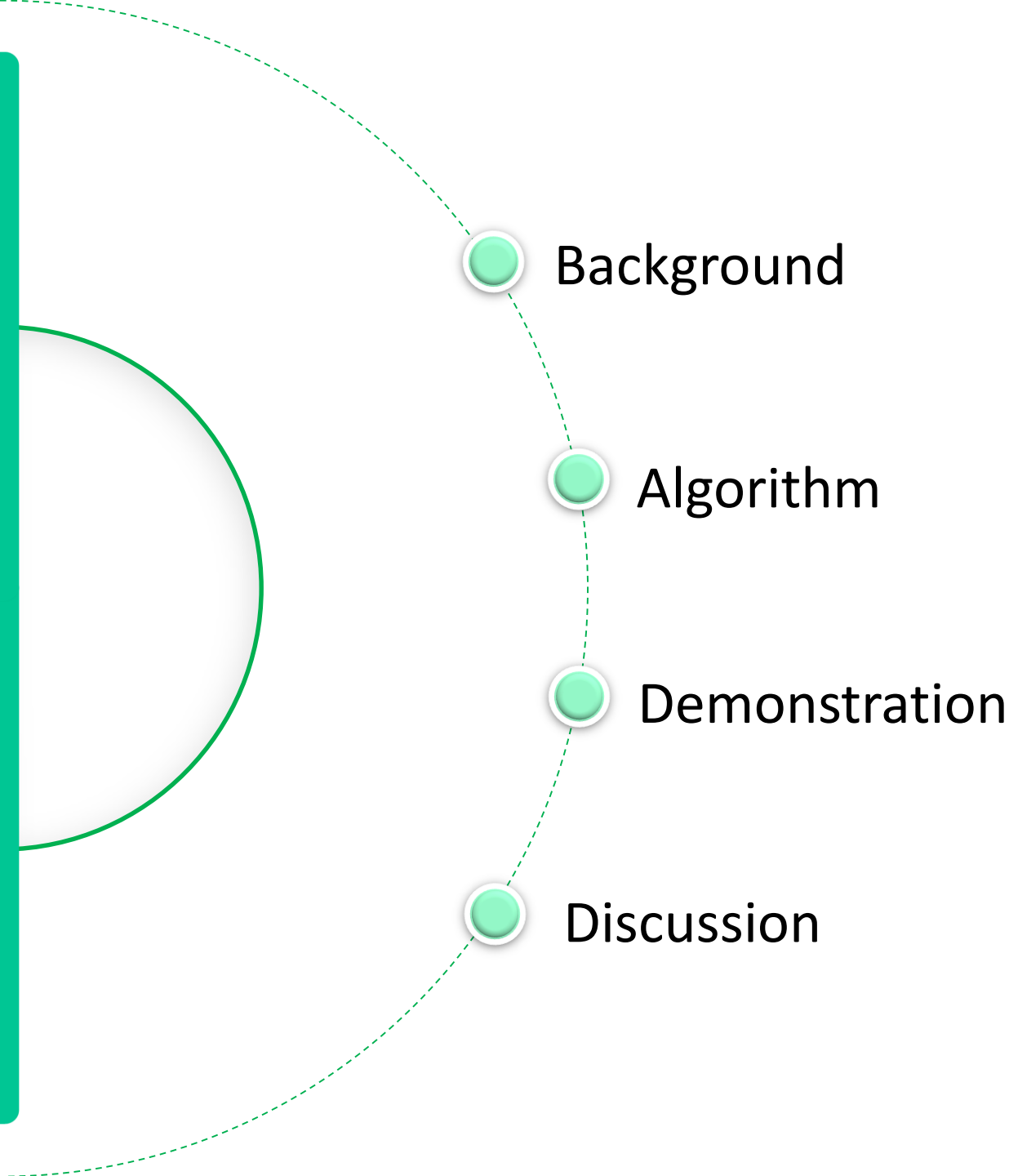
Chinese word segmentation

Presenter: Mingxin Chen

Group: Mingxin Chen, Mingwei Wei, Bin Jiang, Ange Tong


Background
Algorithm
Demonstration
Discussion

What is Chinese word segmentation?

like … this …

• 语言信息处理	• 语言 / 信息 / 处理
• 鉴萍老师美丽大方	• 鉴萍 / 老师 / 美丽 / 大方
• 诚实是一种美德	• 诚实 / 是 / 一种 / 美德

Why segment?

Great Significance!

valuations

The basic of information retrieval, information extraction, information classification and so on.

applications

- Polyphone recognition (多音字识别)
- Text proofreading (文本校对) eg. [于预 >> 干预]

and more …

……

Challenge

What is challenge of Chinese word segmentation?

To solve the ambiguity, of course!!!

So .. What do you mean … ambiguity?

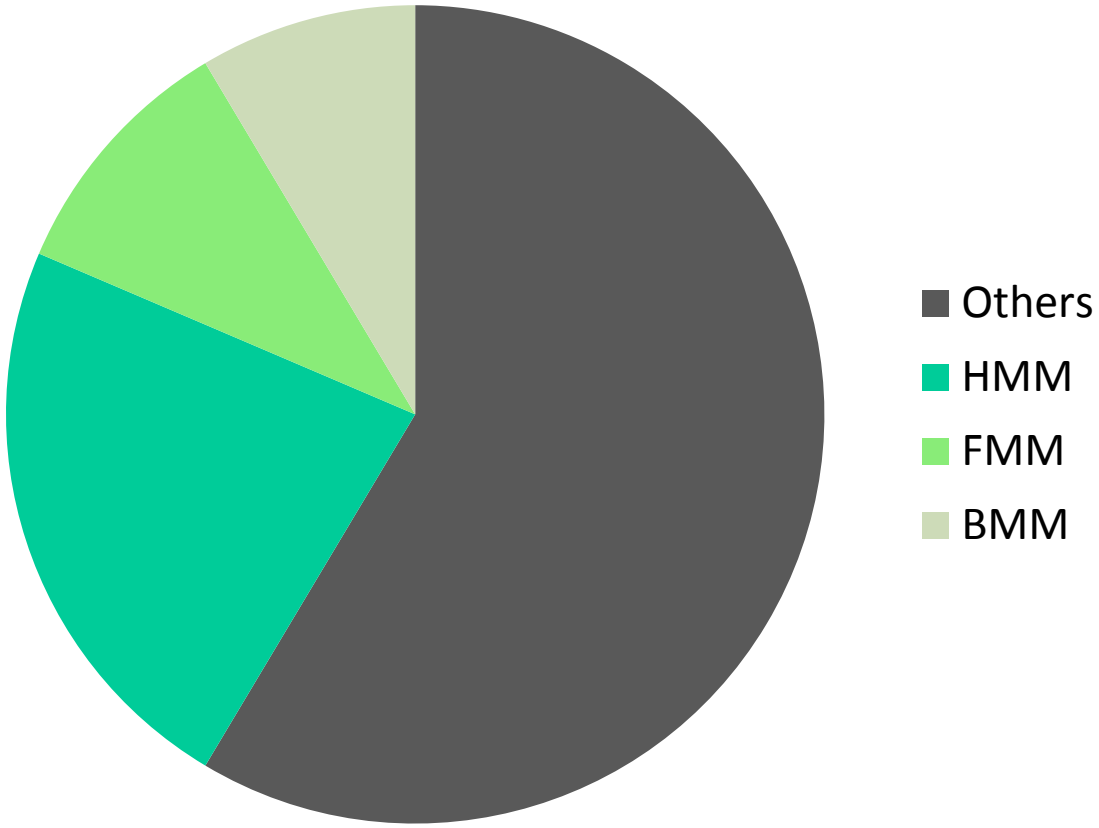
Like this ..

这个学生会打篮球

- 这个 / 学生 / 会 / 打 / 篮球
- 这个 / 学生会 / 打 / 篮球

OK, I got it.

Algorithm classification



Algorithms in our solution

Hidden Markov Model Algorithm

We define C = $\{B, M, E, S\}$, $O = O1O2..Oi$

• B	• Begin of word
• M	• Middle of word
• E	• End of word
• S	• Single word

O1O2O3O4O5O6O7O8O9O10
你/现在/应该/去/幼儿园/了
B BE BE S BME S

Give us O, we should calculate C

That means we calculate $argmax\ C\ P\big((C1, C2..Ci|O1, O2..Oi)\big) = P(C|O)$ (1)

Simplify (1), we get $argmax\ C\ P(O|C)P(C)$ (2)

Analysis (2), we get final formula:

argmax C $P(O1|C1)P(O2|C2)..P(Oi|Ci) * P(C1)P(C2|C1)P(C3|C2)..P(C_i|C_{i-1})$

A Pure-HMM Segmentation

We calculate three matrix

1 initial probability matrix

$$Pi = Count(C_i) / \sum Count(C_j)$$

2 state-transition matrix

$$Aij = P(C_j|C_i) = \frac{P(C_i, C_j)}{P(C_i)} = Count(C_i, C_j)/Count(C_i)$$

3 emitter matrix

$$Bij = P(O_j|C_i) = \frac{P(O_j, C_i)}{P(C_i)} = Count(O_j, C_i)/Count(C_i)$$

Add 1 smooth

$$Bij = P(O_j|C_i) = (Count(O_j, C_i) + 1)/(Count(C_i) + N)$$

Viterbi Algorithm

Formula derivation

$$argmax\ C\ P\big((C1, C2..Ci|O1, O2..Oi)\big)$$

$$= \text{argmax C } P(O1|C1)P(O2|C2)..P(Oi|Ci) * P(C1)P(C2|C1)P(C3|C2)..P(C_i|C_{i-1})$$

$$= \text{argmax C } P(O1|C1)P(O2|C2)..P(O_{i-1}|C_{i-1}) * P(C1)P(C2|C1)P(C3|C2)..P(C_i|C_{i-1}) * P(Oi|Ci)$$

$$= \text{argmax C } P((C1, C2, \dots, C_{i-1}|O1, O2, \dots O_{i-1})) * P(C_i|C_{i-1}) * P(Oi|Ci)$$

$$argmax\ C\ P\big((C1, C2..Ci|O1, O2..Oi)\big)$$

$$= \text{argmax C } P((C1, C2, \dots, C_{i-1}|O1, O2, \dots O_{i-1})) * P(C_i|C_{i-1}) * P(Oi|Ci)$$

Viterbi Algorithm

Process

Calculate the initial probability: $P(C1|O1) = P(C1) * P(O1|C1)$

Calculate$P\big((C1, C2..Ct|O1, O2..Ot)\big) =$
$argmax\ C\ \ P((C1, C2, \dots, C_{t-1})\,|\,O1, O2, \dots O_{t-1}))) * P(C_t|C_{t-1})\ *\ P(Ot|Ct)$,
using an array path to record the value of Ct when taking the maximum

Get the maximum sequence: $argmax\ C\ P((C1,C2..\ Ci\,|\,O1,O2..Oi))$,
we can get the value of Ci when P taking the maximum.

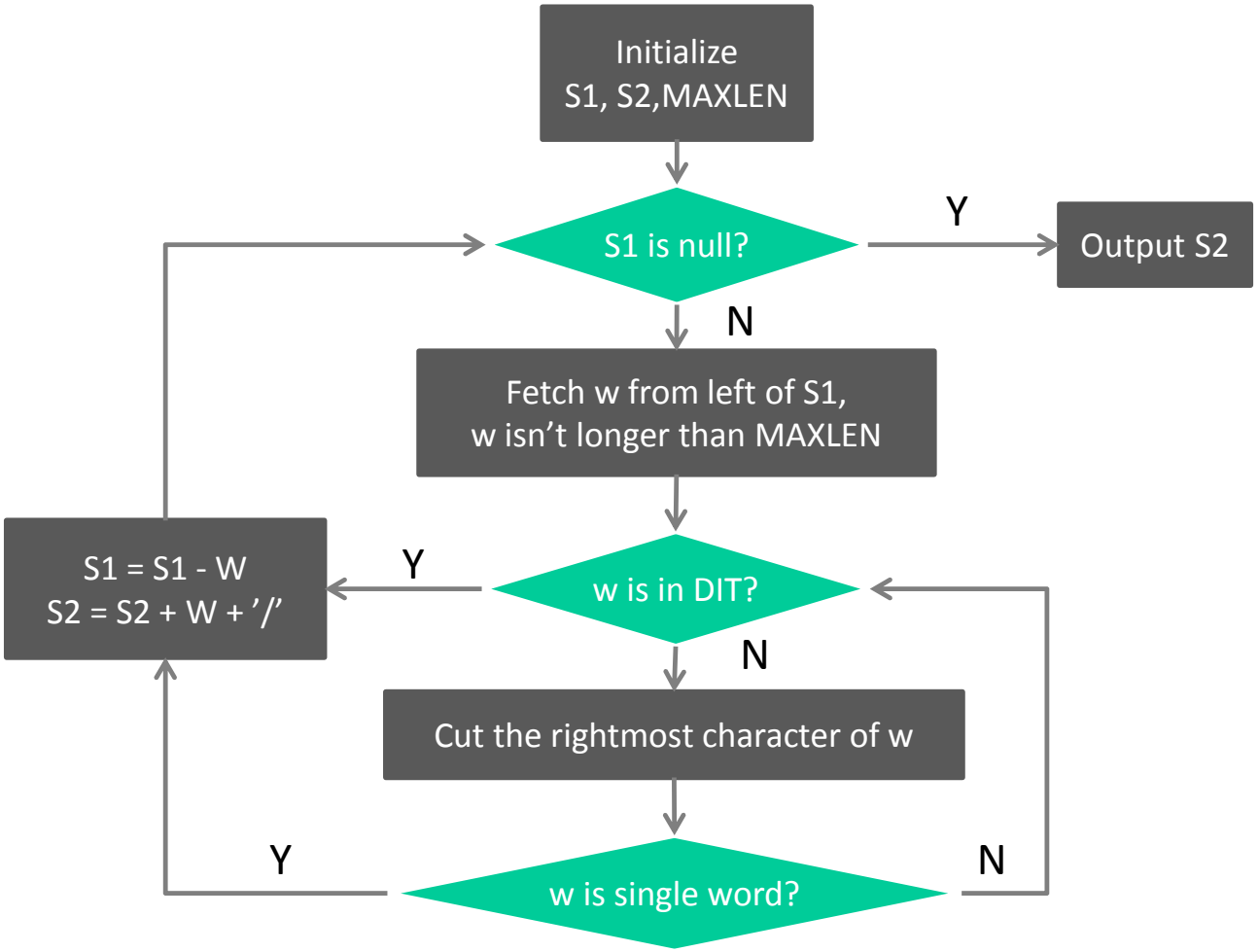
backtracking:
Using the path array to backtrack and get the hidden sequence.
If the value is E or S then can be divided into a word.

A Pure-HMM Segmentation

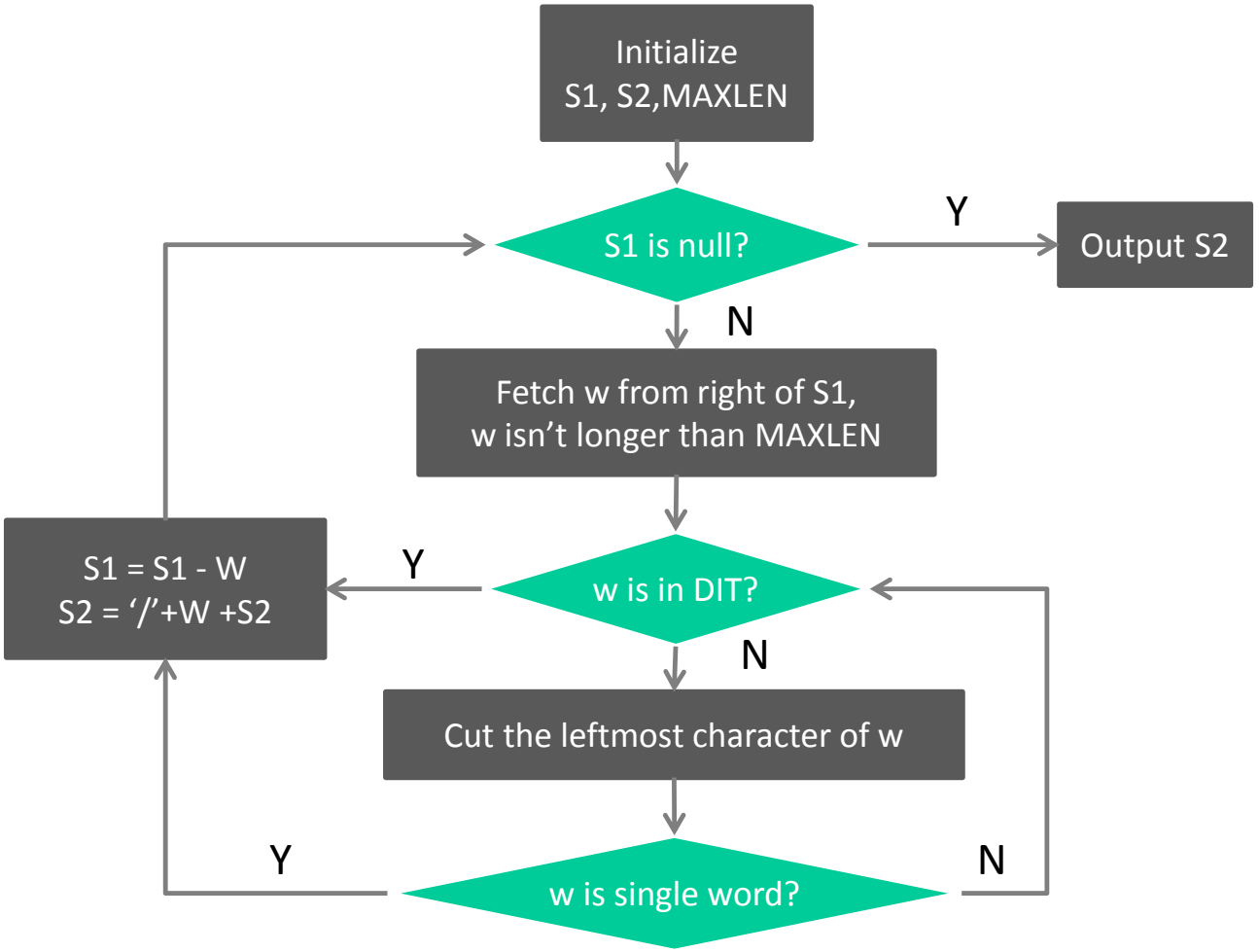
Example

1. Process train set and get the P, A, B matrix
   - Pi
   - Aij
   - Bij
2. Process test set
   Remove punctuation, spaces and ..
   - 语言信息处理
   - 中文分词
3. Change sentence to codes
   That means O
   - 122 123 133 143 185 104
   - 384 654 343 212
4. Observe O, calculate hidden sequence by using viterbi (0:B 1:M 2:E 3:S)
   - 020202
   - 0202
5. Give the last result
   - 语言 / 信息 / 处理
   - 中文 / 分词

Forward Maximum Matching Algorithm



Backward Maximum Matching Algorithm

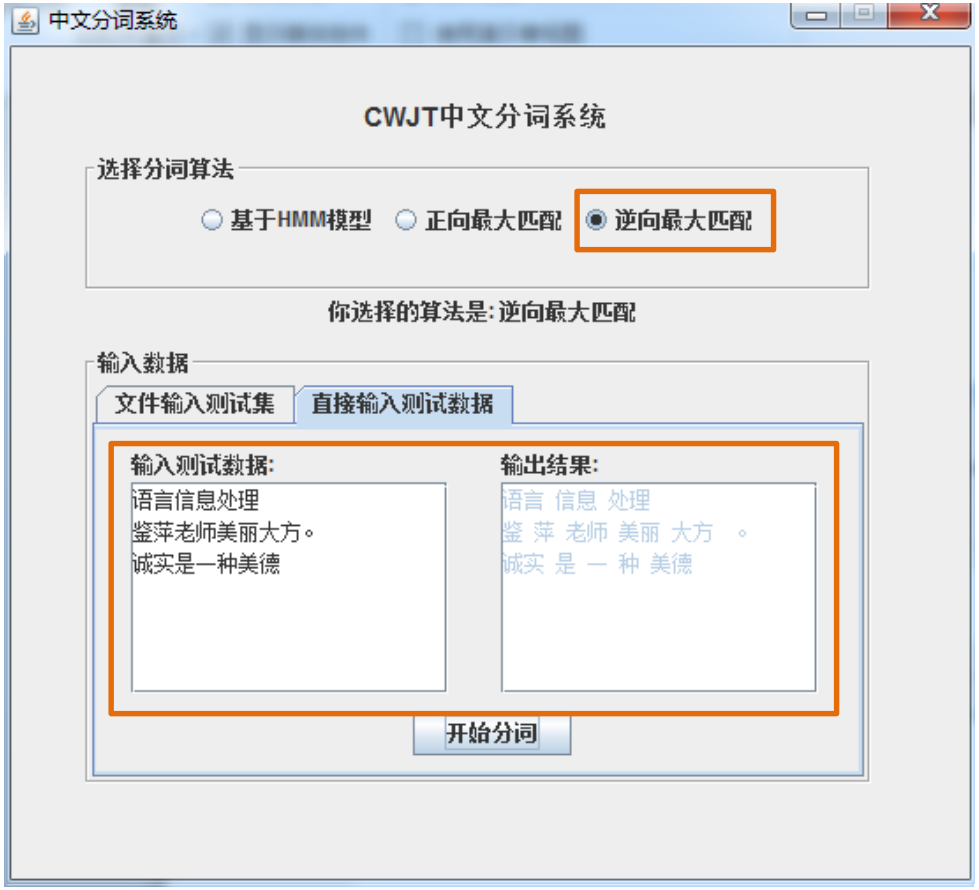


Demo

Demo

扬 帆 远东 做 与 中国 合作 的 先行
希腊 的 经济 结构 较 特殊 。
海 运业 雄踞 全球 之首 ， 按吨 位计 占 世界 总数 的 １７％ 。
另外 旅游 、 侨汇 也 是 经济 收入 的 重要 组成 部分 ， 制 造业 规模 相对 较 小 。
多年来 ， 中 希贸易始 终处 于 较低 的 水平 ， 希腊 几乎 没有 在 中国 投资 。
十几年来 ， 改革 开放 的 中国 经济 高速 发展 ， 远东 在 崛起 。
瓦西里斯 的 船 只 中有 ４０％ 驶 向 远东 ， 每个月几乎 都 有 两三 条船 停靠 中国 港口 。
他 感受 到 了 中国 经济 发展 的 大潮 。
他 要 与 中国 人 合作 。
他来 到 中国 ， 成为 第一个 访华 的 大 船主 。

Demo



Score of HMM


```
151227  INSERTIONS: 2
151228  DELETIONS:  3
151229  SUBSTITUTIONS:  5
151230  NCHANGE:    10
151231  NTRUTH: 45
151232  NTEST:  44
151233  TRUE WORDS RECALL:  0.822
151234  TEST WORDS PRECISION:   0.841
151235  === SUMMARY:
151236  === TOTAL INSERTIONS:   8496
151237  === TOTAL DELETIONS:    4193
151238  === TOTAL SUBSTITUTIONS:    16237
151239  === TOTAL NCHANGE:  28926
151240  === TOTAL TRUE WORD COUNT:  106873
151241  === TOTAL TEST WORD COUNT:  111176
151242  === TOTAL TRUE WORDS RECALL:    0.809
151243  === TOTAL TEST WORDS PRECISION: 0.778
151244  === F MEASURE:  0.793
151245  === OOV Rate:   0.026
151246  === OOV Recall Rate:    0.431
151247  === IV Recall Rate: 0.819
```

The result is not very satisfactory

Score of FMM

```
148865  INSERTIONS: 2
148866  DELETIONS:  5
148867  SUBSTITUTIONS:  7
148868  NCHANGE:    14
148869  NTRUTH: 45
148870  NTEST:  42
148871  TRUE WORDS RECALL:  0.733
148872  TEST WORDS PRECISION:   0.786
148873  === SUMMARY:
148874  === TOTAL INSERTIONS:   6134
148875  === TOTAL DELETIONS:    4075
148876  === TOTAL SUBSTITUTIONS:    11818
148877  === TOTAL NCHANGE:  22027
148878  === TOTAL TRUE WORD COUNT:  106873
148879  === TOTAL TEST WORD COUNT:  108932
148880  === TOTAL TRUE WORDS RECALL:    0.851
148881  === TOTAL TEST WORDS PRECISION: 0.835
148882  === F MEASURE:  0.843
148883  === OOV Rate:   0.026
148884  === OOV Recall Rate:    0.305
148885  === IV Recall Rate: 0.866
```

not bad

Score of BMM


```
152233  INSERTIONS: 4
152234  DELETIONS:  3
152235  SUBSTITUTIONS:  7
152236  NCHANGE:    14
152237  NTRUTH: 45
152238  NTEST:  46
152239  TRUE WORDS RECALL:  0.778
152240  TEST WORDS PRECISION:   0.761
152241  === SUMMARY:
152242  === TOTAL INSERTIONS:   9502
152243  === TOTAL DELETIONS:    1908
152244  === TOTAL SUBSTITUTIONS:    10211
152245  === TOTAL NCHANGE:  21621
152246  === TOTAL TRUE WORD COUNT:  106873
152247  === TOTAL TEST WORD COUNT:  114467
152248  === TOTAL TRUE WORDS RECALL:    0.887
152249  === TOTAL TEST WORDS PRECISION: 0.828
152250  === F MEASURE:  0.856
152251  === OOV Rate:   0.026
152252  === OOV Recall Rate:    0.197
152253  === IV Recall Rate: 0.905
```

Recall rate is high.

Thanks !