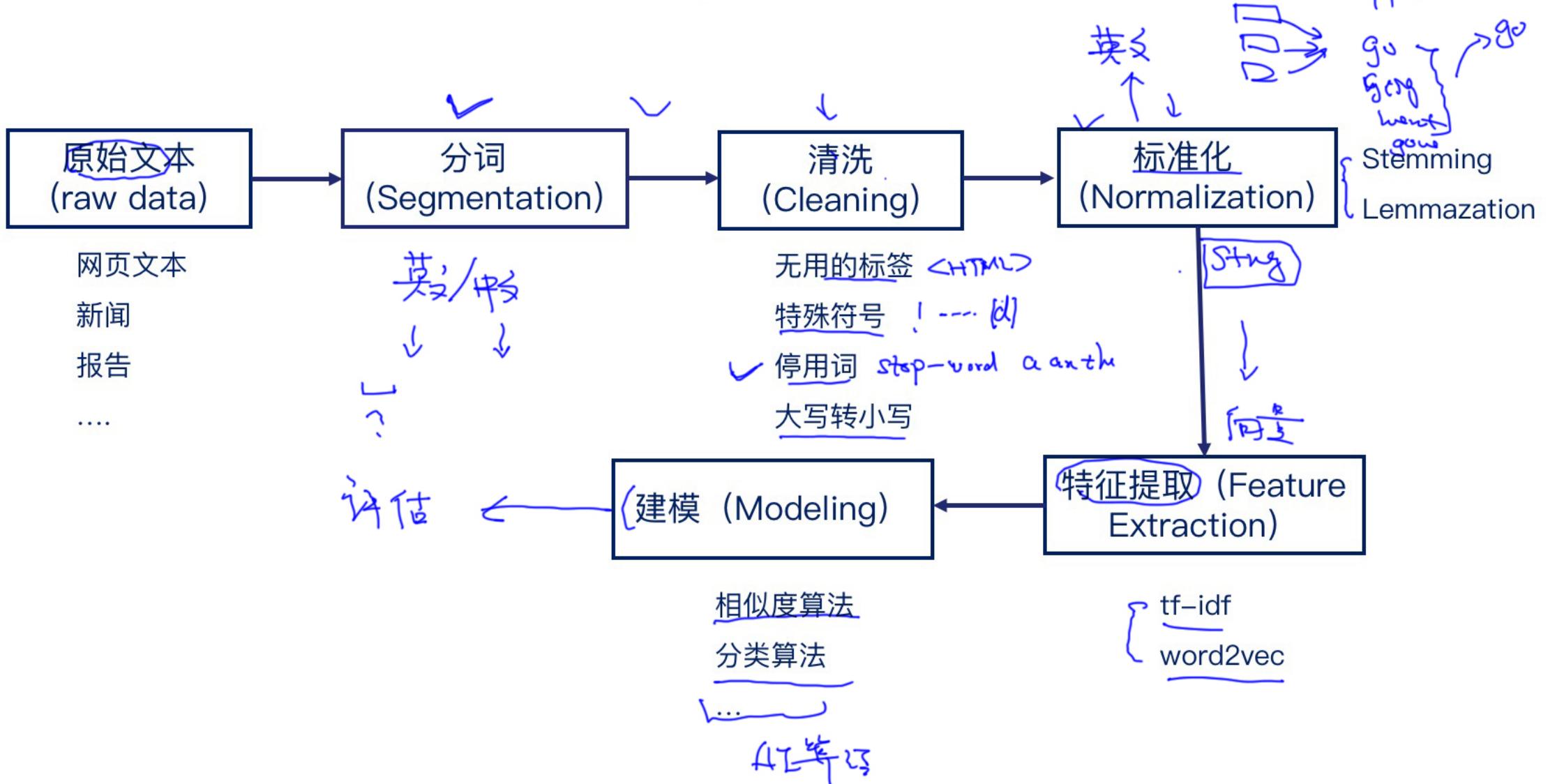


自然语言处理训练营 (3)

点此获取更多资源

NLP 项目

Pipeline



What we cover today

- Word Segmentation
 - Spell Correction ←
 - Stop Words Removal
 - Stemming
- } 规则
} 变换

Word Segmentation (分词)

Word Segmentation Tools

Jieba分词 <https://github.com/fxsjy/jieba>

SnowNLP <https://github.com/isnowfy/snownlp>

LTP <http://www.ltp-cloud.com/>

HanNLP <https://github.com/hankcs/HanLP/>

....
FudanNLP

Segmentation Method 1: Max Matching(最大匹配)

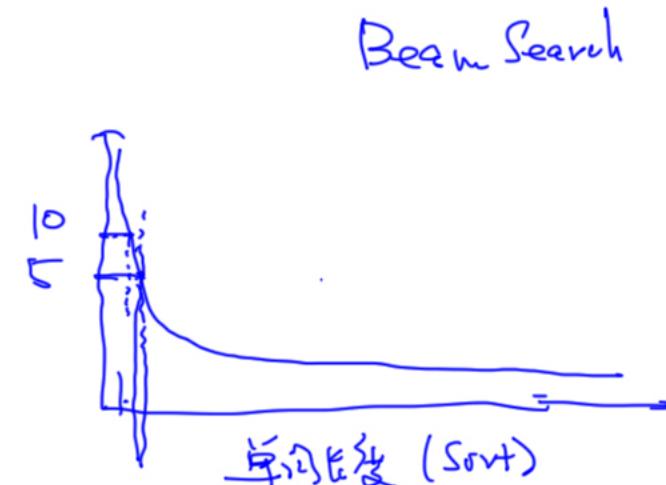
[前向最大匹配] (forward-max matching) , max_len = 5 (10)

例子: [我们 经常 有 意见 分歧]

词典: [“我们”, “经常”, “有”, “有意见”, “意见”, “分歧”]

①	[我们经常有意见分歧]	✗
	[我们经常]	✗
	[我们经]	✗
	[我们]	✓
②	[经常有意见]	✗
	[经常有意]	✗

③	[经常有]	✗
	[经常]	✓
④	[有意见分歧]	✗
	[有意见分]	✗
	[有意见]	✓
⑤	[分歧]	✓



贪心

Segmentation Method 1: Max Matching(最大匹配)

后向最大匹配 (backward-max matching)

例子: 我们经常[有意见]分歧 Max-len=5

90% 贪心
10% 不一样

词典: [“我们”, “经常”, “有”, “有意见”, “意见”, “分歧”]

① 有意见分歧 ×
 意分歧 ×
 见分歧 ×
 分歧 ✓

② 有经常有意见 ×
 常有意见 ✓
 有意 ✓

③ 我们 ✓

④ 我们经常 ×
 们经常 ×
 经常 ✓

Segmentation Method 1: Max Matching(最大匹配)

最大匹配的缺点?

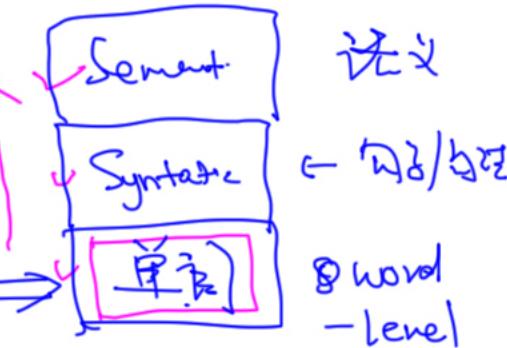
例子: 我们经常有意见分歧

词典: [“我们”, “经常”, “有”, “有意见”, “意见”, “分歧”]

我们|经常|有意见|分歧

我们|经常|(有)意见|分歧

Max Matching

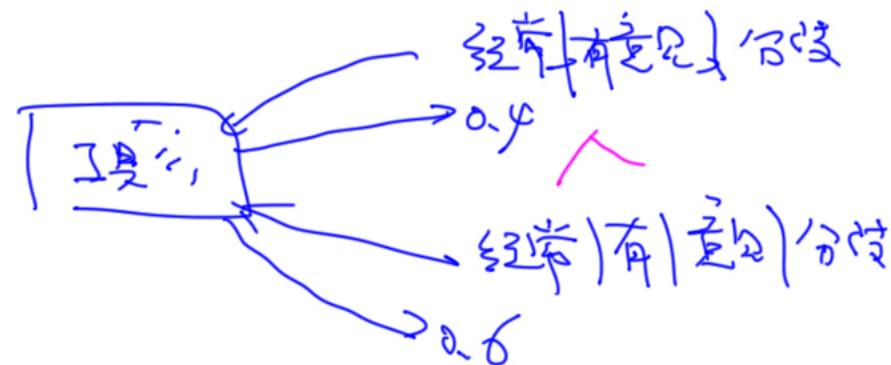
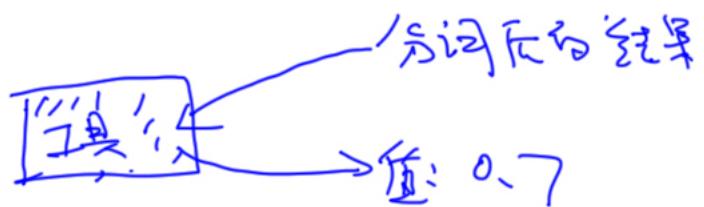


- 细分 (有可能是更好)
- 局部最优
- 简单 (Max-kew)
- 语义 (不能考虑语义)

Segmentation Method 2: Incorporate Semantic (考虑语义)

- 例子：经常有意见分歧

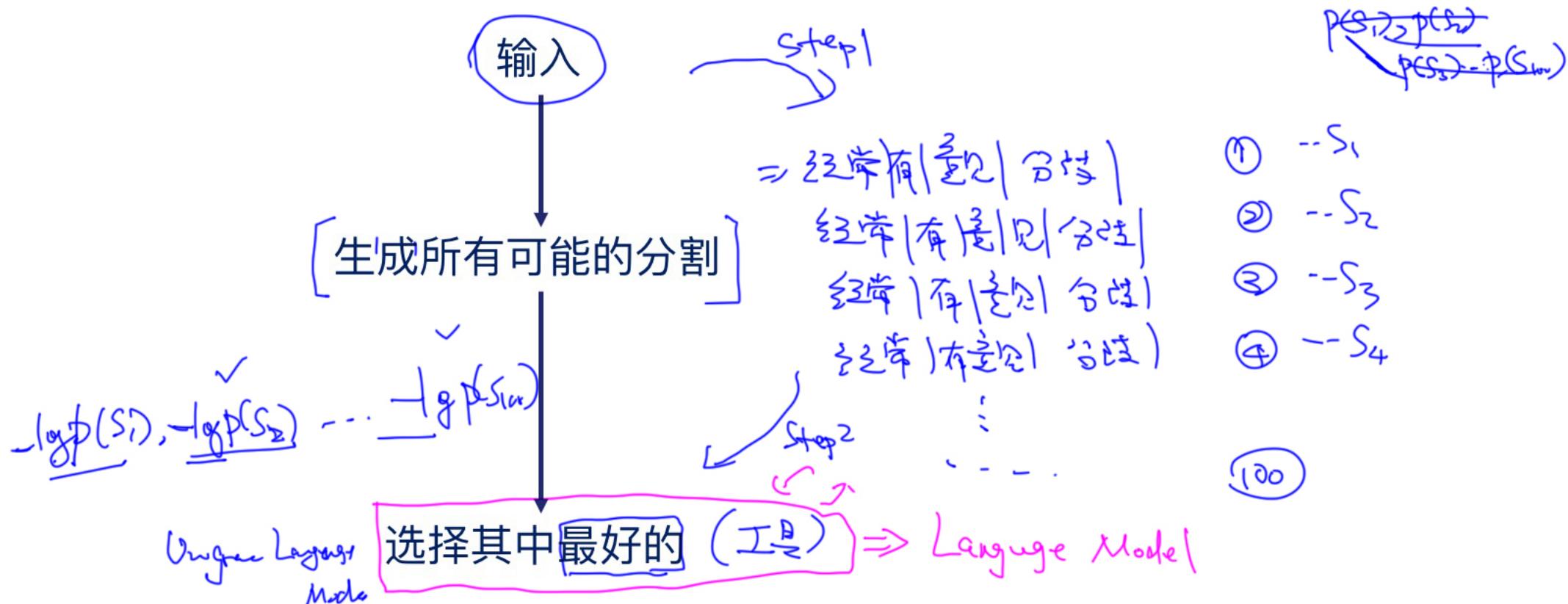
词典：[“有”，“有意见”，“意见”，“分歧”，“见”，“意”，]



Segmentation Method 2: Incorporate Semantic (考虑语义)

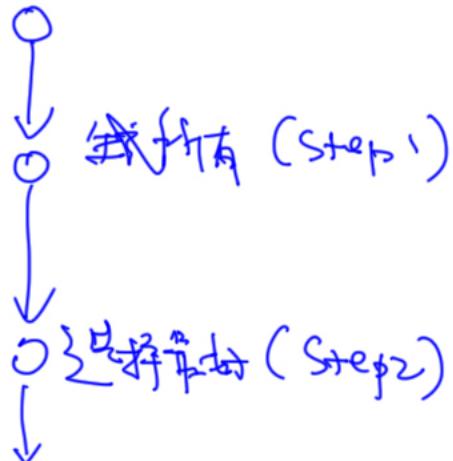
例子: [经常有意见分歧] ^{input}

词典: ["有", "有意见", "意见", "分歧", "见", "意"] "经常"



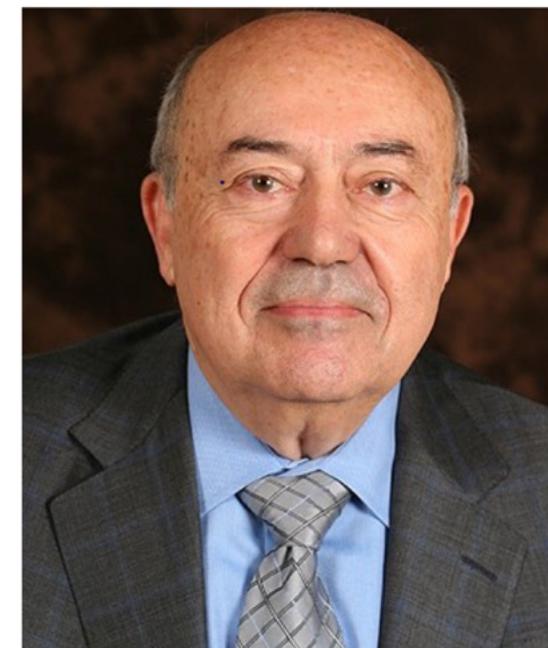
Segmentation Method 2: Incorporate Semantic (考虑语义)

怎么解决效率问题?



Ask him!

Step 1 > Step 2 分开
Step 1 + Step 2 ?
↓ Viterbi
(Dp)



Andrew

Viterbi

(use \rightarrow 高效)

Viterbi diagram

例子：“经常有意见分歧”

10↑

$$P(\text{分歧}) = 10^{-8}$$

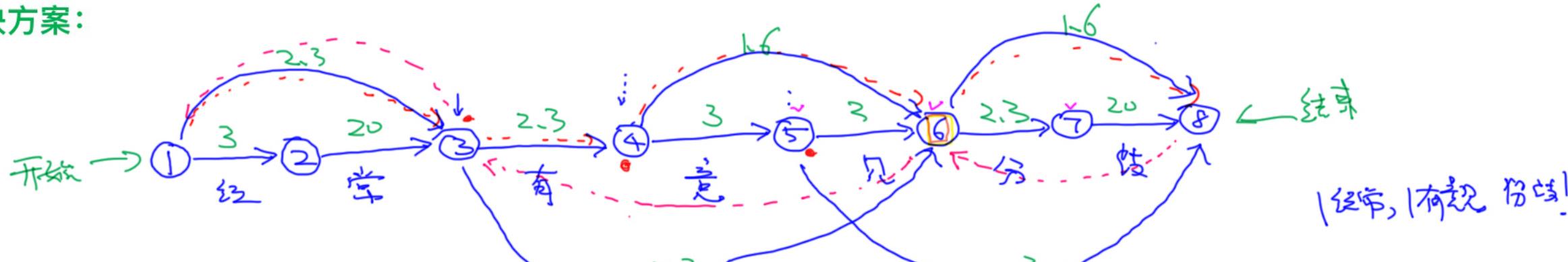
$$\frac{-\log(10^{-8})}{20}$$

词典：[“经常”，“经”，“有”，“有意见”，“意见”，“分歧”，“见”，“意”，“见分歧”，“分”] $P(\text{分歧}) = 0.1$

概率：[0.1, 0.05, 0.1, 0.1, 0.2, 0.2, 0.05, 0.05, 0.05, 0.1] ↪

Log(x): [2.3, 3, 2.3, 2.3, 1.6, 1.6, 3, 3, 3, 2.3]

解决方案：



| 经常 | 有 | 意 | 分 | 差 |

定义

$f(8) = \min\{f(5) + 3, f(7) + 2.0\}$

$f(7) = \min\{f(6) + 1.6, f(8) + 2.0\}$

$f(6) = \min\{f(4) + 3, f(5) + 2.3\}$

$$f(7) = f(6) + 2.3$$

$$f(6) = \min \begin{cases} f(4) + 1.6 = 6.2 \\ f(5) + 3 = 10.6 \end{cases} \checkmark$$

$$\min \begin{cases} f(4) + 1.6 = 6.2 \\ f(5) + 3 = 10.6 \end{cases}$$

$$P(\text{经常, 有, 意}) > P(\text{经常, 有, 差})$$

$$\log P(\text{经常, 有, 意}) > \log P(\text{经常, 有, 差})$$

$$(\log P(\text{经常}) + \log P(\text{有}) + \log P(\text{意})) > \log P(\text{经常}) + \log P(\text{有}) + \log P(\text{差})$$

$$f(8) = \min \begin{cases} f(5) + 3 \dots \text{从第5个} \\ f(7) + 2.0 \dots \text{从第7个} \end{cases}$$

0	3	2.3	4.6	7.6	4.6	6.9	6.2	13.68
$-f(1)$	$-f(2)$	$-f(3)$	$-f(4)$	$-f(5)$	$-f(6)$	$-f(7)$	$-f(8)$	

开始 1 1 3 4 3 6 6

Word Segmentation Summary

知识点总结：

- { ✓ - 基于匹配规则的方法 [Max-Match] }
- ✓ - 基于概率统计方法 (LM, HMM, CRF..)
 Unigram
 Language Model
- ✓ - 分词可以认为是已经解决的问题

需要掌握什么？

- 可以自行实现基于最大匹配和Unigram LM的方法

拼写纠正



Spell Correction (拼写错误纠正)

① 手写

Spell Correction (拼写错误纠正)

① 错别字

(用户输入 (input)) → AI → (用户输入 (correction))

② 不符
错别字，不全
I am go home

(天起)

→ (天气)

↓ going
拼写模型

✓ theris

→ (theirs)

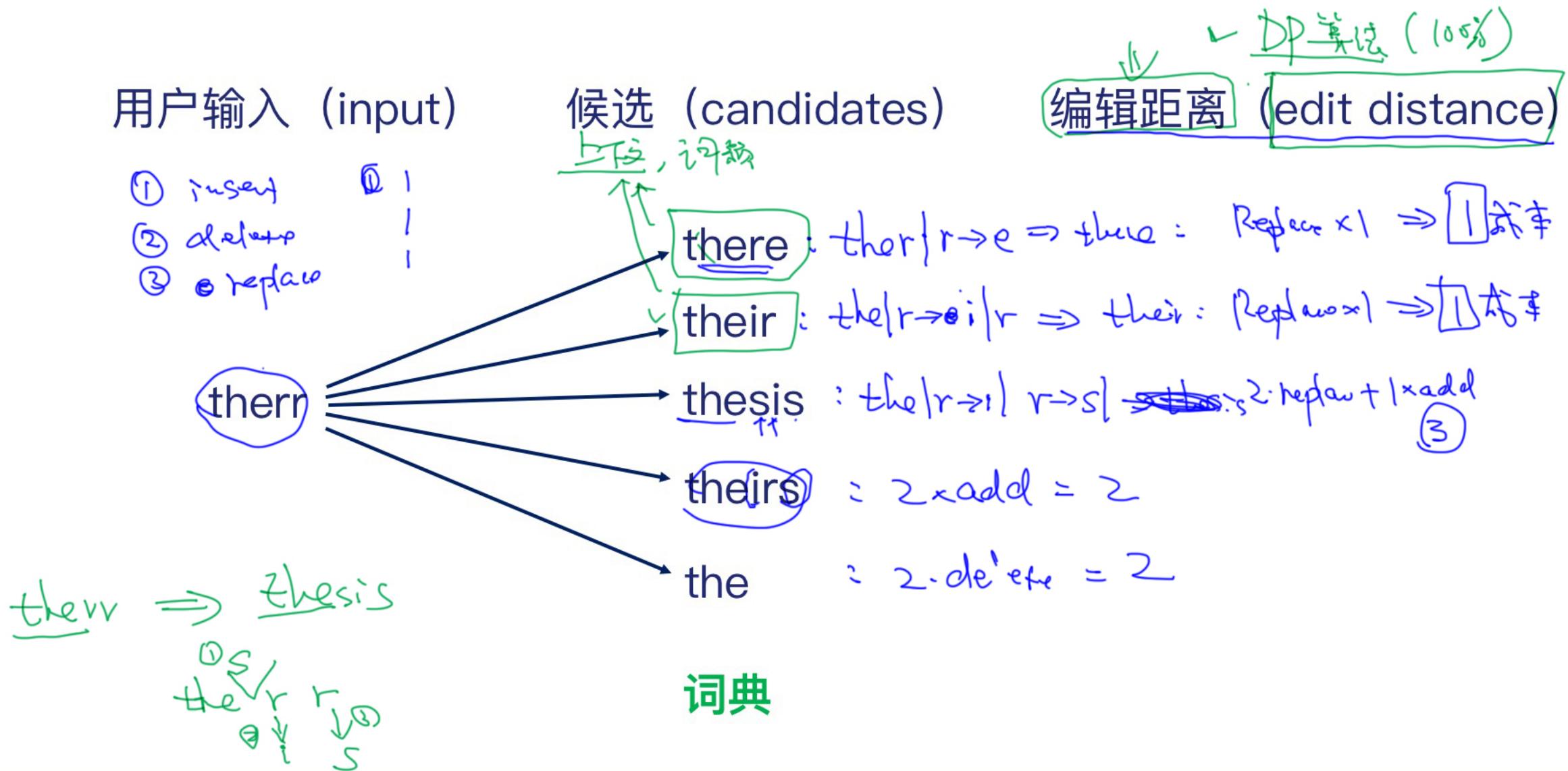
go

机器学系

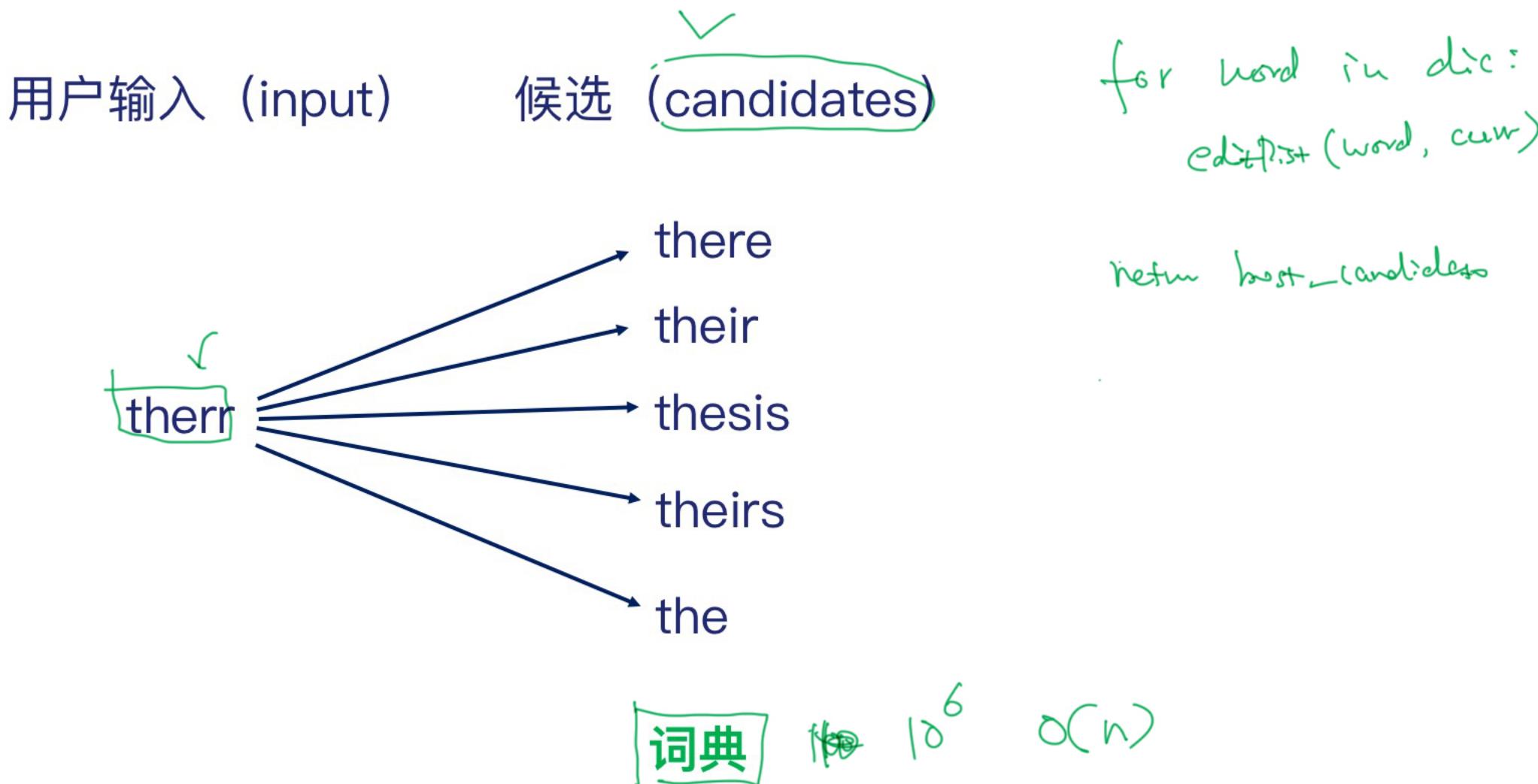
→ (机器学习)

怎么做?

Find the words with smallest edit distance



Find the words with smallest edit distance

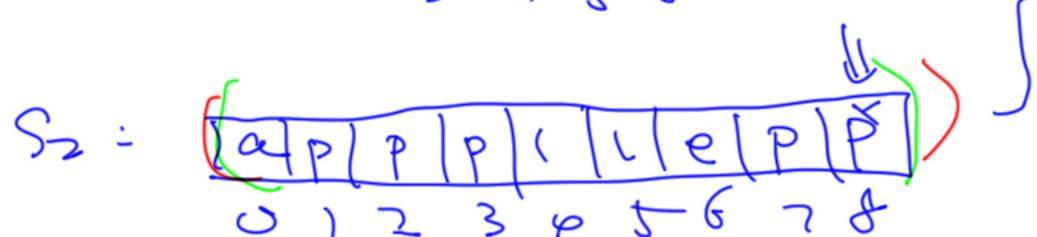


DP

DP 算法核心：

insert del.

Big Problem \Rightarrow Smaller Problem



$$d(S_1[0:6], S_2[0:8]) = d(S_1[0:5], S_2[0:7]),$$

$$F(n) = F(n-2) + F(n-1)$$

$$F(n+1) = F(n-1) + F(n)$$

$$\begin{bmatrix} F(n) \\ F(n-1) \end{bmatrix} = \left(\begin{array}{cc} & \\ & \end{array} \right) \begin{bmatrix} F(n-1) \\ F(n-2) \end{bmatrix}$$

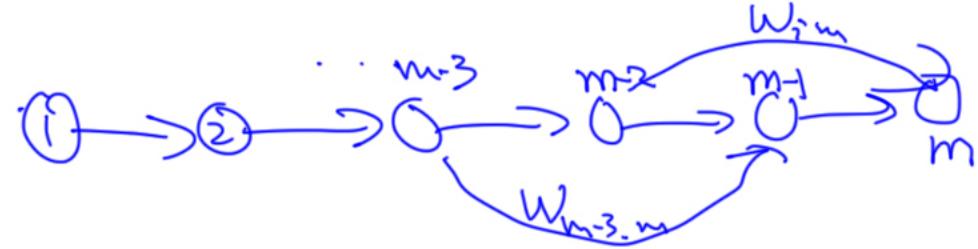
$$\begin{bmatrix} F(n) \\ F(n-1) \end{bmatrix} = \left(\begin{array}{cc} & \\ & \end{array} \right)^n \begin{bmatrix} F(1) \\ F(0) \end{bmatrix}$$

$$A = C \oplus D$$

$$A^n = (C \oplus D)(C \oplus D)$$

$$[]^n =$$

$$f(i) = f(i-2) + f(i-1)$$

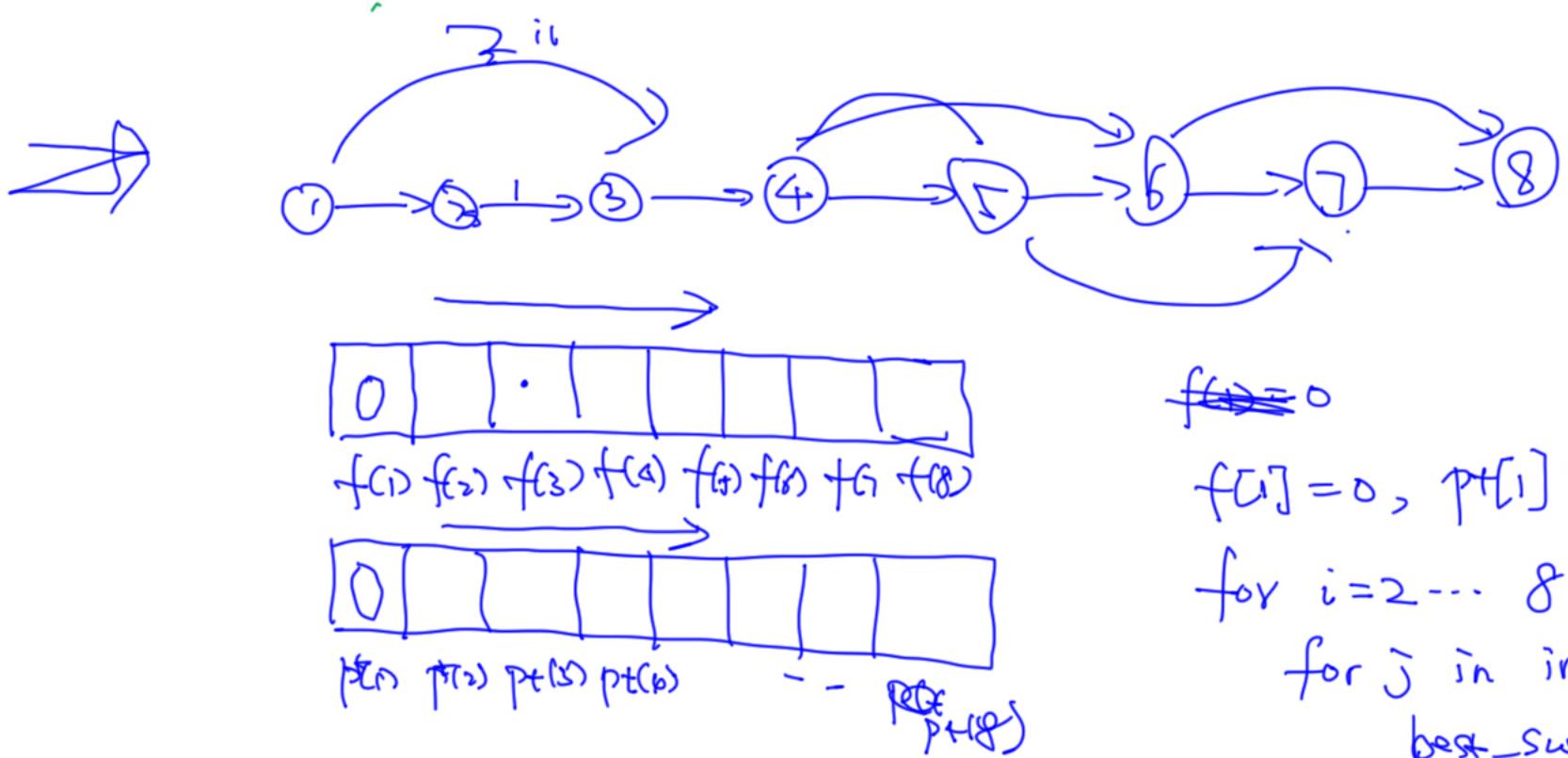


$f(m)$: 从节点 1 到节点 m 的最短路径 值

$$f(m) = \min_{\text{all paths}} \{ f(i) + w_{im} \}$$

$$f(m) = \min_{\substack{i \in \{1, 2, \dots, m-1\} \\ i \in \Omega(m)}} \{ f(i) + w_{im} \}$$

$\boxed{\Omega(m)}$ all the incoming links



$$\begin{aligned} f(3) &= f(2) + 1 \\ &= f(1) + 2 \end{aligned}$$

~~$f(0) = 0$~~

$f[0] = 0, p[0] = 0$

for $i = 2 \dots 8$

for j in incoming links

best_swe = + ∞

best_inmg_link_idx = i

swe = $f(j) + w(j \rightarrow i)$

if $swe < best_swe$:

best_swe = swe;

~~point~~
 $p[i] = j$

input: 经常有意见吗

$\{ \begin{array}{l} S_1 - \sim \\ S_2 - [] | \dots \\ S_3 - . \\ \vdots \\ S_{100} \end{array} \}$

$$\begin{aligned} P(\text{经常有意见吗}) &= P(\text{经常})P(\text{有})P(\frac{3}{5}\%)P(\text{意见}) \\ &= 0.0001 \cdot 0.000035 = 0.000002 \cdot 0.0001 \\ &= \text{Q} - \text{inf} / \text{underflow} \end{aligned}$$

计算: double/float

$p(S_1) > p(S_2) \dots p(S_{100})$

$$\log P(\text{经常有意见吗}) = \log p(\text{经常}) + \log p(\text{有}) + \log p(\frac{3}{5}\%) + \log p(\text{意见})$$

$$\log(x \cdot y \cdot z) = \log x + \log y + \log z$$

等同 $\begin{cases} p(S_1) > p(S_2) > p(S_3) > p(S_4) \\ \log p(S_1) > \log p(S_2) > \log p(S_3) > \log p(S_4) \end{cases}$

3

$$S_1 = \textcircled{1} \text{ 经常 | 有 } \frac{3}{2} \text{ 见 } \text{ 分歧}$$

$$S_2 = \textcircled{2} \text{ 经常 | 有 } \frac{3}{2} \text{ 见 } \text{ 分歧}$$

Unigram Model ←

Unigram Large Model

$$P(\text{经常}) = \frac{100}{\# \text{ of words} \text{ in book}}$$



经常: 100

有: 1000

见: 500

分歧: 200

⋮

工具 ⇒ 语言模型 (Language Model)

$$P(S_1) = 0.3$$

$$\textcircled{1} P(\text{经常}, \text{有}, \frac{3}{2}, \text{见}, \text{分歧}) = P(\text{经常}) \cdot P(\text{有}) \cdot P(\frac{3}{2}) \cdot P(\text{见}) \cdot P(\text{分歧}) = 0.3$$

$$P(S_2) = 0.35$$

$$P(\text{经常}, \text{有}, \frac{3}{2}, \text{见}, \text{分歧}) = P(\text{经常}) \cdot P(\text{有}) \cdot P(\frac{3}{2}) \cdot P(\text{见}) \cdot P(\text{分歧}) = 0.35$$

$$P(\text{我们今天上课}) = P(\text{我们}, \text{今天}, \text{上课})$$

$$= P(\text{我们}) \cdot P(\text{今天}) \cdot P(\text{上课})$$