齐结果，通常为TextGrid格式文件，这种格式可以被如Praat之类的工具用来查看和分析音素对齐的时间边界。

简而言之，MFA通过将音频和文本信息输入到HMM中，使用Viterbi算法寻找最佳路径，将音频信号与音素标注进行对齐。通过使用HMM的状态转移和发射概率，MFA可以有效地对齐音素边界，使对齐结果尽可能准确地反映真实发音情况。

### 3.4 （3分）在MFA中，HMM的训练过程是如何进行的，以及如何利用训练好的HMM模型进行音素级别的对齐？

#### （1）HMM模型的训练过程

MFA中的HMM训练主要是为了生成声学模型，使其能够捕捉音频特征和音素之间的对应关系。训练过程主要包括以下几个部分：

1. **音频和文本对齐数据准备**：需要提供音频文件（如 .wav 格式）和对应的文本标注文件（如 .txt 文件）。文本文件应当包含与音频内容一致的标注，通常逐句或逐词标注。
2. **音频特征提取**：对每个音频文件进行特征提取，生成每一帧的声学特征（通常为梅尔频率倒谱系数 MFCC）。这些特征用于描述语音信号的时间频率特性。
3. **初始参数设定**：MFA在训练开始时，需要对HMM的初始参数进行设定。HMM参数主要包括：
  - **发射概率**：即音素状态生成声学特征的概率。初始时可通过均值和协方差矩阵来随机初始化这些参数。
  - **转移概率**：即从一个状态（音素）转换到下一个状态的概率。初始状态的转移概率通常是根据音素的平均持续时间进行估计。
4. **使用Baum-Welch算法进行训练**：MFA使用Baum-Welch算法对HMM参数进行重新估计。Baum-Welch算法是HMM的参数估计算法。这是一种期望最大化（EM）算法，它在每次迭代中分为两个步骤：期望步骤（E步）和最大化步骤（M步）。
  - **E步**：计算在给定现有模型参数的情况下，观察序列（即音频特征）匹配每个状态序列的期望概率。
  - **M步**：根据E步的结果，更新模型参数（如发射概率和转移概率），使得模型更好地匹配训练数据。

通过多次迭代，模型的参数逐渐收敛，使得HMM能准确地表示音频与音素之间的关系。



## (2) 利用训练好的HMM模型进行音素对齐

一旦声学模型训练完成，可以利用训练好的HMM模型进行音素级别的对齐。具体步骤如下：

### 1. 初始化对齐

- **文本转音素序列**：MFA使用词典将文本标注转换为音素序列。词典定义了每个单词的音素表示。如果没有词典，MFA会通过G2P（字到音）模型生成音素表示。
- **创建初始HMM**：每个音素用一个HMM表示，音素序列形成一连串HMM模型，这些HMM通过状态转移构成一个完整的状态网络。

2. **使用Viterbi算法进行对齐**：将音频信号特征输入到训练好的HMM中。使用Viterbi算法来寻找使观测数据（即音频特征）最可能的状态路径。通过找到音素状态的最可能路径，确定每个音素在音频中的开始和结束时间，实现音素级别的对齐。

3. **生成对齐文件**：对齐的结果通常以TextGrid格式输出，TextGrid是Praat软件支持的格式，用于标注音频中的时间间隔。这些文件标注了每个音素的边界，可以进一步用于语音分析和处理。

## 3.5 （3分）在MFA中，如何根据HMM模型调整参数以优化音素对齐的准确性？针对所选择的参数写明理由即可，可以通过HMM模型的概念出发。

(1) **状态数**：HMM通常用多个状态来表示一个音素，状态的数量影响模型对音素的时间结构的表示能力。音素的发音具有持续时间，使用更多状态可以更好地捕捉到这种时间上的变化。如果状态数太少，HMM可能无法充分捕捉音素的内部动态特性；如果状态数太多，可能会过拟合。因此，可以适当增加每个音素的状态数，以便对音素在时间上的特性进行更精细的建模，从而提高对齐准确性。

(2) **发射概率模型**：发射概率描述了给定状态生成特定观测（即音频特征）的概率，通常使用高斯混合模型（GMM）进行建模。通过增加高斯混合模型的分数量，可以使发射概率的模型更加精细，以适应复杂的音频特征。这有助于更准确地匹配音素和音频的特征，从而提高对齐质量。GMM成分数量的增加可以提高模型的表现力，但也增加了计算复杂度，因此需要在准确性和计算效率之间找到平衡。

(3) **状态转移概率**：状态转移概率控制音素之间和音素内部状态之间的转换。通过调整状态转移概率，可以控制音素的持续时间特性。例如，可以增加同一状态保持的概率，从而使模型更倾向于对音素有合理的持续时间估计，而不是频繁地改变状态。这在对齐持续时间较长的音素时尤为重要。对这些概率进行合理的调整，可以避免模型在对齐过程中对音素的持续时间估计不准确。

(4) **语言模型约束**：MFA中通常使用词典和语言模型来限制可能的音素序列，从而提高对齐的准确性。通过优化词典（确保词典中有足够丰富的词汇）和使用更精确的语言模型，可以减少在对齐过程中不合理音素序列的出现，提高整体对齐质量。合理的语言模型约束有助于HMM模型排除掉一些低概率的音素组合，进而提升对齐精度。

(5) **初始参数估计**：HMM模型的训练使用Baum-Welch算法，这种算法对初始参数敏感。更好的初始参数可以帮助加速训练的收敛，并提高最终模型的质量。例如，可以利用一些预训练模型来初始化发射概



率和转移概率，从而避免陷入次优解。此外，利用已有的类似语料数据进行初始化，也可以显著提高模型的对齐效果。

(6) **正则化和平滑**：在训练HMM时，可能会出现一些概率为零的状态转移或发射概率，这可能会导致模型在面对新的数据时表现不佳。通过引入平滑技术（例如拉普拉斯平滑），可以避免这种情况，确保所有状态和观测都有非零概率，从而提升模型在面对不确定输入时的鲁棒性。此外，正则化有助于防止模型过拟合，尤其是对于较小的数据集，可以提高对未知音频的泛化能力。

### **3.6（附加题，上限3分）：官网下载标贝女声数据集，自己生成MFA数据集文件，提供脚本代码截图以及运行结果截图。**

首先从官网<https://www.data-baker.com/data/index/TNtts/> 下载数据集，得到BZNSYP.rar文件，再解压得到BZNSYP。

然后假定脚本文件在BZNSYP的父文件夹中，写成脚本代码MFA\_dataset.py如下：

```

import os
import shutil

base_dir = 'BZNSYP'
mfa_base_dir = 'MFAdataset'

# 创建MFAdataset/speaker1文件夹
os.makedirs(mfa_base_dir, exist_ok=True)
os.makedirs(os.path.join(mfa_base_dir, 'speaker1'), exist_ok=True)

# 将wav文件复制到speaker1文件夹下
wav_source = os.path.join(base_dir, 'Wave')
wav_destination = os.path.join(mfa_base_dir, 'speaker1')
if os.path.exists(wav_source):
    for wav_file in os.listdir(wav_source):
        if wav_file.endswith('.wav'):
            shutil.copy(os.path.join(wav_source, wav_file), wav_destination)

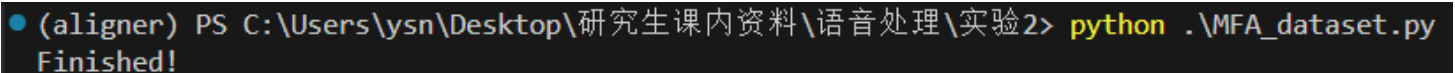
# 生成lab文件并保存到speaker1文件夹下
input_file = os.path.join(base_dir, "ProsodyLabeling/000001-010000.txt")
line_number = 0
with open(input_file, 'r', encoding='utf-8') as file:
    for line in file:
        line_number += 1
        if line_number % 2 == 0:
            with open(os.path.join(mfa_base_dir, f'speaker1/{line_number//2:06d}.lab'), 'w', encoding='utf-8') as lab_file:
                lab_file.write(line[1:])#去掉开头的缩进符号

print("Finished!")

```

只需执行该脚本，即可生成一个同一级的文件夹MFAdataset，即符合要求的MFA数据集文件。

脚本运行结果截图如下所示：

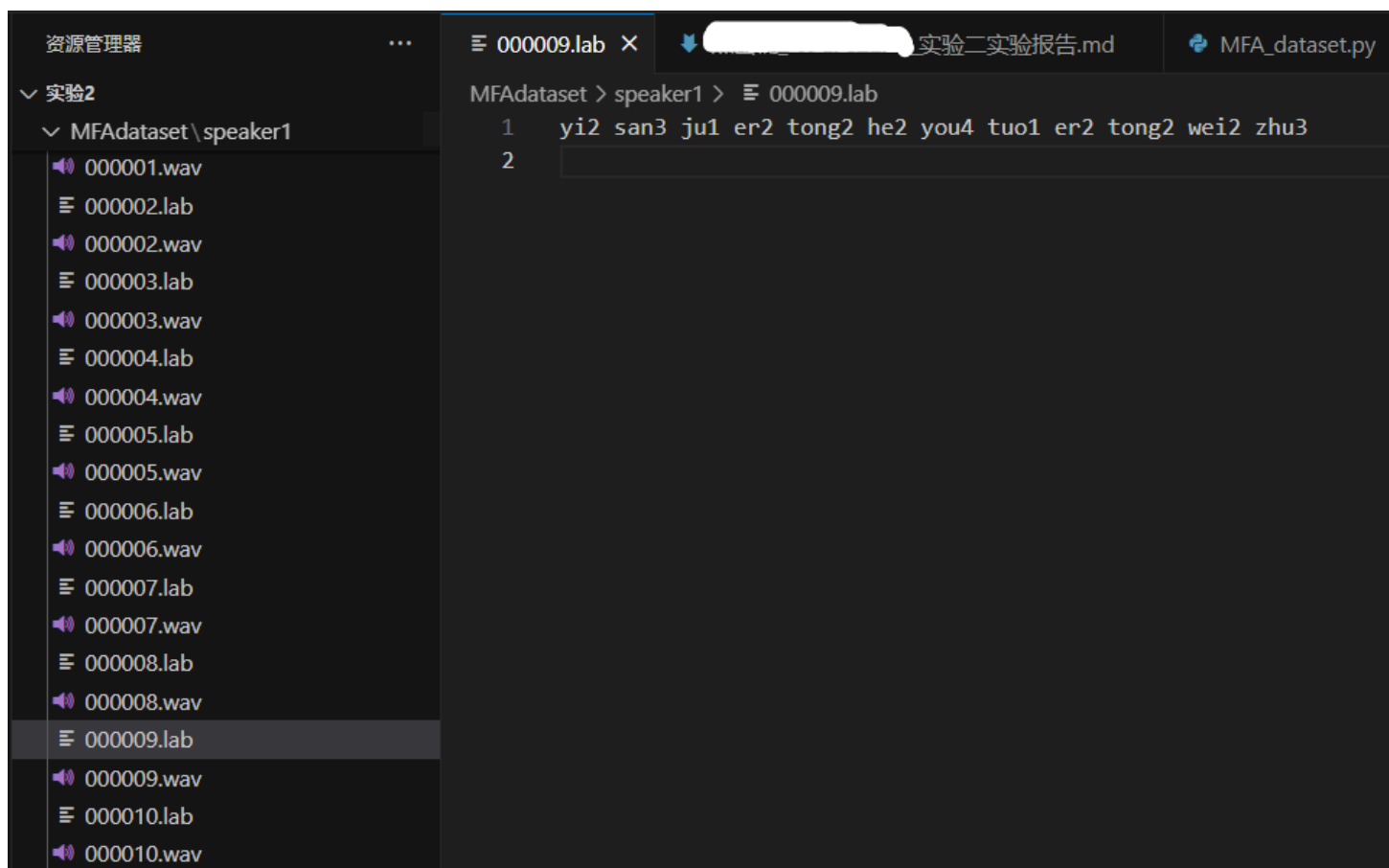


```

• (aligner) PS C:\Users\ysn\Desktop\研究生课内资料\语音处理\实验2> python .\MFA_dataset.py
Finished!

```

生成数据集结构和内容如下所示：



### 3.7（附加题，上限10分）：如何改进对齐结果？这是个开放的问题，需要附上实验，用pratt说明改进后的textgrid比原来的textgrid好即可。

本来想换个最新版的MFA普通话拼音声学模型，但在网上没有找到。一般来说最新预训练模型训练时使用的数据集更多，对齐效果应该会更好。

## 参考文献

- [1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [2] F. Yang, S. Balakrishnan, and M. J. Wainwright, "Statistical and computational guarantees for the baum-welch algorithm," *Journal of Machine Learning Research*, vol. 18, no. 125, pp. 1–53, 2017.
- [3] S. Mehta, E. Szekely, J. Beskow, and G. E. Henter, "Neural hmms are all you need (for high-quality attention-free tts)," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7457–7461, 2021.