



# MLDS **FINAL PROJECT**

廖宜修 沈昇勳 吳彥諶

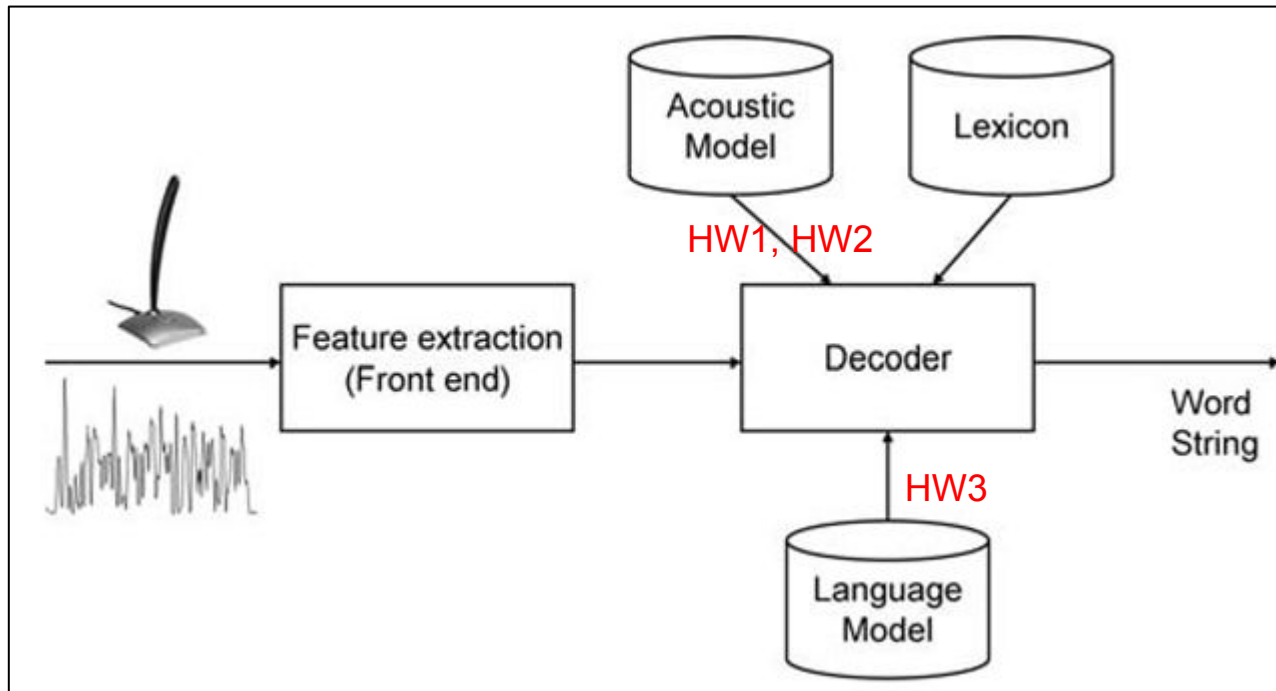
# Outline

- Goal
- N-best Decoding
- Lexicon (Phones to Words)
- Language Model Rescoring
- Homework Description
- Additional Rules
- Grading

# Goal

Use all homeworks to build an Automatic Speech Recognition(**ASR**) system.

In another word, Input voice and output word.





# From Viterbi to N-Best

Yen-Chen Wu

# Why Do We Need N Best?

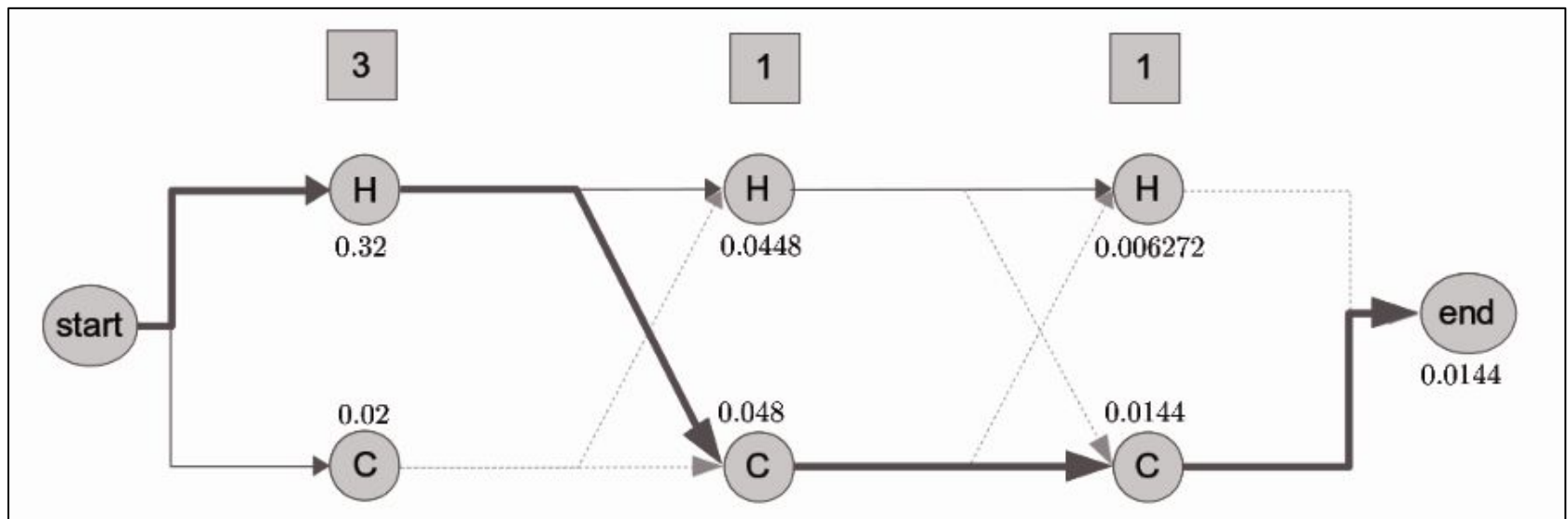
- N Best
  - The most possible N paths
- Combine with lexicon and language model (rescore)
- Find the most possible path after rescoring
- Viterbi provides the best sequence
  - We only need to modified the Viterbi Algorithm

# Example

- Best sequence:
  - aaabbccccc      score: 0.92
- N Best sequence: (N = 3)
  - aaabbccccc    score: **0.92**   re-score: 0.77
  - aabbbccccc    score: 0.88   re-score: **0.79**
  - aaddccccc    score: 0.83   re-score: 0.76
- Then we pick second sequence as output!

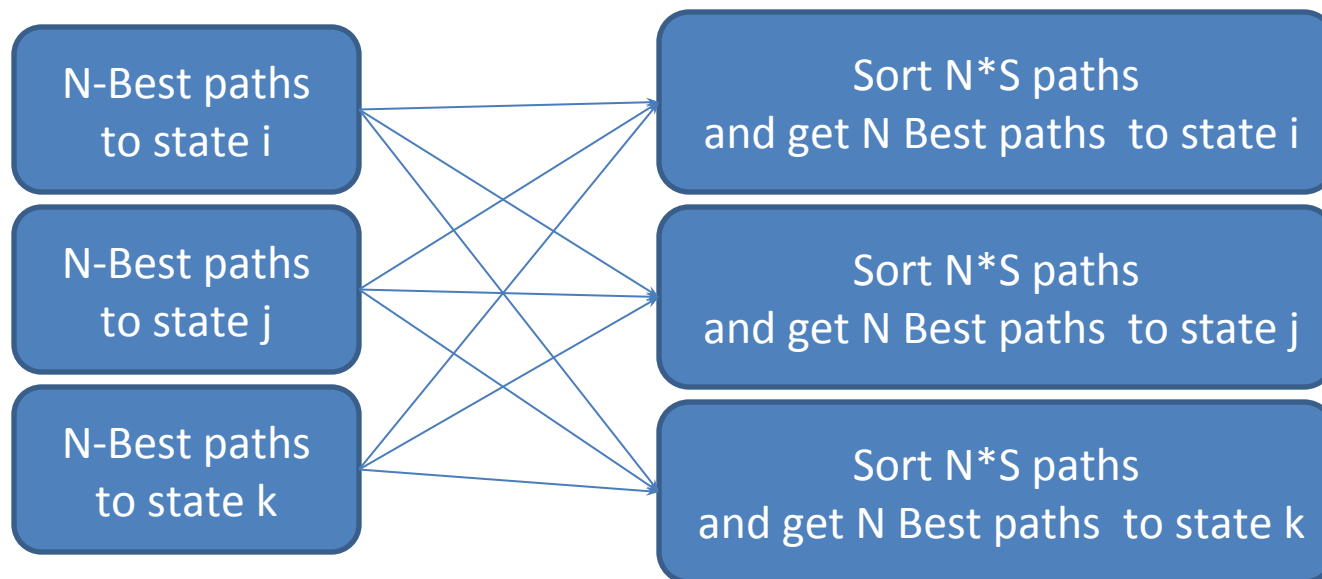
# Review: Viterbi

- Keep the most possible path in each state
- Complexity :  $O(L * S^2)$ 
  - L : sequence length
  - S : number of states



# N Best

- Keep the most possible N path in each state
- Sort  $N \times S$  paths and get N Best paths at final layer
- Complexity :  $O(N \times L \times S^2)$







# Lexicon (Phones to Words)

Yi-Hsiu Liao

# What's Lexicon

We already have phone sequence. The next step is using **lexicon** to translate from phones to words.

lexicon



```
divorce /d ax v ao1 r s/  
divorced /d ax v ao1 r s t/  
divorcee /d ax v ao2 r s ey1/  
do /d uw1/  
doctor /d aa1 k t axr/  
doctors /d aa1 k t axr z/  
doctrine /d aa1 k t r ax n/  
documented /d aa1 k y uw m eh2 n t ix  
documents /d aa1 k y uw m ax n t s/  
dodging /d aa1 jh ix ng/  
does /d ah1 z/  
doesn't /d ah1 z en t/
```

/ hh w aa t sil d uw sil y uw sil th ih ng k /  
what do you think?

# Lexicon

Sometimes there is more than one way to decode a phone sequence.

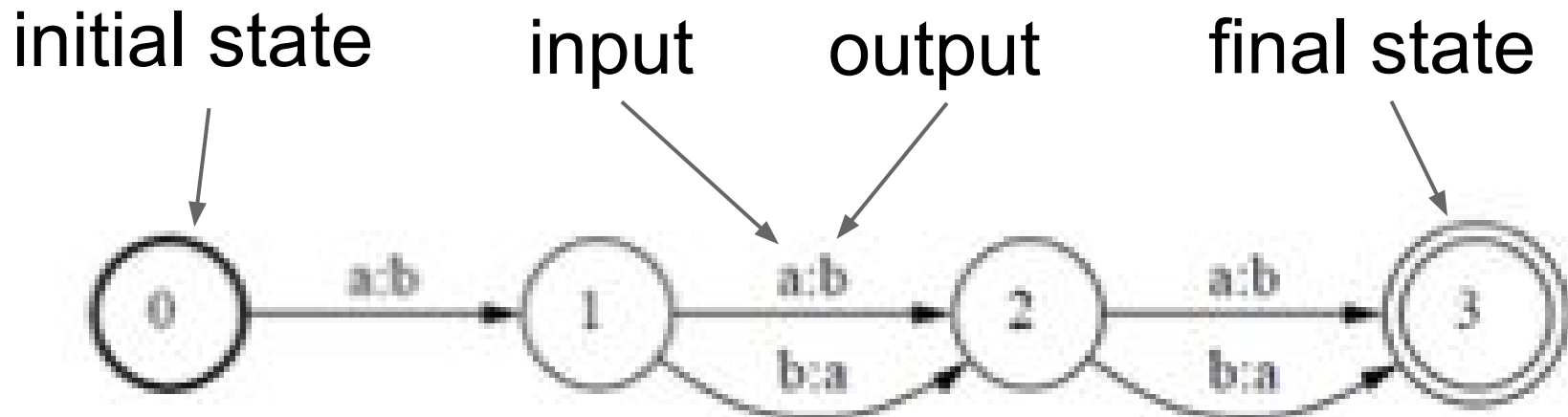
ex. 呻吟 vs 聲音, for vs four, too vs two

We can use **WFST** to decode a phone sequence for all possible combinations.

# Finite State Transducer

FST is a finite state machine with two tapes:  
an **input tape** and an **output tape**.

–Transduce any legal input string to another output string, or reject it

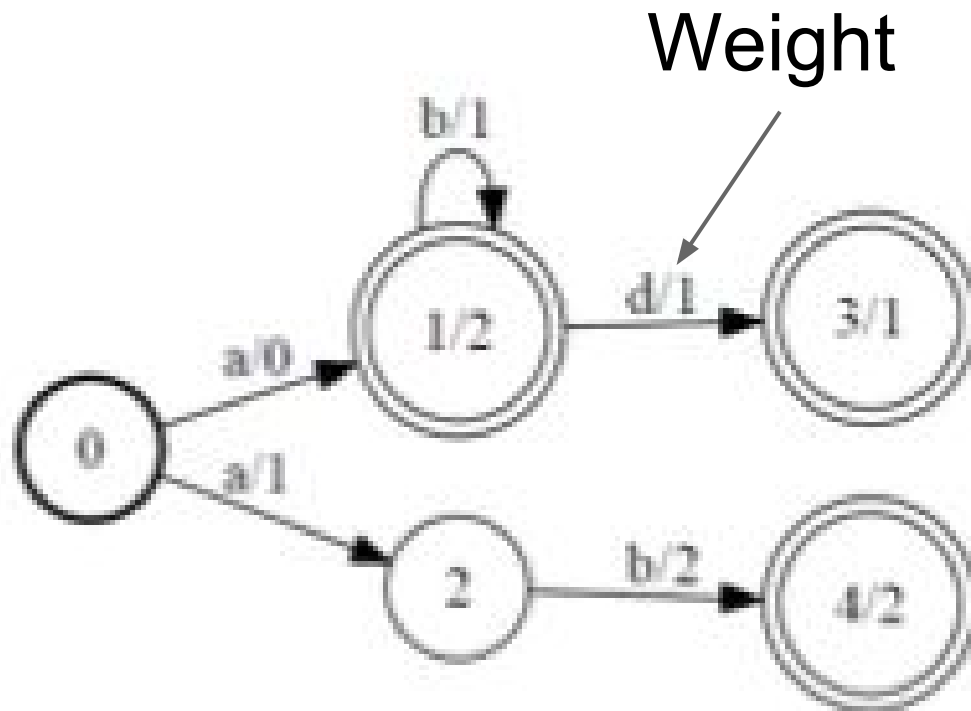


{aaa, aab, aba, abb} -> {bbb, bba, bab, baa}

# Weighted Finite State Machine

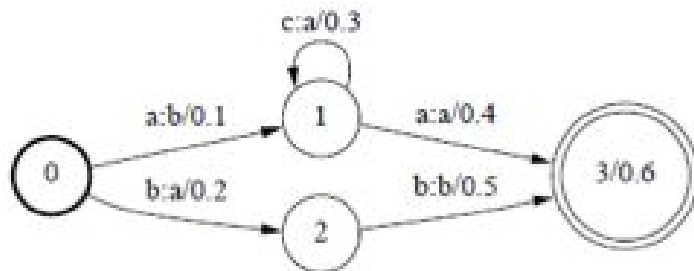
Through states (0, 1, 1); cost is  $(0+1+2) = 3$

Through states (0, 2, 4); cost is  $(1+2+2) = 5$



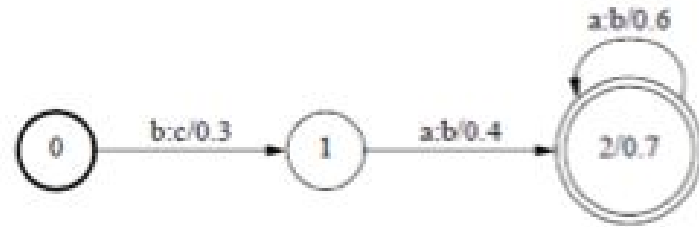
# WFST operations

## Composition



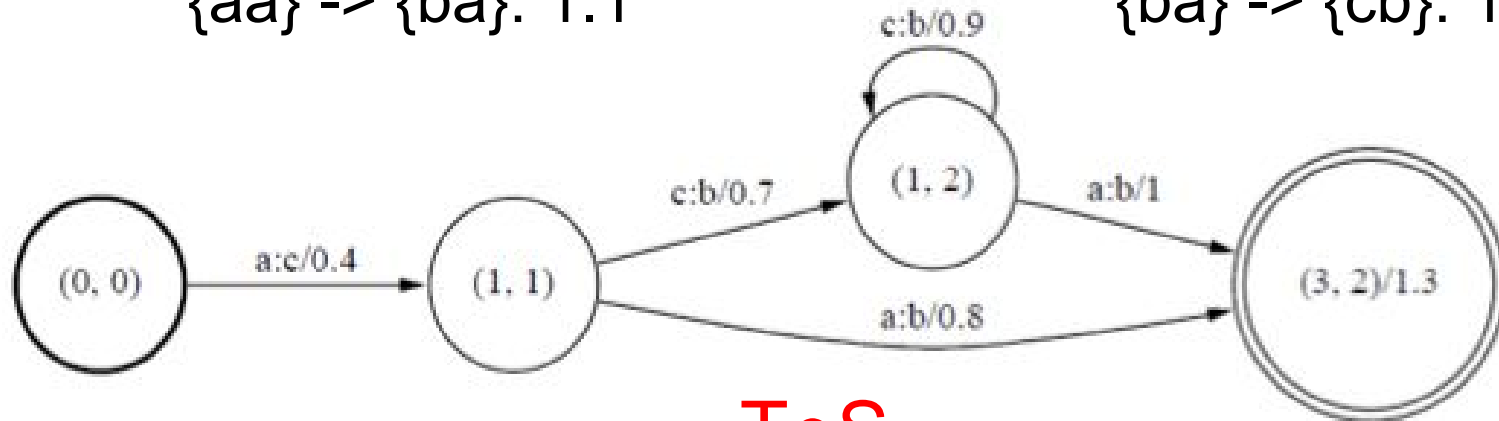
T

{aa} → {ba}: 1.1



S

{ba} → {cb}: 1.4



T o S

{aa} → {cb}: 2.5

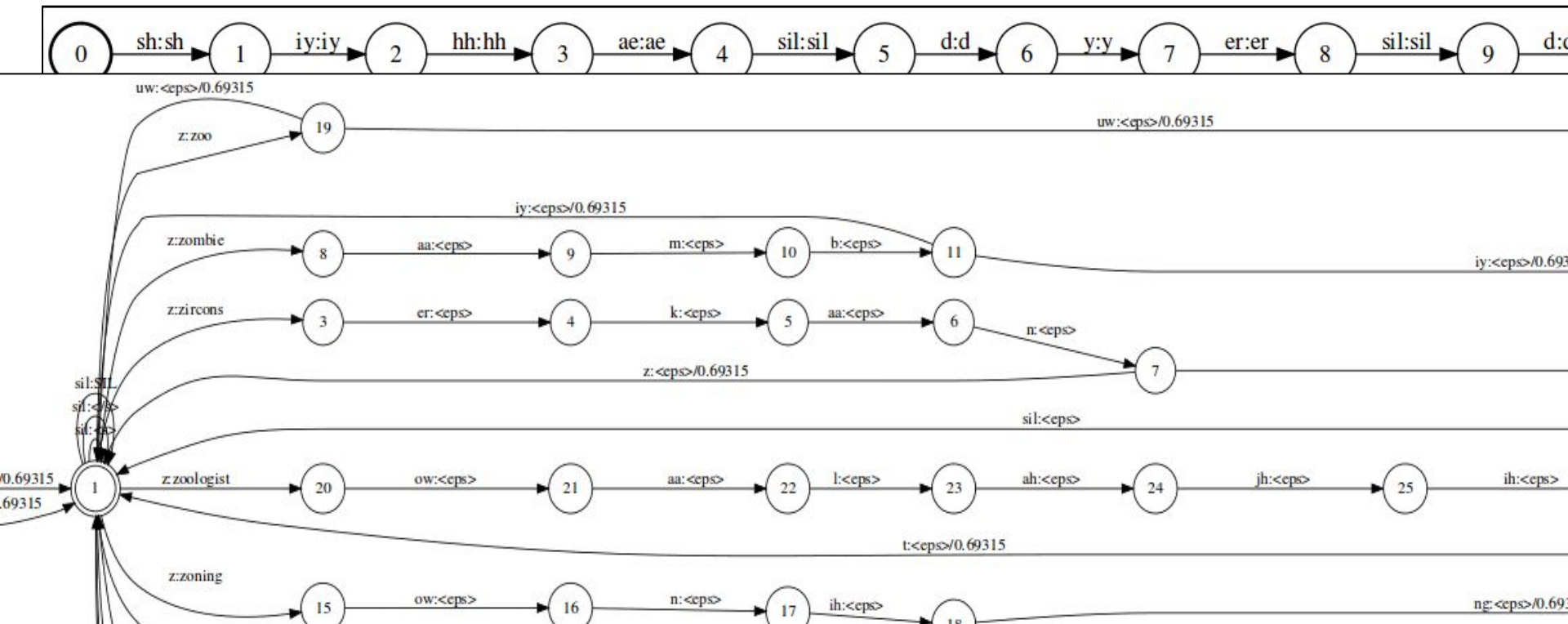
# Lexicon and WFST

Lexicon can be viewed as a WFST.

Phone sequence can be rewritten in a WFST form, too.

The translation process is actually a composition of 2 WFSTs.

# phone sequence WFST



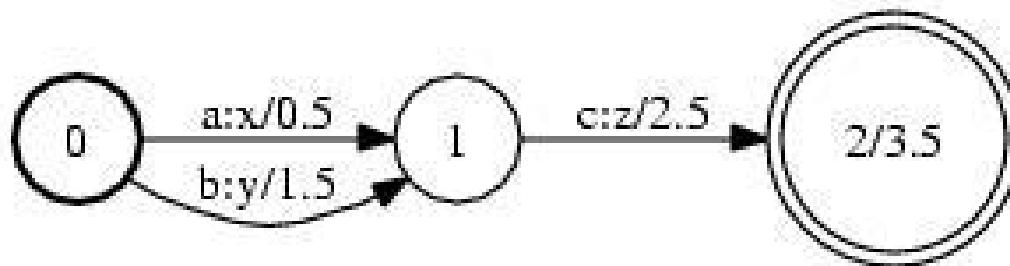


# FST example

`fstcompile --isymbols=isyms.txt`

`--osymbols=osyms.txt text.fst binary.fst`

----- text.fst ----- 0 1 a x .5 0 1 b y 1.5 1 2 c z 2.5 2 3.5	----- isyms.txt ----- <eps> 0 a 1 b 2 c 3	----- osyms.txt ----- <eps> 0 x 1 y 2 z 3
---	---	---



**binary.fst**

# OpenFST and package

## 1. Install OpenFST

<http://www.openfst.org/twiki/bin/view/FST/WebHome>

## 2. Download Lexicon and script package

[https://github.com/Liao-YiHsiu/Lexicon\\_WFST](https://github.com/Liao-YiHsiu/Lexicon_WFST)



# Language Model Rescoring

Sheng-Syun Shen

# Language Model Rescoring

You may get n-best word sequences for every example. Now you need to rescore the n-best sequences using language model, and then report the best one.

# Language Model Rescoring

How to calculate the sentence sequence probability via language model?

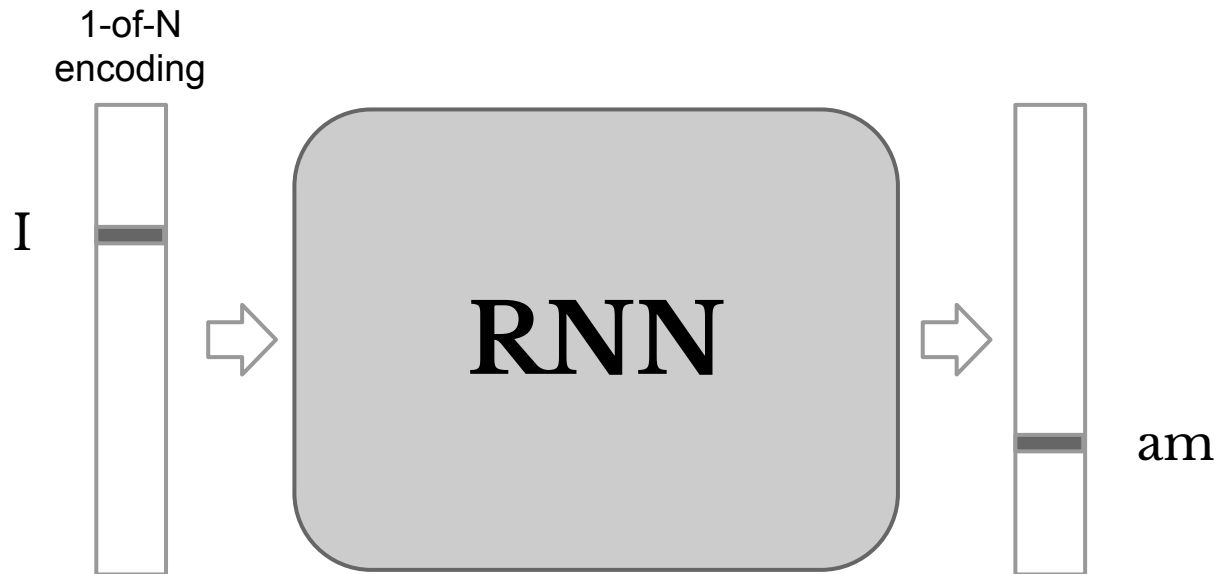
- You can use the recurrent neural network from HW3
- Or any other toolkits, such as [RNNLM](#)

# Probability Calculation

Calculate the bi-gram probability via RNN.

$$\begin{aligned} &P(\text{"I am a student"}) \\ &= P(\text{"am"} \mid \text{"I"}) * P(\text{"a"} \mid \text{"am"}) * P(\text{"student"} \mid \text{"a"}) \\ &\quad * P(\text{"</s>"} \mid \text{"student"}) \end{aligned}$$

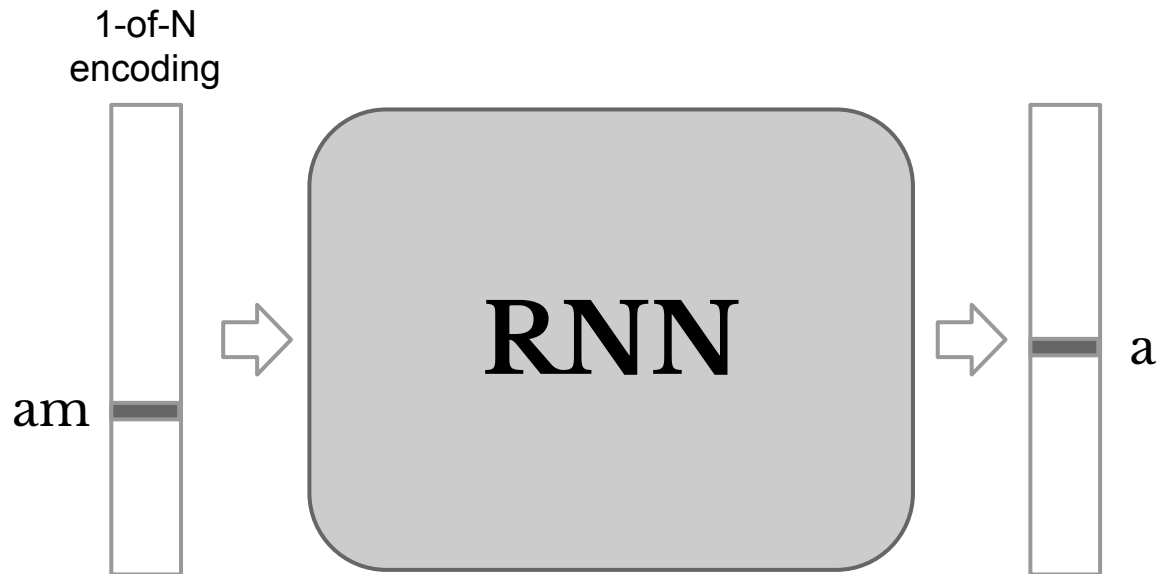
# Probability Calculation



You can get the probability of “am”,  
which is the bi-gram probability.

$$P(\text{“am”} \mid \text{“I”})$$

# Probability Calculation

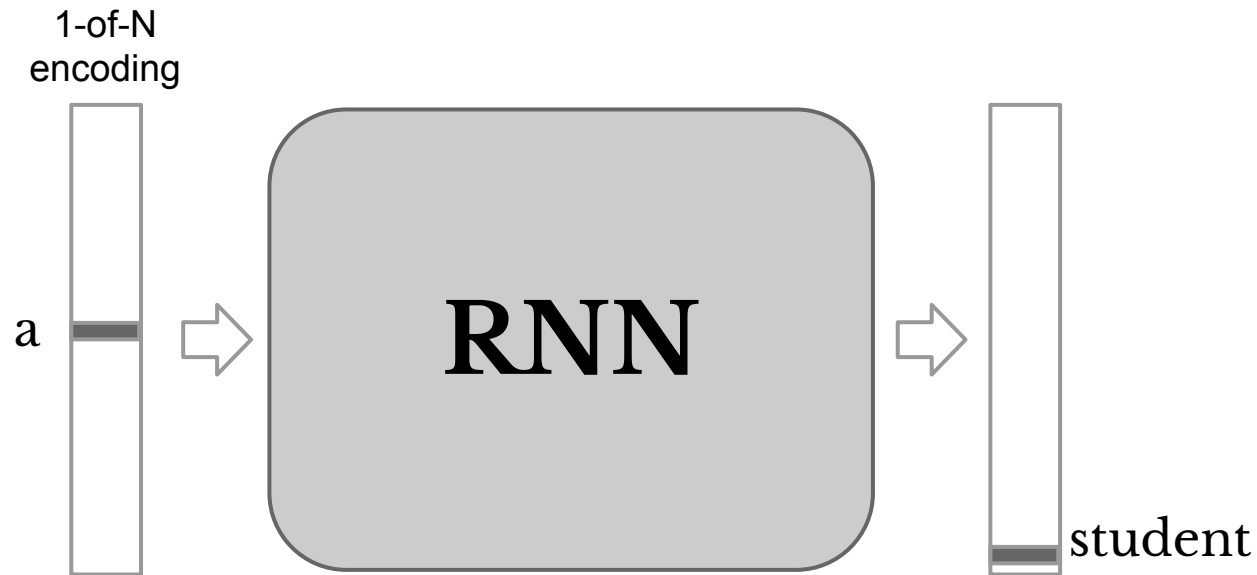


You can get the probability of “a”,  
which is the bi-gram probability.

$$P(\text{“a”} \mid \text{“am”})$$



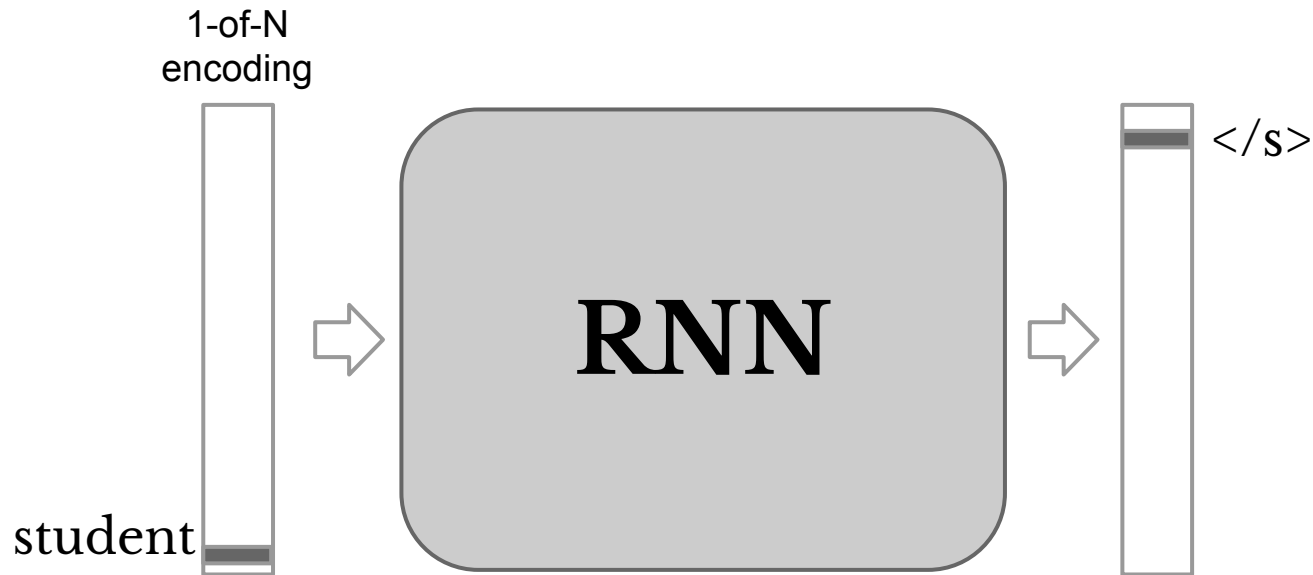
# Probability Calculation



You can get the probability of “student”, which is the bi-gram probability.

$$P(\text{“student”} \mid \text{“a”})$$

# Probability Calculation



You can get the probability of “</s>”,  
which is the bi-gram probability.

$$P(\text{“</s>”} \mid \text{“student”})$$

# N-best rescoring

For an input utterance, you need to calculate the probability of each n-best word sequence

$P(\text{"I am a student"})$

Highest!!!

$P(\text{"I an a student"})$

$P(\text{"I am a stood aunt"})$

$P(\text{"I am stood"})$



# Homework Description

# Requirement

- **Cascade** all previous works into a whole system
- **Fine tune/improve** your previous works using toolkits.
- You may use **any** library and toolkit to further improve your results. **Online ASR speech API is forbidden.**
- You should focus on manifold **algorithms** you want to apply or state-of-the-art **techniques** instead of implementation details.
  - ex. LSTM, RNNLM, CNN... etc.

# 轉換中文字

Due to the evaluation criterion in Kaggle is character-wise, you have to transform your output words into Chinese characters.

#Note : Our lexicon only supports lower-case vocabulary.

breast	蝓	惠訴
breath	賁	
breathed		
breathlessly		
breed	鸛	完答
breeze	復	
brick	欣	
bricks	楯	
bridal	沓	
bride	豈	
bride's	籊	
bridge	岬	
bridges	巖	
brief	詛	
briefly	鉈	
bright	留	
brighter		
brightly		
timit.chmap		

# Data Format

Same as in HW1 and HW2.

Note: the **testing dataset** has changed. Use new testing dataset.

```
.
|-- conf
|   |-- phones.60-48-39.map
|   |-- timit.chmap
|   `-- timitdic.txt
|-- fbank
|   |-- test.ark
|   `-- train.ark
|-- label
|   `-- train.lab
|-- mfcc
|   |-- test.ark
|   `-- train.ark
|-- sentence
|   `-- train.set
`-- wav
    |-- fadg0_si1279.wav
    |-- fadg0_si1909.wav
    |-- fadg0_si649.wav
    |-- fadg0_sx109.wav
    |-- fadg0_sx19.wav
    |-- fadg0_sx199.wav
    |-- fadg0_sx289.wav
```

```
makb0_sx26,Most young rise early every morning.
makb0_sx296,The two artists exchanged autographs.
makb0_sx386,Which church do the Smiths worship in?
makr0_si1352,As such it acts as an anchor for the people
makr0_si1982,Two cars came over a crest, their chrome an
makr0_si722,This changes the formula to an equation.
makr0_sx182,Thick glue oozed out of the tube.
makr0_sx272,Allow each child to have an ice pop.
makr0_sx362,The annoying raccoons slipped into Phil's ga
makr0_sx452,In developing film, many toxic chemicals are
makr0_sx92,A large household needs lots of appliances.
mapv0_si1293,These curves were derived by an analysis of
mapv0_si1923,What did you mean by that rattlesnake gag?
mapv0_si663,Satellites, sputniks, rockets, balloons; wha
mapv0_sx123,A screwdriver is made from vodka and orange
mapv0_sx213,The news agency hired a great journalist.
mapv0_sx303,The bluejay flew over the high building.
mapv0_sx33,Coconut cream pie makes a nice dessert.
mapv0_sx393,She uses both names interchangeably.
sentence/train.set
```

# Data Format

m	m	m
n	n	n
ng	ng	ng
nx	n	n
ow	ow	ow
oy	oy	oy
p	p	p
pau	sil	sil
pcl	cl	sil
q		
r	r	r
s	s	s
sh	sh	sh
t	t	t
tcl	cl	sil
th	th	th
uh	uh	uh
uw	uw	uw
ux	uw	uw

conf/phones.60-48-39.map

breast	蜻	
breath	貫	
breathed		惠訴
breathlessly		
breed	鶴	
breeze	復	
brick	欣	
bricks	楯	
bridal	沓	
bride	豈	
bride's	籊	
bridge	岬	
bridges	巖	
brief	詛	
briefly	鉦	
bright	呂	
brighter		完答
brightly		

timit.chmap

above /ax b ah1 v/  
 abruptly /ax b r ah1 p t l iy/  
 absences /ae1 b s ix n s ix z/  
 absent /ae1 b s ix n t/  
 absolute /ae2 b s ax l uw1 t/  
 absolution /ae2 b s ax l uw1 sh ix n/  
 absorbed /ax b s ao1 r b d/  
 absorption /ax b s ao1 r p sh ix n/  
 absurd /ax b s er1 d/  
 absurdly /ax b s er1 d l iy/  
 abyss /ax b ih1 s/  
 academic /ae2 k ix d eh1 m ih k/  
 accelerating /ae k s eh1 l axr ey2 t ix ng/  
 accelerometer /ae k s eh2 l axr aa1 m ax t axr/  
 accelerometers /ae k s eh2 l axr aa1 m ax t axr z/  
 accent /ae1 k s eh2 n t/  
 accept /ae k s eh1 p t/  
 acceptance /ae k s eh1 p t ix n s/  
 accepted /ae k s eh1 p t ix d/  
 access /ae1 k s eh2 s/  
 conf/timitdic.txt




# Upload File Format

```
id,sequence
fadg0_si1279,楯畧剗楯畧剗楯畧剗
fadg0_si1909,扯靠塏衢郤酹裴扯靠
fadg0_si649,悱瘵煬簋鵠駢佷駢從諒
fadg0_sx109,畫蔚諤马矚括遇裴嬪畫
fadg0_sx199,鉤譟车篁吟瘖噉鉤譟车
fadg0_sx289,閤園畧瓴柙塏閤園畧瓴
fadg0_sx379,紉衣鍾漚遇紉衣鍾漚遇
faks0_si1573,豐給讓怪諤饒諤豐箒
```

# Submission (Kaggle)

- Your **predict.csv**.
- Twice a day.
- 50 % public score and 50% private score
- Evaluate by **Edit distance**.

<https://inclass.kaggle.com/c/leehungyi-gi-bong>

#	Δ0h	Team Name	Score ?	Entries	Last Submission UTC (Best - Last Submission)
		Empty baseline	8.28411		

# Submission (Ceiba)

Upload your report to ceiba.

Use **PDF** file format.

You don't need to upload other files.



# **Additional Rules**

(Weekly Bonus)

# Additional Rules (Weekly Bonus)

- Every Friday before 23:59, every group can explain the detail of your method and setting for your best uploaded score
  - Which model? What features & training parameters?
  - Bonus **2 points** would be granted if properly described
- 1st week 6/05: <https://goo.gl/wM6rrX>
- 2nd week 6/12: <https://goo.gl/0dpk9m>
- 3rd week 6/19: <https://goo.gl/eUNwyU>
- 4th week 6/26: <https://goo.gl/coMTIx>

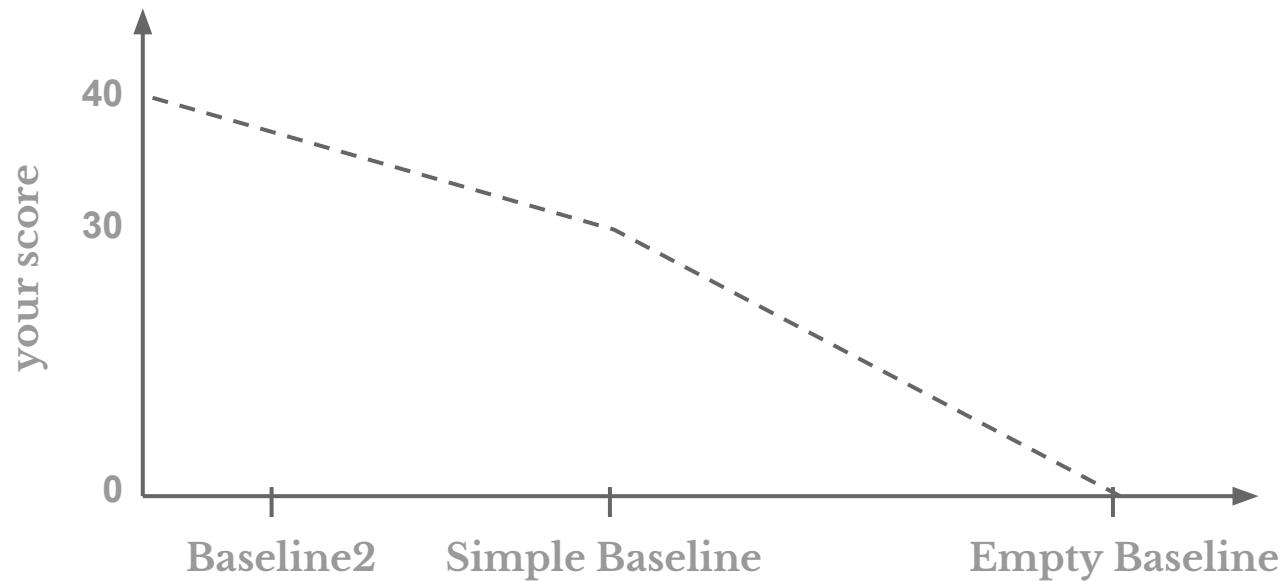


# Grading

# Grading - Kaggle Accuracy(40%)

- Empty baseline
- Simple Baseline in Kaggle (released day 7th)
  - Once achieve this baseline, you can get partial credit (30%) in this part.
- Baseline2 in Kaggle (released day 14th)
  - Once achieve this baseline, you can get full credit (40%) in this part.

# Grading - Kaggle Accuracy(40%)





# Grading

- Kaggle Accuracy 40%
- Report 60%
- Bonus on Kaggle Leaderboard
  - First Place 15%
  - First Runner-up 10%
  - Second Runner-up 5%
- Top-10 in Kaggle Leaderboard wins the ticket to the **final stage**.

# Grading - Report

- Report (60%)
  - Group Information, Division of teamwork
  - Data Preprocessing/Flow Chart/**Algorithm**
  - What have you done?
    - technique, trick. Not implementation details
  - Experiments and **Results**
    - **including charts & figures & comparsons.**
  - No more than 4 A4 pages with font size 12
    - 包含封面封底(if any),reference
  - **PDF** format, or you will get **zero credit**
  - 報告只能以**中文書寫(專有名詞除外)**



**Deadline 6/29 23:59**

**No delay is allowed.**

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

**No delay is allowed.**

No delay is allowed.

No delay is allowed.

# Final stage

We will have a final project presentation on **July 3rd**. Top-**10** gets the opportunity to present what you done in this final project. A **voting** will be held to decide which team is the best team.

First place: **50%**

First Runner-up: **30%**

Second Runner-up: **10%**

