# Lyrics Alignment

Hữu Tường Tú, Hồ Anh, Nguyễn Hải Dương, Nguyễn Phúc Tân

Hanoi University of Science and Technology
School of Information and Communication Technology

Hà Nội, 2023

# Summary

# Lyrics Alignment

- Quá trình đồng bộ hóa lời bài hát của một bài hát với âm thanh tương ứng của nó
- Mục tiêu của sự liên kết lời bài hát là tạo ra trải nghiệm liền mạch và trực quan cho người dùng, giúp họ dễ dàng hát theo hoặc học lời cho một bài hát
- Ứng dụng: hệ thống karaoke, phần mềm giáo dục âm nhạc và trò chơi video âm nhạc. Nó cũng có thể được sử dụng để tạo chú thích đóng cho các video âm nhạc hoặc biểu diễn trực tiếp, làm cho nội dung rõ ràng hơn.

# Bài toán

Nhiều người trong chúng ta thích hát theo những bài hát theo cách của các ca sĩ yêu thích trong album, (phong cách karaoke). Để thực hiện nó, chúng tôi có thể cần phải loại bỏ giọng hát của (các) ca sĩ khỏi các bài hát, sau đó cung cấp lời bài hát phù hợp kịp thời với âm thanh đệm. Có nhiều công cụ khác nhau để loại bỏ giọng hát, nhưng rất khó để căn chỉnh lời bài hát với bài hát.
Để giải quyết vấn đề này, chúng ta sẽ xây dựng một mô hình để điều chỉnh lời bài hát với âm thanh âm nhạc

# Yêu cầu

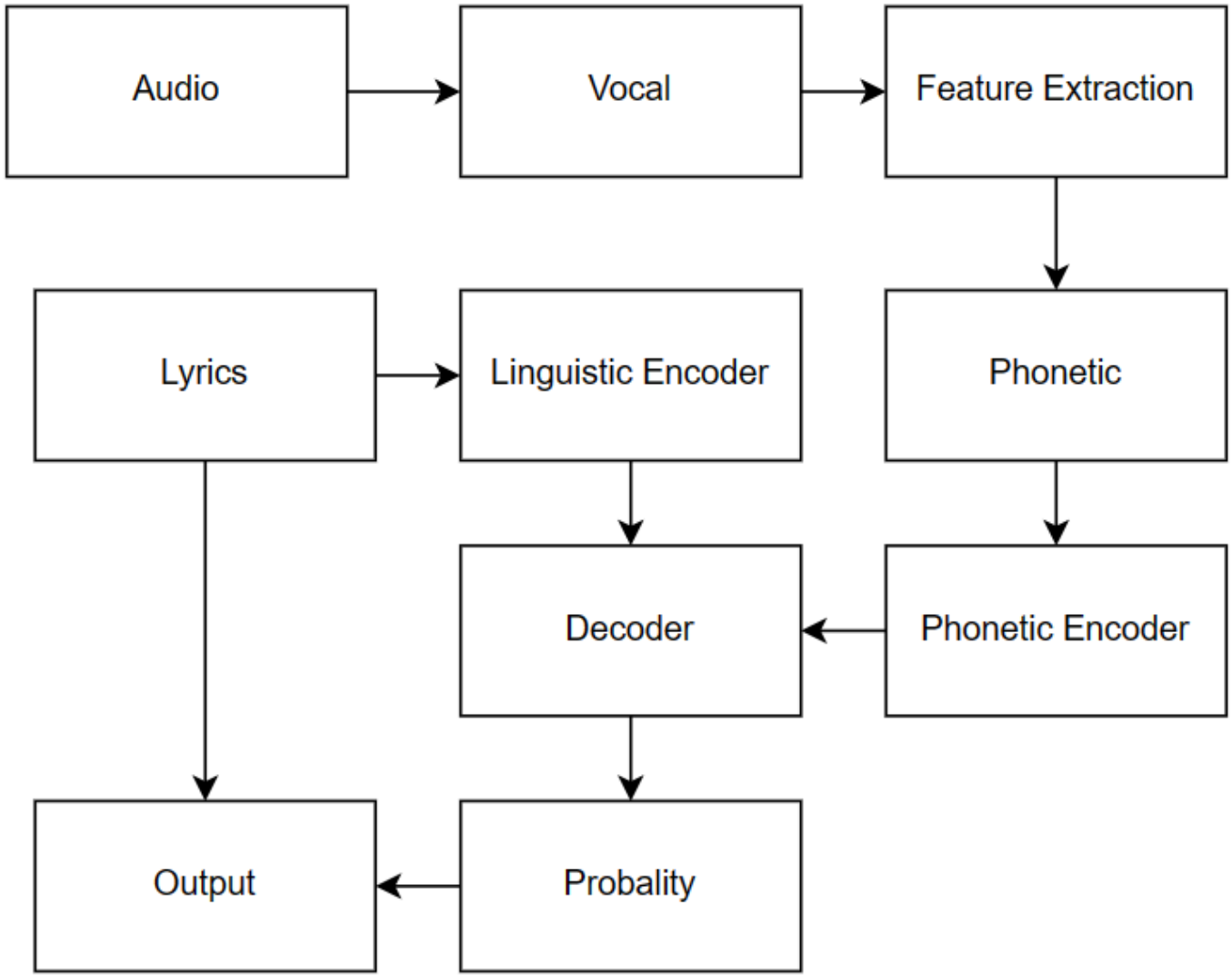
- **Đầu vào**: Một phân đoạn âm nhạc (bao gồm cả giọng hát) và lời bài hát của nó.
- **Đầu ra**: Thời gian bắt đầu và thời gian kết thúc của mỗi từ trong lời bài hát

# Các nghiên cứu liên quan

- wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations
- Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks
- CTC-Segmentation of Large Corpora for German End-to-end Speech Recognition
- Attention is all you need

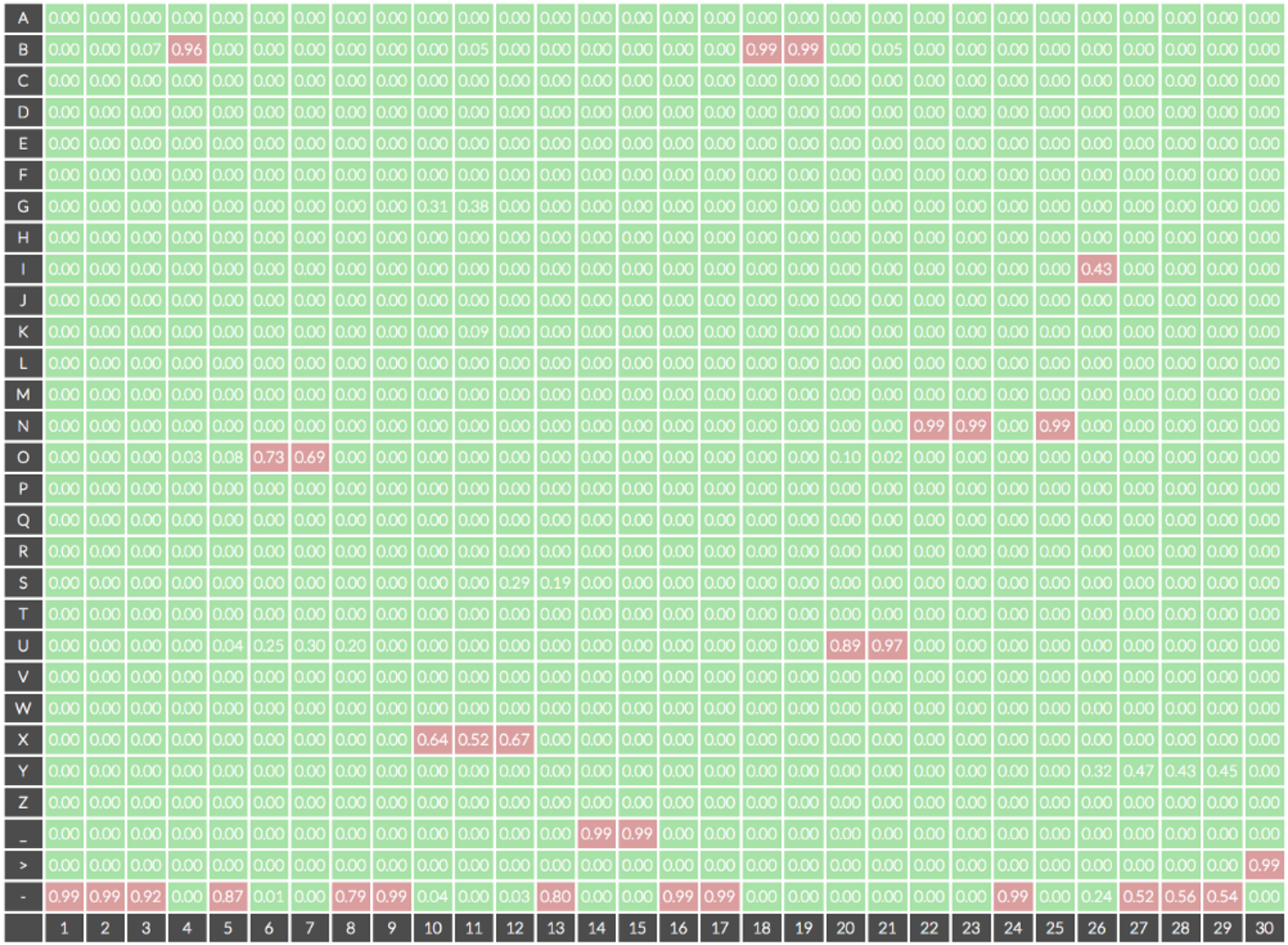
# Kiến trúc tổng quan

## Kiến trúc tổng quan model



# Kiến trúc tổng quan

## Output probability



# Feature Extraction

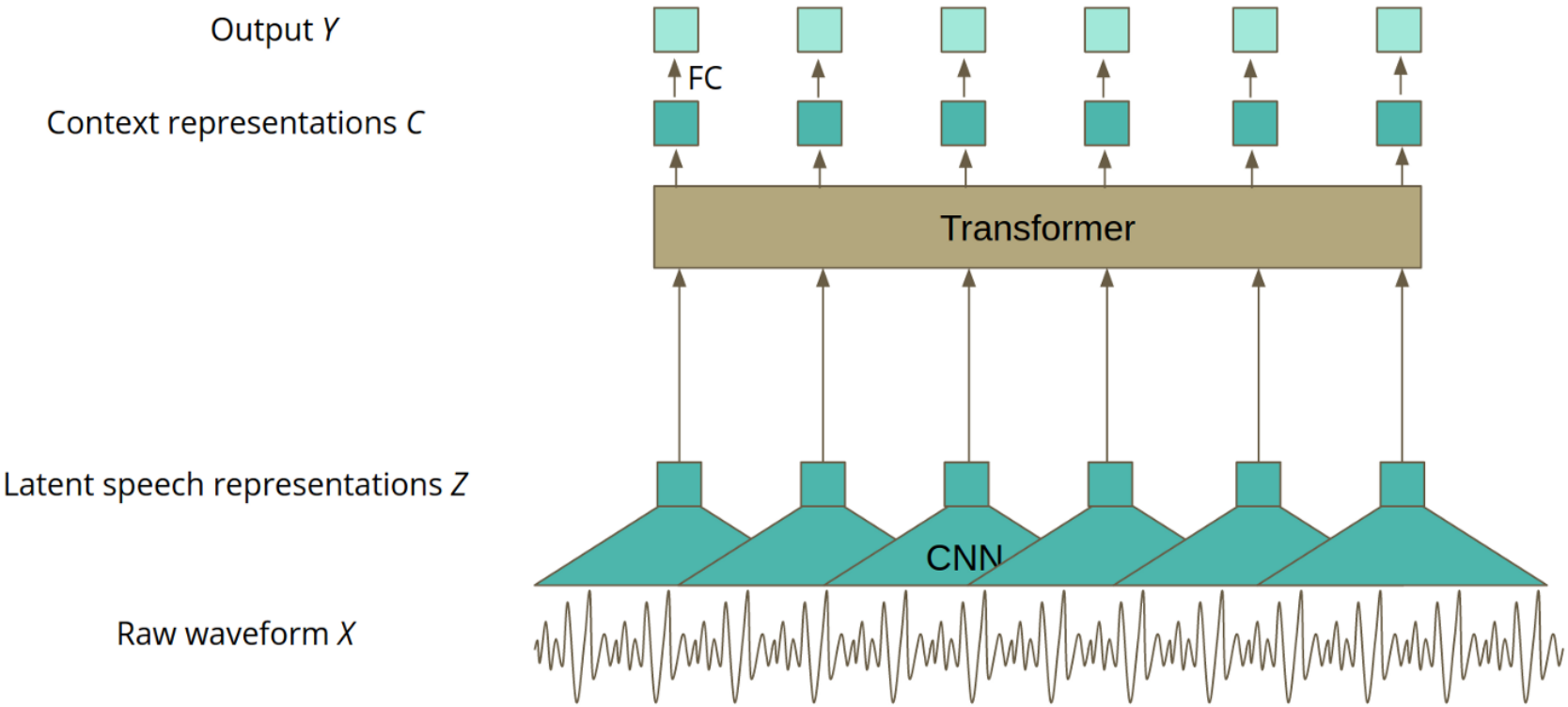
**Input** được chia làm các frame có độ dài 0.02s. Sau đó được đưa qua model Wav2vec2.0 trích xuất được đặc trưng.

**Input:**

**Output** Feature vector kích thước [time/0.02] x 768:

# Feature Extraction

## Kiến trúc Wav2vec2.0 model



# Encoder

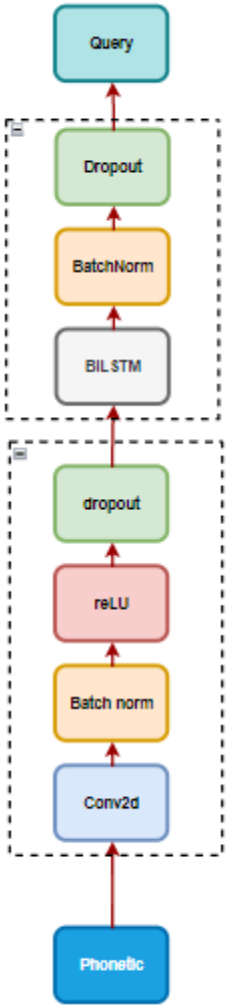



Figure: Phonetic Encoder

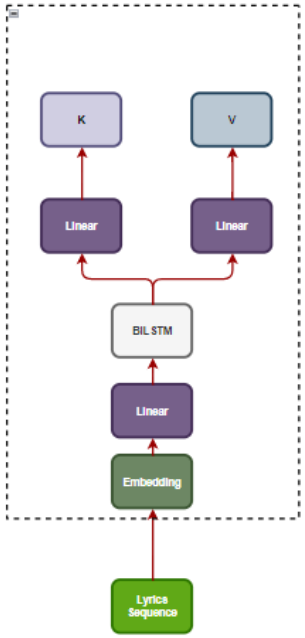
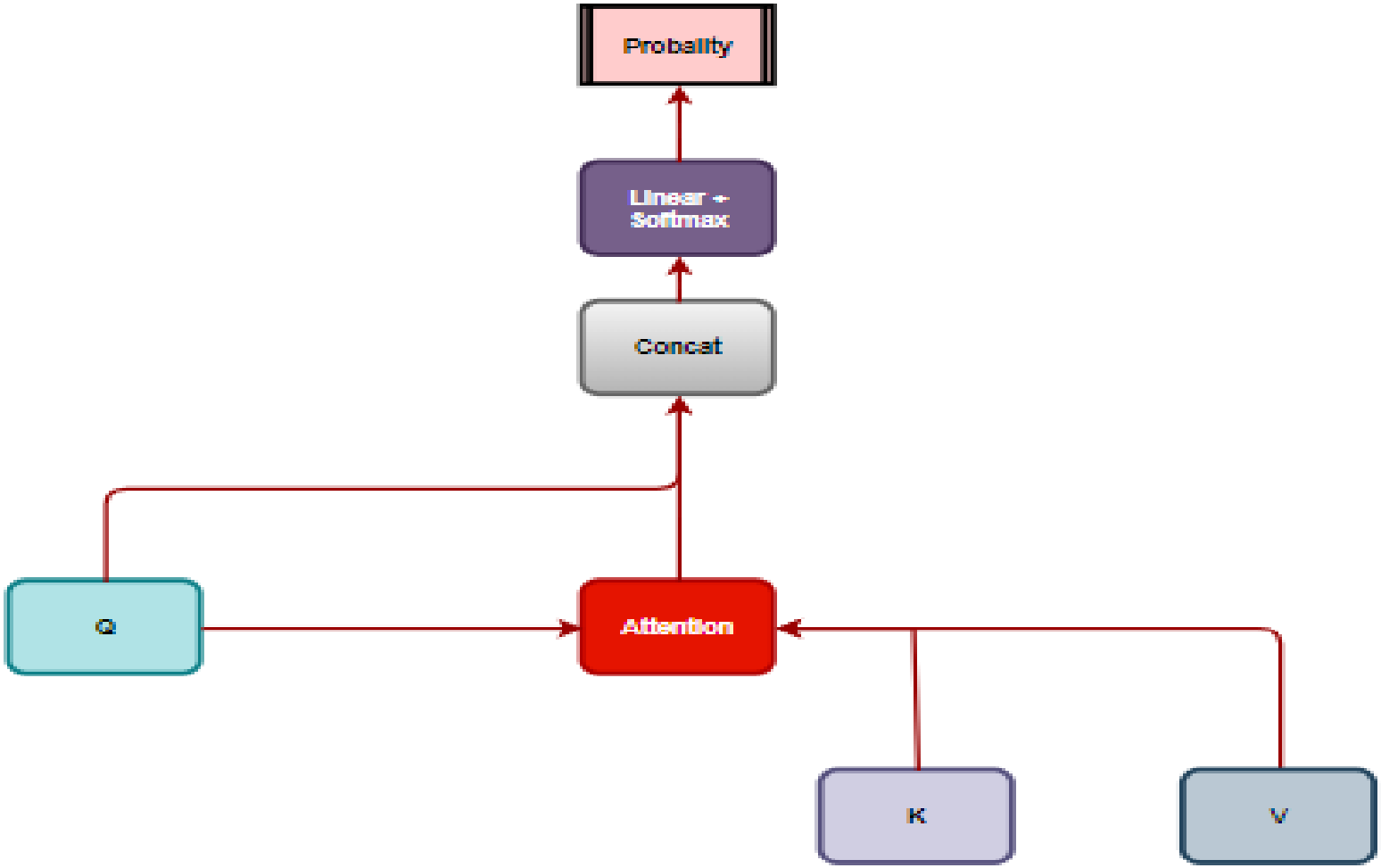



Figure: Linguistic Encoder

,,

# Decoder



# CTC Loss - Training Phase

- Thực thi trên từng time-step của ma trận probability đầu ra.
- S là tập huấn luyện bao gồm các cặp dữ liệu (x, y) - (x là dữ liệu đầu vào, y là groundtruth).
- B là hàm loại bỏ các kí tự trùng cạnh nhau và ký tự trống. Ví dụ:

$$B("-t-oo") = "to"; B(-ttt-oooo-o-) = "too"$$

- $\pi$ là path (mỗi path là một cách chọn ký tự ở một timestamp rồi kết hợp chúng lại với nhau)
- Alignment của một nhãn L là tập các path $\pi$ với độ dài time-step thỏa mãn $B(\pi) = L$
- Xác suất của nhãn y đối với dữ liệu đầu vào x:

$$p(y \mid x) = \Sigma_{\pi \in B^{-1}(y)} p(\pi \mid x)$$

- Loss function:

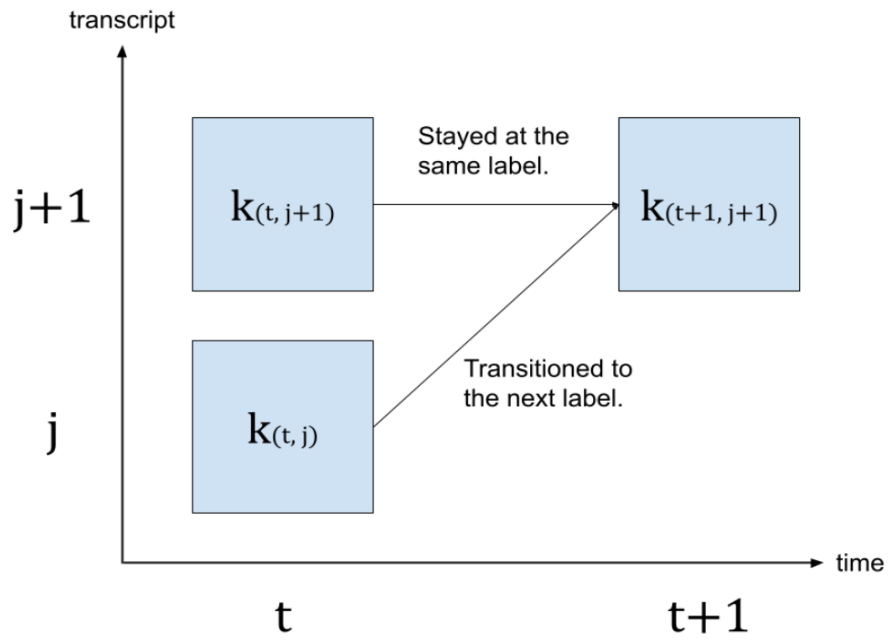
$$Loss(S) = -\sum_{(x,y) \in S} \ln(p(y \mid x))$$

# CTC Forced Alignment

- Từ ma trận probability thu được và lyric của đầu vào tương ứng. Ta sử dụng thuật toán Force Alignment để căn chỉnh thời gian phù hợp cho từng kí tự của lyric.
- Thuật toán sẽ xây dựng ma trận Trellis **k** có 2 chiều, với chiều **t** là chiều thời gian theo từng time-stamp, và chiều **j** là chiều duyệt qua từng kí tự trong lyric (transcript).

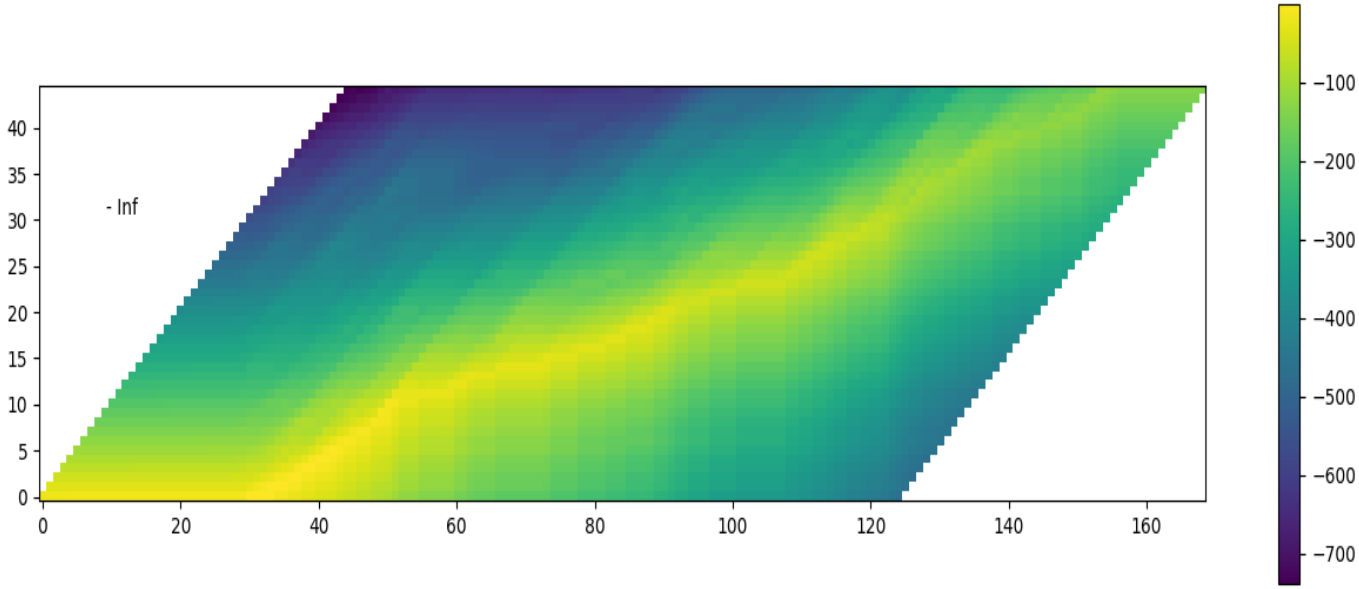
# CTC Forced Alignment

$$k_{t,j} = \begin{cases} \max(k_{t-1,j} \cdot p(blank|t),\ k_{t-1,j-1} \cdot p(c_j|t)) & \text{if } t > 0 \wedge j > 0 \\ 0 & \text{if } t = 0 \wedge j > 0 \\ 1 & \text{if } j = 0 \end{cases}$$




Xây dựng Trellis Matrix

# CTC Forced Alignment

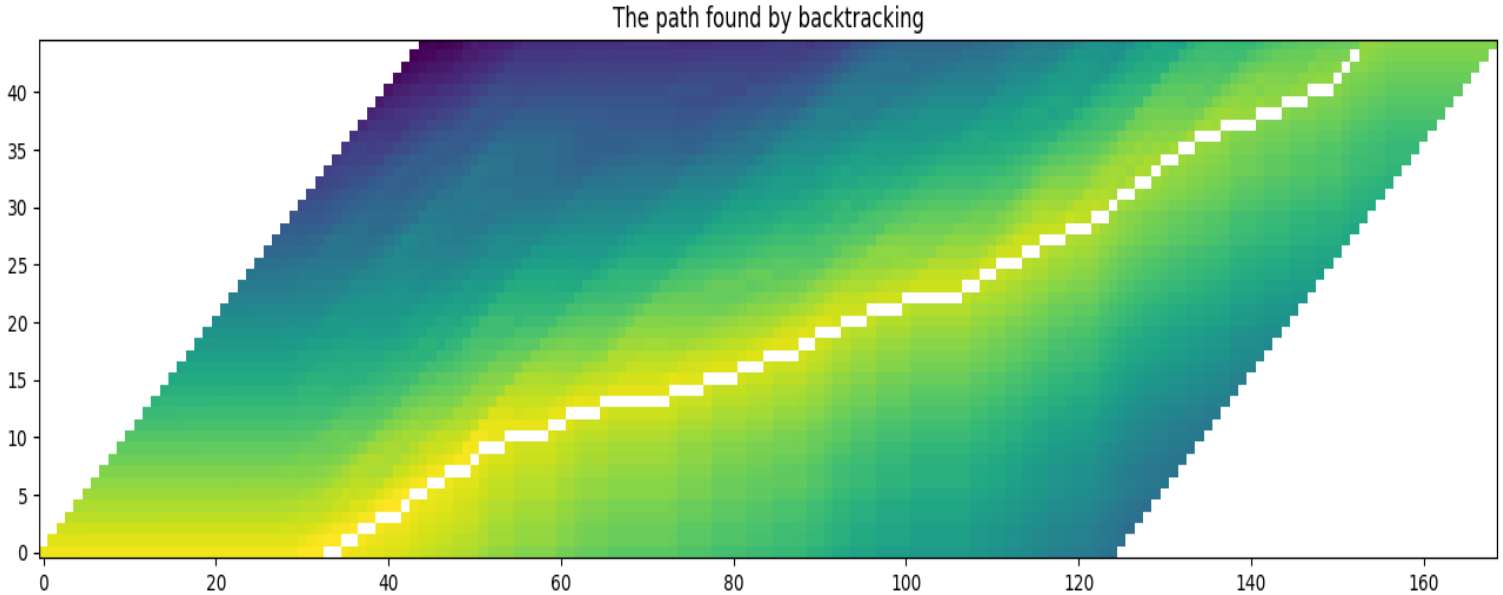



Ma trận Trellis thu được

# CTC Forced Alignment

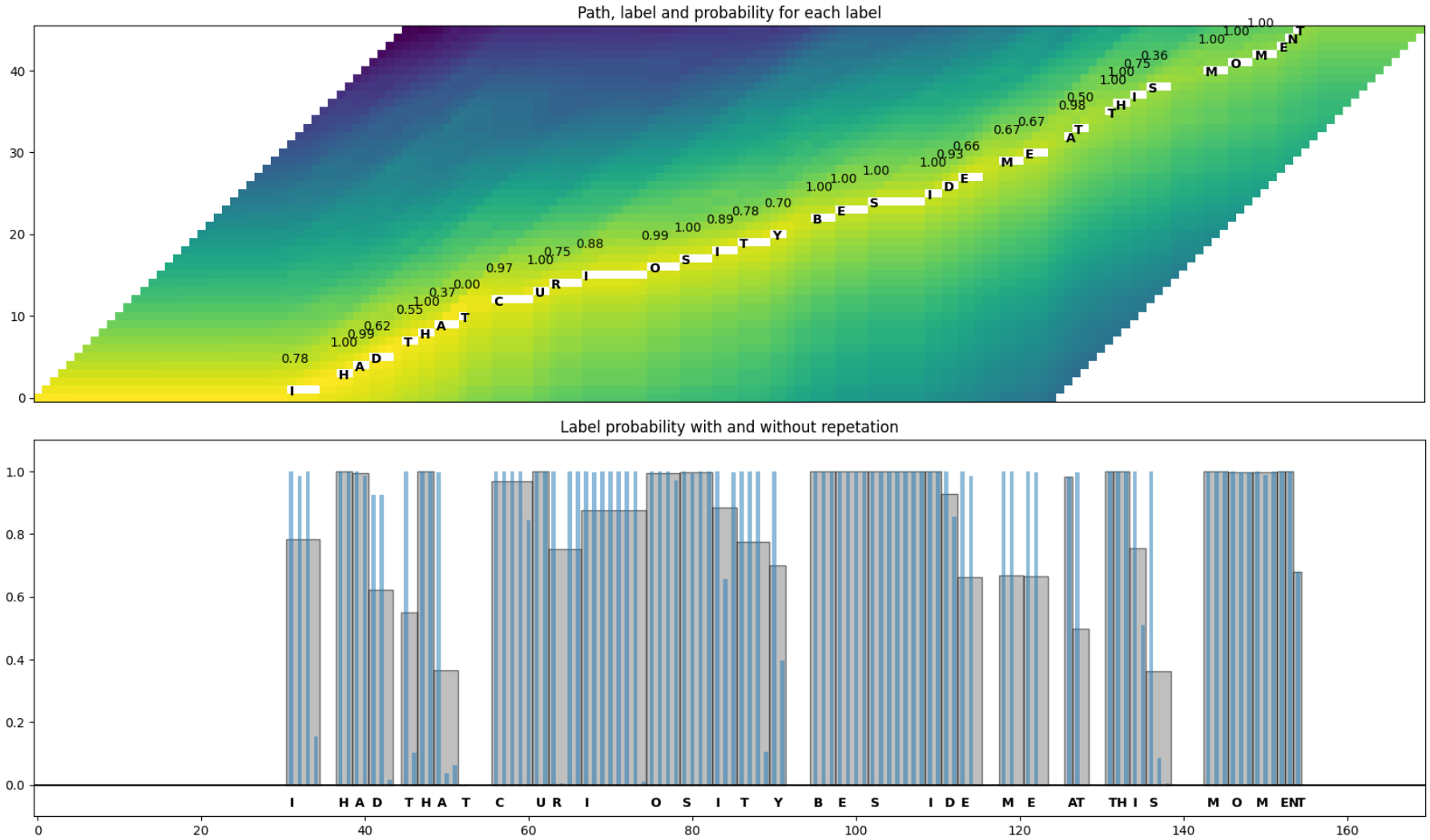
Sau khi thu được ma trận Trellis, ta sử dụng thuật toán backtracking để tìm ra đường đi có xác suất lớn nhất.

$$a_t = \begin{cases} a_{t+1} & k_{t,a_{t+1}} p(blank|t+1) > k_{t,a_{t+1}-1} p(c_{a_{t+1}-1}|t+1) \\ a_{t+1} - 1 & else \end{cases}$$




Kết quả tìm được bởi backtracking

# CTC Forced Alignment



# Dữ liệu

**Dữ liệu training:**

- 1057 phân đoạn âm nhạc từ  480 bài
- Mỗi phân đoạn được cung cấp một âm thanh được định dạng dưới dạng tệp WAV và tệp JSON bao gồm lời bài và khung thời gian được căn chỉnh từng từ như ví dụ trên.

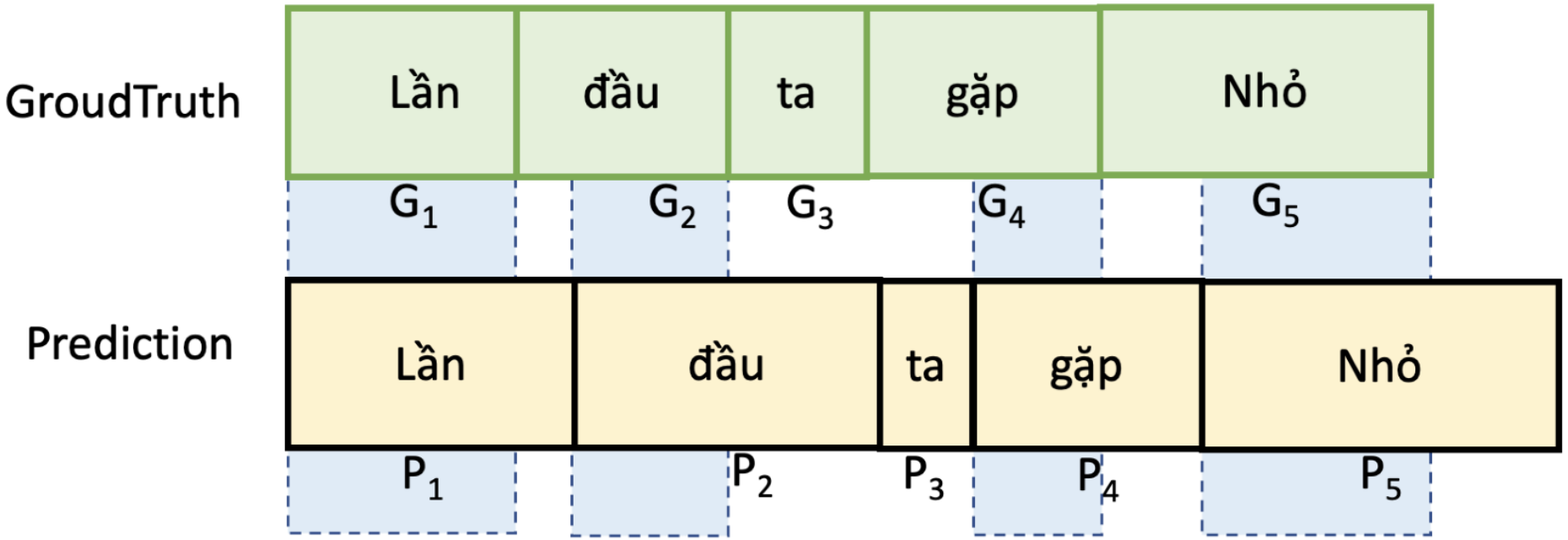
**Dữ liệu testing**

- 264 đoạn nhạc từ 120 bài hát mà không mà không tệp căn chỉnh lời bài hát

# Evaluation Metrics

- Để đánh giá kết quả của dự đoán bằng cách sử dụng Intersection over Union (IoU).
- Với IoU metric, chỉ số càng cao thì càng tốt.

# Intersection over Union (IoU)




Ví dụ của IoU cho prediction and ground truth

# Intersection over Union (IoU)

IoU của prediction và ground truth của audio segment $s_i$ được tính bởi công thức sau:

$$IoU(s_i) = \frac{1}{m} \sum_{j=}^{m} \frac{G_j \bigcap P_j}{G_j \bigcup P_j}.100\% \quad (1)$$

trong đó: $m$ là số lượng các token của $s_i$
IoU trên tất cả n phân đoạn âm thanh là trung bình của IoUs tương ứng của chúng.

$$Final\_IoU = \frac{1}{n} \sum_{i=1}^{n} IoU(s_i) \quad (2)$$

# Các mô hình thực nghiệm

- Wav2vec2.0 finetune layer linear cuối cùng
- Kết hợp wav2vec2.0 + vocal và finetune layer cuối cùng
- Sử dụng Phonetic qua Phonetic Encoder
- Sử dụng Phonetic + Linguistic qua Phonetic Encoder + Linguistic Encoder
- Sử dụng Phonetic + Vocal qua Phonetic Encoder
- Sử dụng Phonetic + Vocal + Linguistic qua Phonetic Encoder + Linguistic Encoder

# Kết quả đánh giá

| Model | Result |
|---|---|
| Phonetic Finetune OnlyLinear | 45.87 |
| Phonetic Finetune OnlyLine + Vocal | 46.21 |
| Phonetic | 50.22 |
| Phonetic + Linguistic | 50.32 |
| Phonetic + Vocal | 49.98 |
| Phonetic + Linguistic + Vocal | 50.26 |

# Hướng cải tiến

Sử dụng các pretrained ASR khác để extract feature phonetic, sau đó ta có thể fusion các mô hình xác suất lại với nhau để tránh overfit