Project: Chinese Word Segmentation System

Team Name: The Beacon

Team Leader: Wu Zhengke

Team Member(s): Wu Zhengke

Date: 2017/12/31

Index

Brief Introduction

This project is the assignment of Introduction to Computer Science in Dec. 2017.

In this project, a Chinese Word Segmentation System mainly base on HMM, Viterbi Algorithm and Maximum Matching Algorithm is built.

In addition, a local website is developed with Flask and Twitter Bootstrap as the User Interface.

1. Prototype System Introduction

1.1 Functions

This system mainly has the following five functions:

a. Training with given data

b. Cutting Chinese text into sentences

c. Segmenting Chinese sentences or text into words

d. User settings to correct the segmentation

e. A local website as User Interface

1.2 Running Environment

Windows 10 or others (not tested yet)

Python 3.6.0 (with packages installed in requirements.txt)

1.3 Developing Environment

Windows 10

Python 3.6.0 (with packages installed in requirements.txt)

Sublime Text 3

2. Task Allocation

System Overall Design: Wu Zhengke

Algorithm: Wu Zhengke

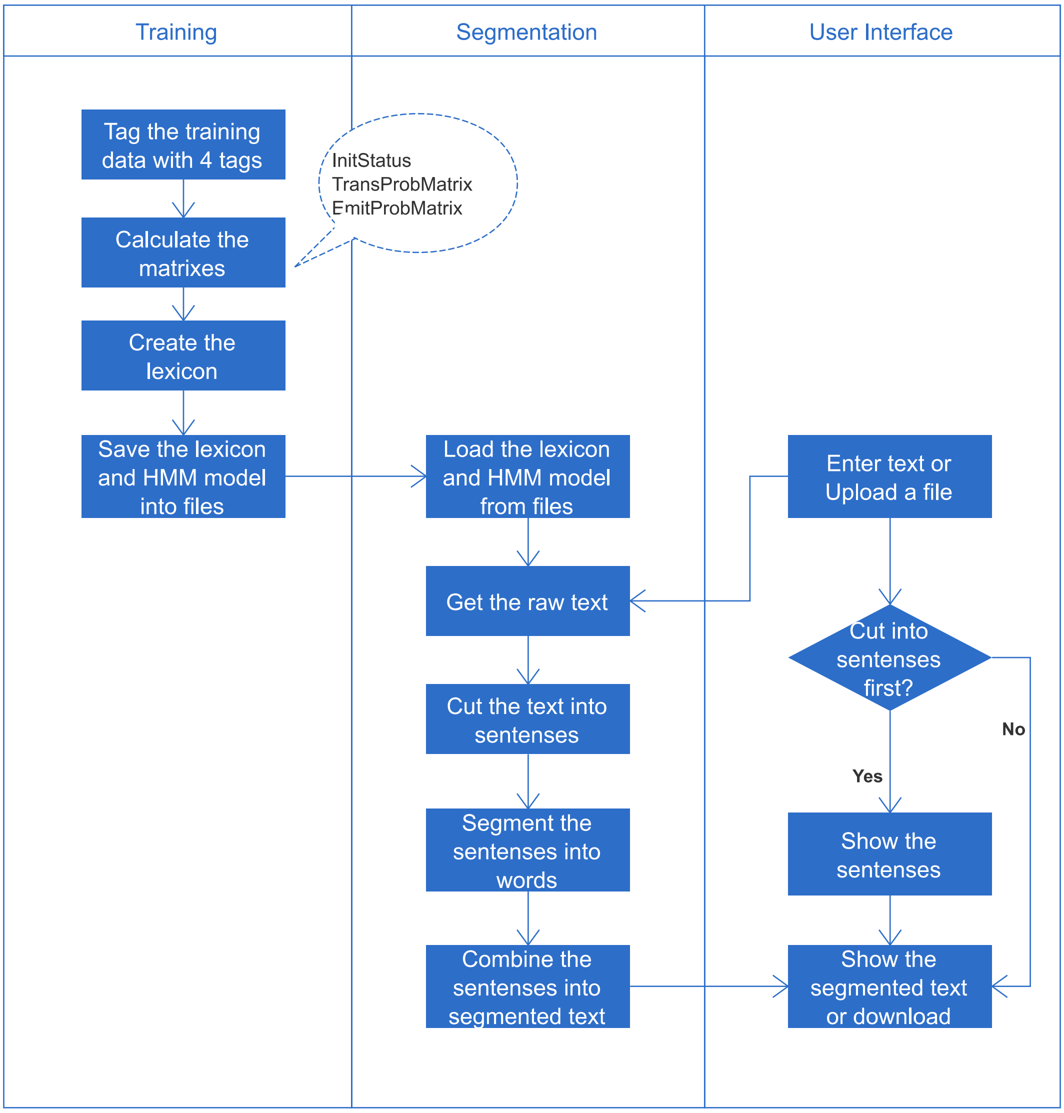
User Interface: Wu Zhengke

Debug and Test: Wu Zhengke

Report: Wu Zhengke

3. System Architecture

3.1 Overall Flowchart


Training
Segmentation
User Interface
Tag the training data with 4 tags
Calculate the matrixes
Create the lexicon
Save the lexicon and HMM model into files
InitStatus
TransProbMatrix
EmitProbMatrix
Load the lexicon and HMM model from files
Get the raw text
Cut the text into sentenses
Segment the sentenses into words
Combine the sentenses into segmented text
Enter text or Upload a file
Cut into sentenses first?
No
Yes
Show the sentenses
Show the segmented text or download

As illustrated in the above flowchart, the system is mainly composed of three components: Training, Segmentation and User Interface.

3.2 Training

The program opens the given training files and tags every character in them with 4 tags: B, E, S, M. Then according to the tags, it counts the occurrence of tags and calculates 3 matrixes: InitStatus, TransProbMatrix and EmitProbMatrix. (This part will be further explained in Section 4.)

After all these works done, the HMM model, i.e. the 3 matrixes, are saved into files.

In addition, the program creates a lexicon with all the words in the given training data and save it into files.

3.3 Segmentation

First, the program loads the trained HMM model and the lexicon into memory from files. When dealing with the given raw text, it cuts the text into sentences using the punctuations as separator and then segments every sentence before combining them together.

(This part will also be further explained in Section 4.)

3.4 User Interface

There are two ways for users to input text: entering text directly or uploading a file.

Then the program transfers the raw text to the part of Segmentation and show the segmented text. And if the user wants to cut the text into sentences first, it will also show the sentences.

For convenience, users can download the result.

4. Algorithm Description

This program is mainly based on the HMM (Hidden Markov Model). Before using this model, the program tags every character in the training corpus with 4 tags: B, M, E, S, which stand for Begin, Middle, End, Single respectively. For instance, the sentence can be tagged in this way:

今天　是　礼拜天　。
B E　　S　B M E　S

Then the program counts all the occurrence of these tags in the corpus and calculates the following 3 matrixes:

a. InitStatus: The possibility of each status of the first character in a sentence.

For example:

Initstatus[‘B’]=0.6
Initstatus[‘S’]=0.4
InitStatus[‘M’]=0
InitStatus[‘E’]=0

b. TransProbMatrix: The possibility of transformation from Status A to Status B

For example:

TransProbMatrix[‘B’][‘M’]=0.4
TransProbMatrix[‘B’][‘E’]=0.6
TransProbMatrix[‘B’][‘S’]=0
TransProbMatrix[‘B’][‘B’]=0

c. EmitProbMatirx: The possibility of a certain character in a certain status

For example:

EmitProbMatrix[‘S’][‘我’]=0.0001

EmitProbMatrix[‘S’][‘们’]=0

When segmenting, the program first cuts the text into sentences according to punctuations. And to preserve the punctuations when splitting, it’s better to use the sub() in the re module(regular expression) rather than split().

Then the program uses these three matrixes to determine the most possible status of each character in a sentence. To specify, that’s Viterbi Algorithm.

From the beginning character of a sentence, the program calculates the most possible trace of status. For the first character, its status traces are “B”, “M”, “E”, “S” and corresponding possibilities are InitStatus[status] * EmitProbMatrix[status][character].

And for every following character, its possible status are ‘B’, ‘M’, ‘E’, ‘S’, and correspondingly, the trace of a certain status is the trace of the previous character and its status that maximizes:

Possibility[previous_character][previous_status] *
TransProbMatrix[previous_status][status] *
EmitProbMatrix[status][character],

appending the certain status.

Manipulate in this way until the end of a sentence, then the program gets the possible traces of the last character and the maximum of them is the most possible status trace of the whole sentence. And the last step is to segment the sentence according to the status trace and combine all the sentences together into the text.

But after some experiments, the pure HMM model doesn’t work well enough (You may see the result in Section 5.2) as it is weak in dealing with IV (In Vocabulary) words. To offset this deficiency, this program combines the HMM with Maximum Matching Algorithm to handle IV words.

As for Maximum Matching Algorithm, it’s relatively easy. For a given sentence, it searches from the beginning of the sentence to match the longest word in the lexicon.

If it fails to match any word that begins with a character, then it adds this character to a string, which will be handled with HMM later.

In this way, this program is better at dealing with both IV and OOV words.

5. Demo and Testing Result

5.1 Demo

The demo can be got by forking from the Github repository[1] or unpacking the ZIP file attached.

But before running it, you need to install all the packages that are listed in requirements.txt. (A simple way to do this is to run *pip install -r requirements.txt*)

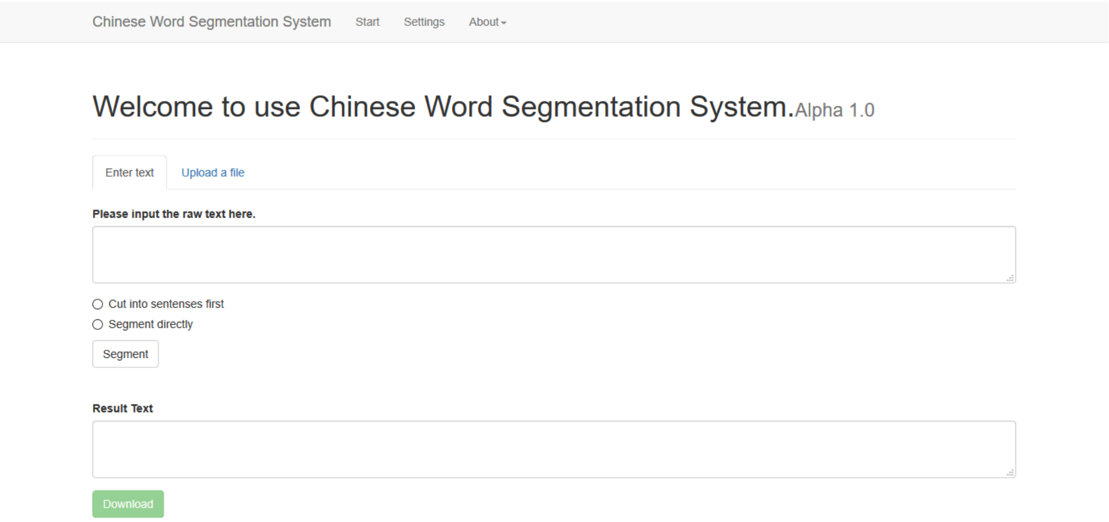
To run the demo, open the folder FlaskUI/ and run the command:

```
python FlaskUI.py runserver
```

Then you can visit *127.0.0.1:5000(or localhost:5000)* to use this demo.

In the index page, you can enter text, choose whether to cut it into sentences first and then click Segment to see the result.(See Figure 1, Figure 2,Figure 3.)

[1] https://github.com/keithnull/ChineseWordSegmentationSystem



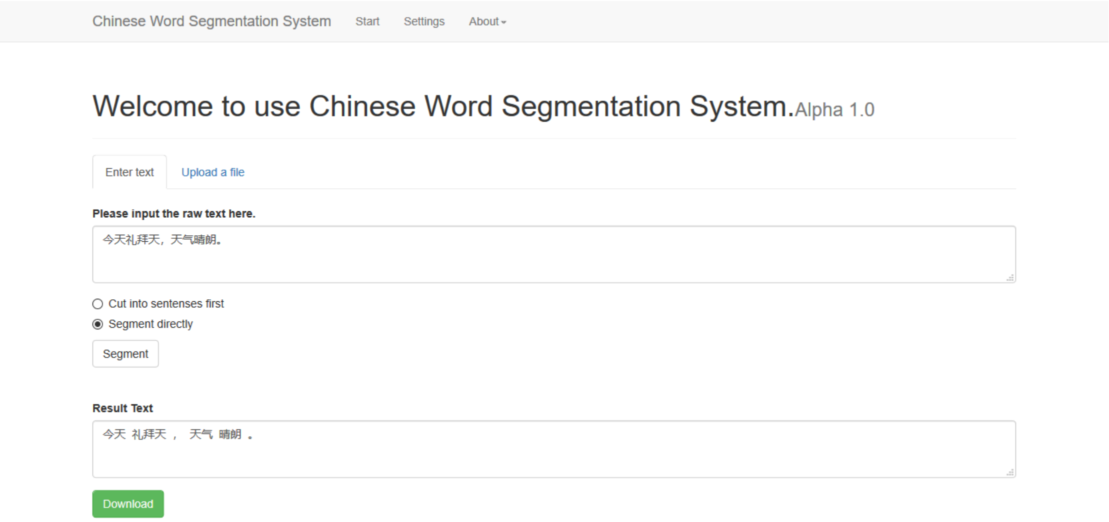
Welcome to use Chinese Word Segmentation System.Alpha 1.0

Please input the raw text here.

○ Cut into sentenses first
○ Segment directly

Result Text

Figure 1



Welcome to use Chinese Word Segmentation System.Alpha 1.0

Please input the raw text here.

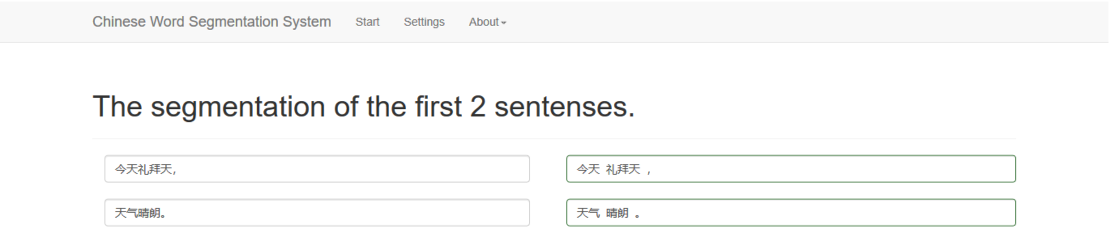
今天礼拜天，天气晴朗。

○ Cut into sentenses first
● Segment directly

Result Text

今天 礼拜天 ， 天气 晴朗 。

Figure 2

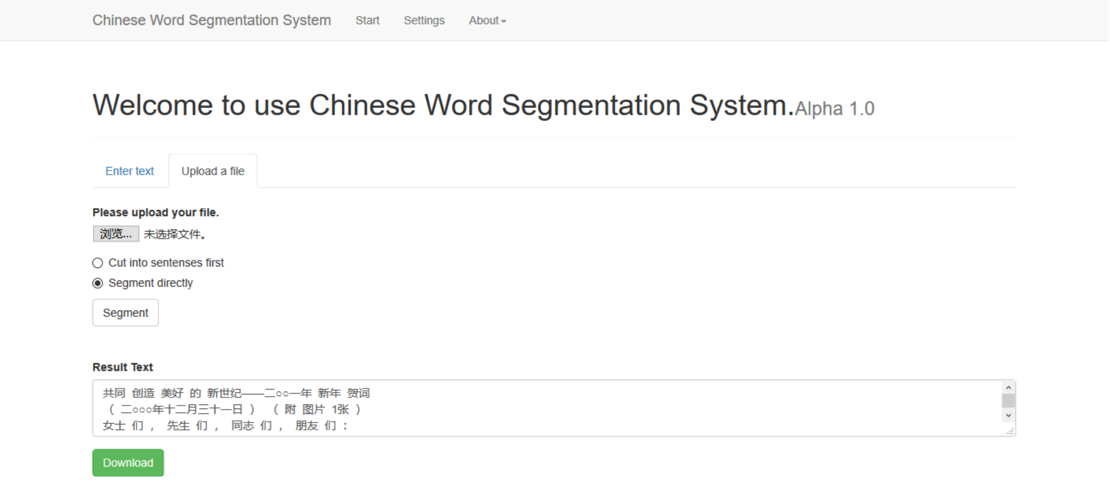


The segmentation of the first 2 sentenses.

今天礼拜天，	今天 礼拜天 ，
天气晴朗。	天气 晴朗 。

Figure 3

Also, you can upload a file. But note that it must be encoded in UTF-8. (See Figure 4, Figure 5.)



Welcome to use Chinese Word Segmentation System.Alpha 1.0

Please upload your file.

浏览... 未选择文件。

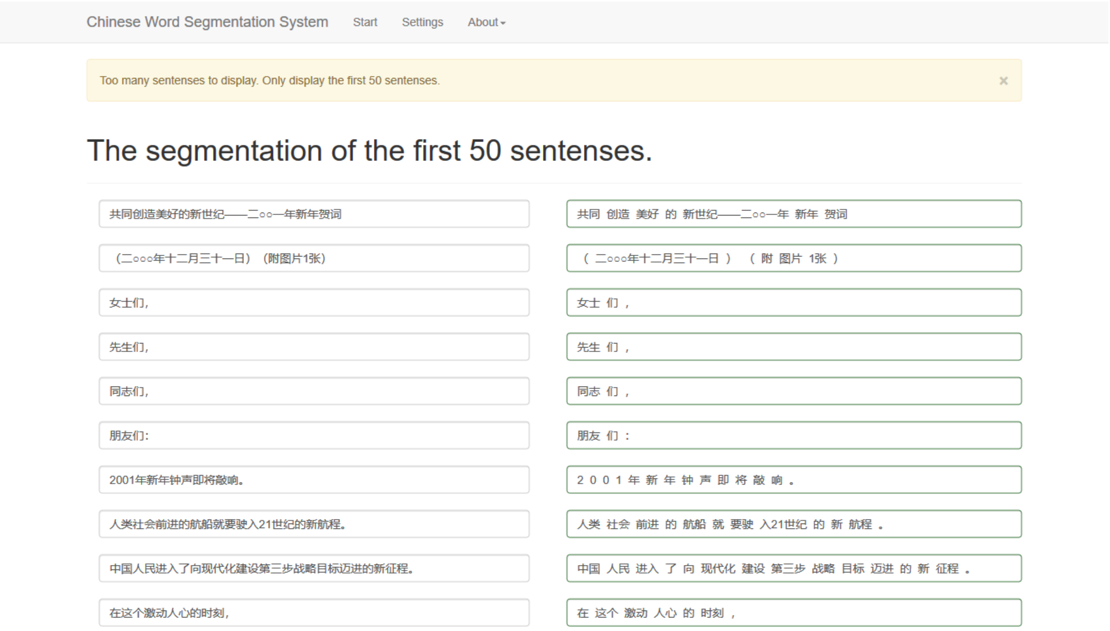
○ Cut into sentenses first

● Segment directly

Result Text

```
共同 创造 美好 的 新世纪——二○○一年 新年 贺词
（ 二○○○年十二月三十一日 ） （ 附 图片 1张 ）
女士 们 ， 先生 们 ， 同志 们 ， 朋友 们 ：
```

Figure 4



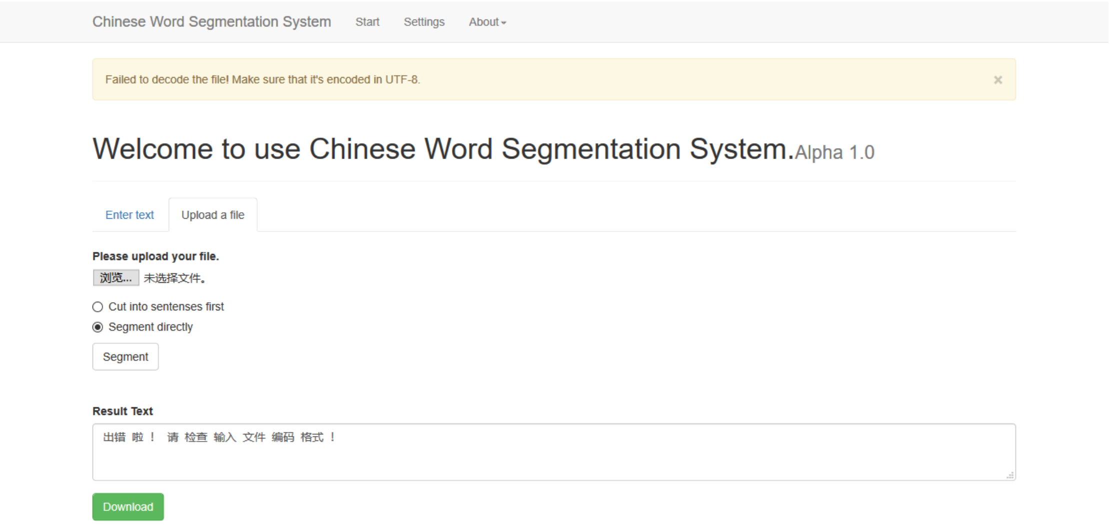
Too many sentenses to display. Only display the first 50 sentenses.

The segmentation of the first 50 sentenses.

共同创造美好的新世纪——二○○一年新年贺词	共同 创造 美好 的 新世纪——二○○一年 新年 贺词
（二○○○年十二月三十一日）（附图片1张）	（ 二○○○年十二月三十一日 ） （ 附 图片 1张 ）
女士们，	女士 们 ，
先生们，	先生 们 ，
同志们，	同志 们 ，
朋友们：	朋友 们 ：
2001年新年钟声即将敲响。	2 0 0 1 年 新 年 钟 声 即 将 敲 响 。
人类社会前进的航船就要驶入21世纪的新航程。	人类 社会 前进 的 航船 就 要驶 入21世纪 的 新 航程 。
中国人民进入了向现代化建设第三步战略目标迈进的新征程。	中国 人民 进入 了 向 现代化 建设 第三步 战略 目标 迈进 的 新 征程 。
在这个激动人心的时刻，	在 这个 激动 人心 的 时刻 ，

Figure 5

If your file is not encoded in UTF-8, the program won’t work. (See Figure 6, Figure 7.)



Failed to decode the file! Make sure that it's encoded in UTF-8.

Welcome to use Chinese Word Segmentation System.Alpha 1.0

Please upload your file.

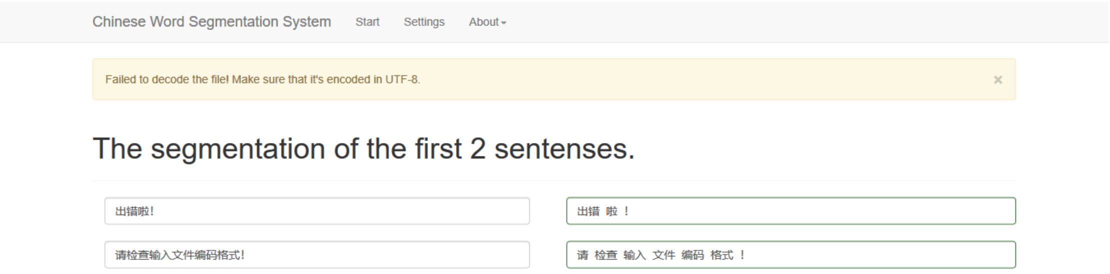
浏览... 未选择文件。

○ Cut into sentenses first
● Segment directly

Result Text

出错 啦 ！ 请 检查 输入 文件 编码 格式 ！

Figure 6



Failed to decode the file! Make sure that it's encoded in UTF-8.

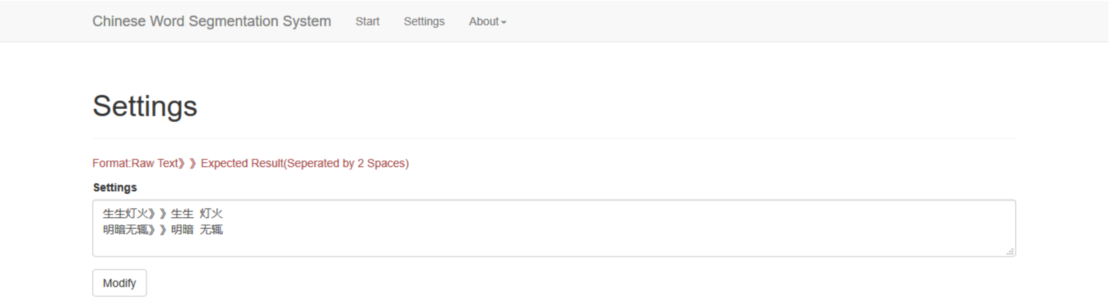
The segmentation of the first 2 sentenses.

出错啦！　出错 啦 ！

请检查输入文件编码格式！　请 检查 输入 文件 编码 格式 ！

Figure 7

Users can modify the settings to correct the segmentations. (See Figure 8, Figure 9.)



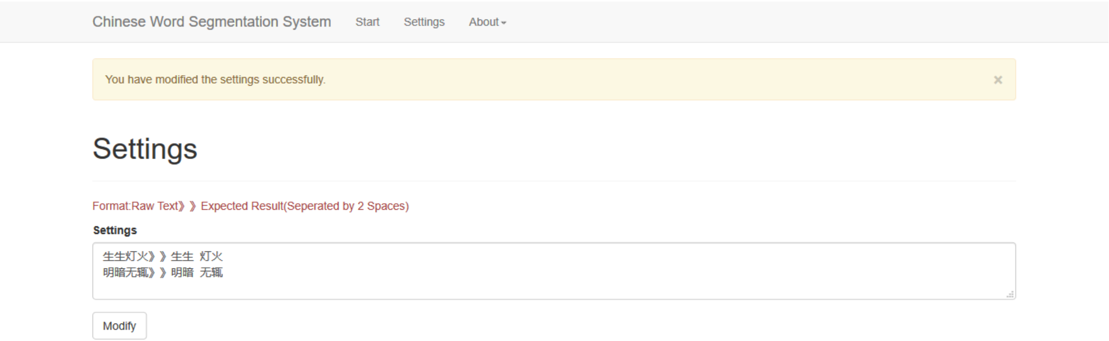
Settings

Format:Raw Text》》Expected Result(Seperated by 2 Spaces)

Settings

生生灯火》》生生 灯火
明暗无辄》》明暗 无辄

Figure 8



You have modified the settings successfully.

Settings

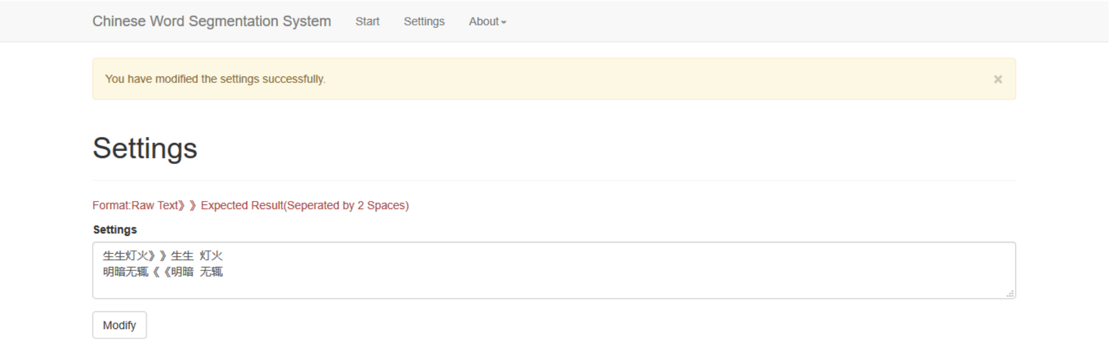
Format:Raw Text》》Expected Result(Seperated by 2 Spaces)

Settings

生生灯火》》生生 灯火
明暗无辄》》明暗 无辄

Figure 9

But if the modified settings are invalid, for example, in a wrong format, they will be ignored. (See Figure 10, Figure 11.)



You have modified the settings successfully.

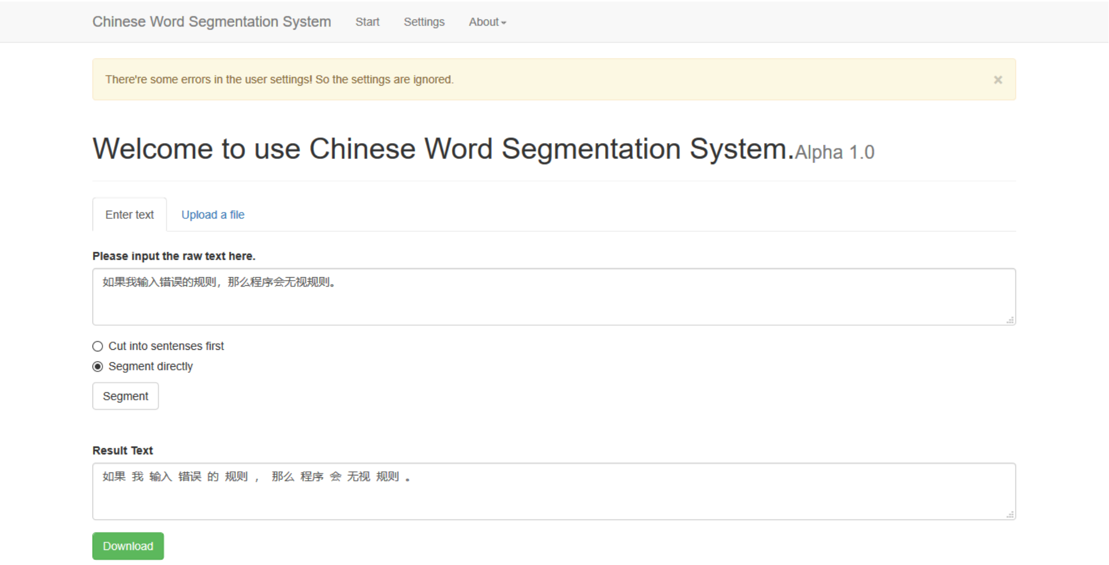
Settings

Format:Raw Text》》Expected Result(Seperated by 2 Spaces)

Settings

生生灯火》》生生 灯火
明暗无辄《《明暗 无辄

Figure 10



There're some errors in the user settings! So the settings are ignored. ×

Welcome to use Chinese Word Segmentation System.Alpha 1.0

Please input the raw text here.

如果我输入错误的规则，那么程序会无视规则。

○ Cut into sentenses first

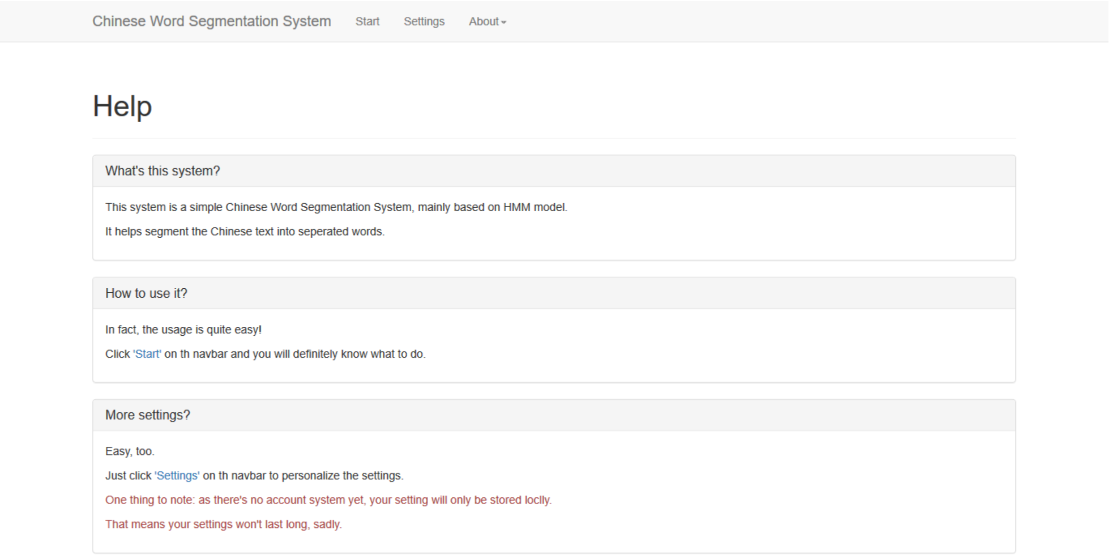
● Segment directly

Result Text

如果 我 输入 错误 的 规则 ， 那么 程序 会 无视 规则 。

Figure 11

And the pages of Help and Copyright look like these. (See Figure 12, Figure 13.)



Help

What's this system?

This system is a simple Chinese Word Segmentation System, mainly based on HMM model.

It helps segment the Chinese text into seperated words.

How to use it?

In fact, the usage is quite easy!

Click 'Start' on th navbar and you will definitely know what to do.

More settings?

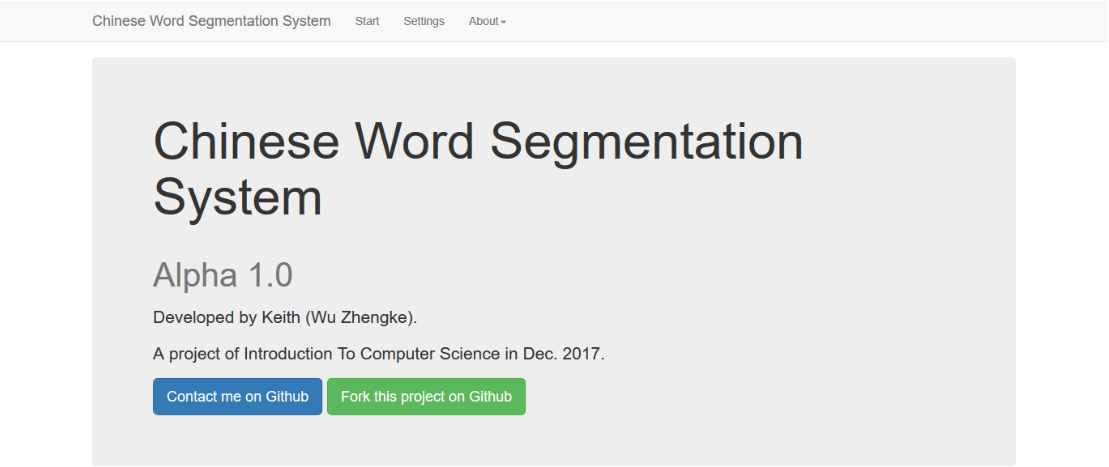
Easy, too.

Just click 'Settings' on th navbar to personalize the settings.

One thing to note: as there's no account system yet, your setting will only be stored loclly.

That means your settings won't last long, sadly.

Figure 12



Chinese Word Segmentation System

Chinese Word Segmentation System

Alpha 1.0

Developed by Keith (Wu Zhengke).

A project of Introduction To Computer Science in Dec. 2017.

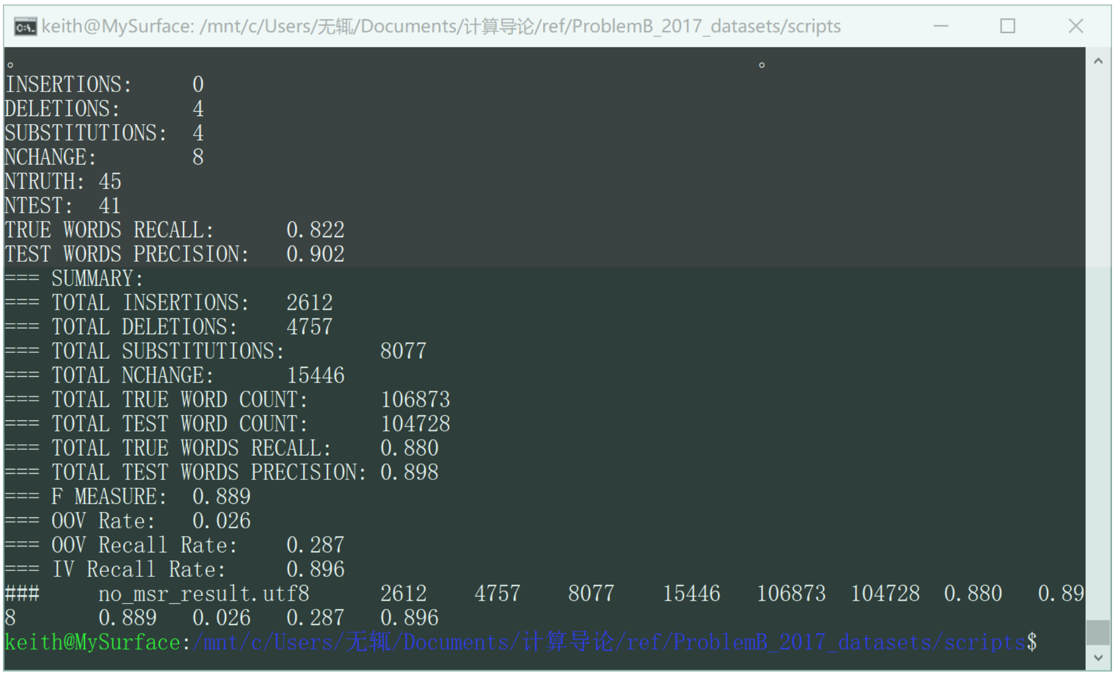
Figure 13

5.2 Test Result:

Using the HMM model trained with given training data (msr_training.utf8 and pku_training.utf8), the program segments the two testing data. The F1 Scores of this program, this program (only HMM) and Jieba (a famous Python Chinese word segmentation module) are listed as follows:

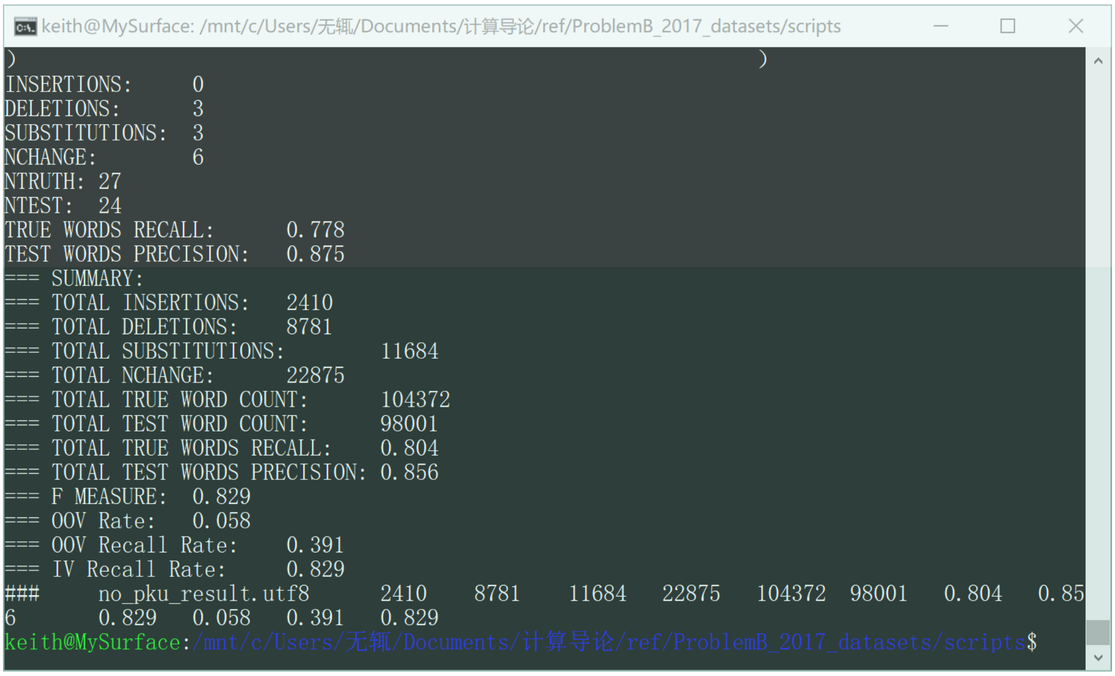
	msr_test.utf8	pku_test.utf8
This program	0.889(Figure 14)	0.829(Figure 15)
This program (only HMM)	0.793(Figure 16)	0.763(Figure 17)
Jieba	0.815(Figure 18)	0.818(Figure 19)

And for detailed results, the following figures show the complete test information.


keith@MySurface: /mnt/c/Users/无粡/Documents/计算导论/ref/ProblemB_2017_datasets/scripts

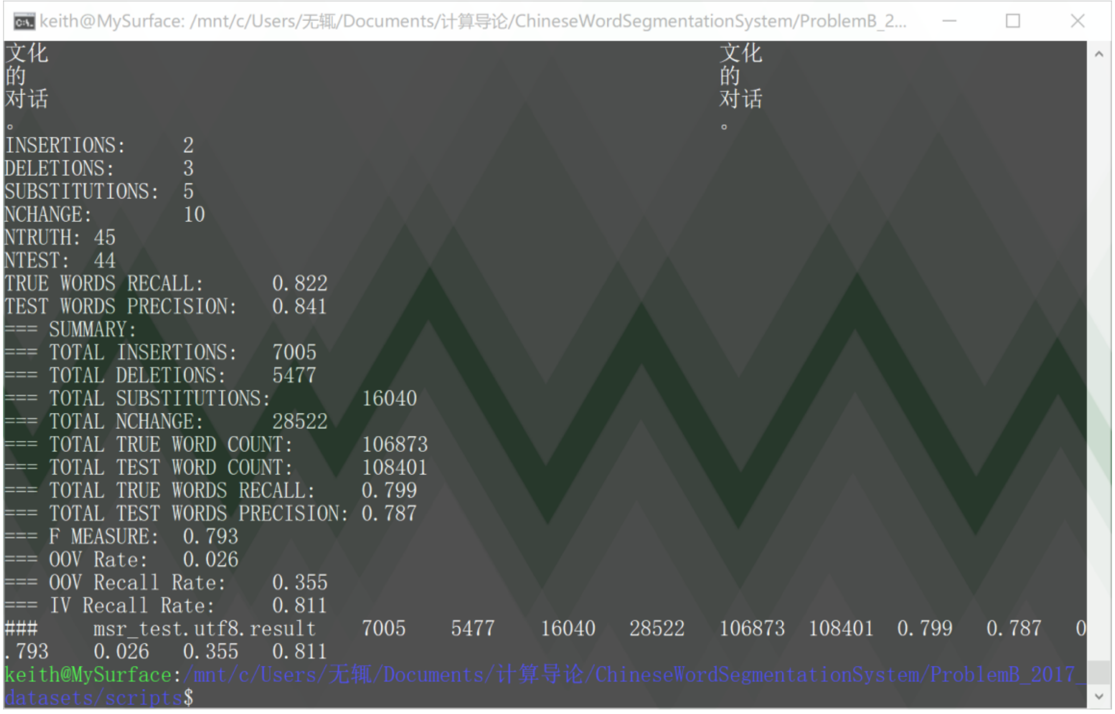
```
。                                                                。
INSERTIONS:     0
DELETIONS:      4
SUBSTITUTIONS:  4
NCHANGE:        8
NTRUTH: 45
NTEST:  41
TRUE WORDS RECALL:      0.822
TEST WORDS PRECISION:   0.902
=== SUMMARY:
=== TOTAL INSERTIONS:   2612
=== TOTAL DELETIONS:    4757
=== TOTAL SUBSTITUTIONS:        8077
=== TOTAL NCHANGE:      15446
=== TOTAL TRUE WORD COUNT:      106873
=== TOTAL TEST WORD COUNT:      104728
=== TOTAL TRUE WORDS RECALL:    0.880
=== TOTAL TEST WORDS PRECISION: 0.898
=== F MEASURE:  0.889
=== OOV Rate:   0.026
=== OOV Recall Rate:    0.287
=== IV Recall Rate:     0.896
###     no_msr_result.utf8      2612    4757    8077    15446   106873  104728  0.880   0.89
8       0.889   0.026   0.287   0.896
keith@MySurface:/mnt/c/Users/无粡/Documents/计算导论/ref/ProblemB_2017_datasets/scripts$
```

Figure 14


keith@MySurface: /mnt/c/Users/无粡/Documents/计算导论/ref/ProblemB_2017_datasets/scripts

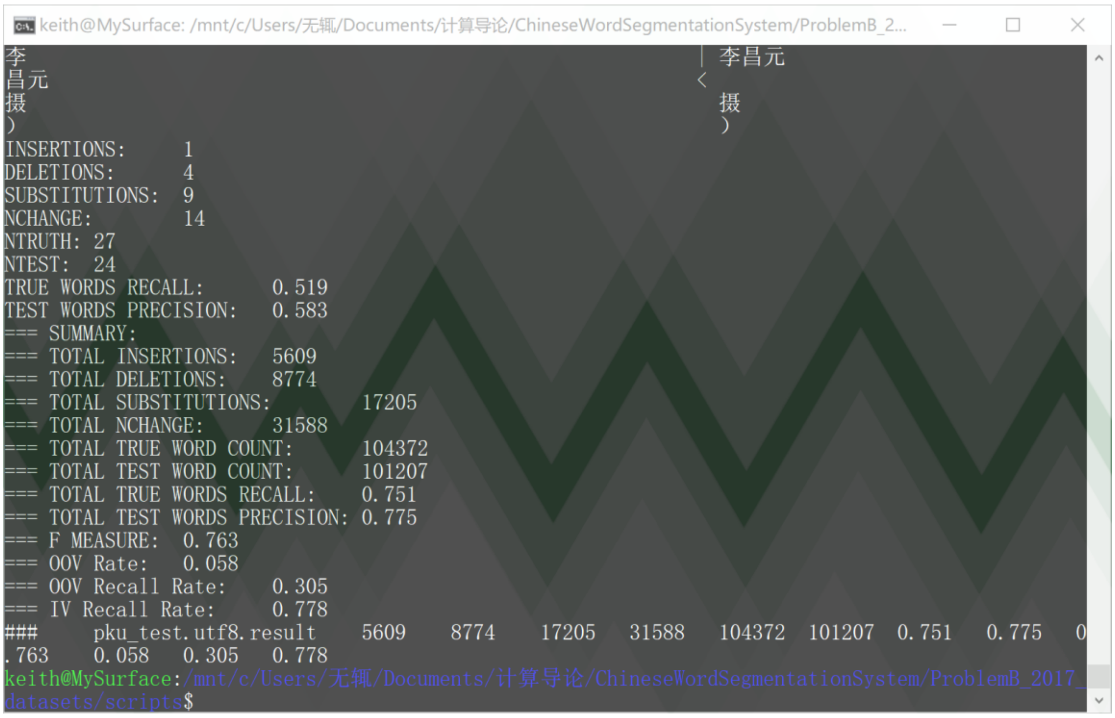
```
)                                                                 )
INSERTIONS:     0
DELETIONS:      3
SUBSTITUTIONS:  3
NCHANGE:        6
NTRUTH: 27
NTEST:  24
TRUE WORDS RECALL:      0.778
TEST WORDS PRECISION:   0.875
=== SUMMARY:
=== TOTAL INSERTIONS:   2410
=== TOTAL DELETIONS:    8781
=== TOTAL SUBSTITUTIONS:        11684
=== TOTAL NCHANGE:      22875
=== TOTAL TRUE WORD COUNT:      104372
=== TOTAL TEST WORD COUNT:      98001
=== TOTAL TRUE WORDS RECALL:    0.804
=== TOTAL TEST WORDS PRECISION: 0.856
=== F MEASURE:  0.829
=== OOV Rate:   0.058
=== OOV Recall Rate:    0.391
=== IV Recall Rate:     0.829
###     no_pku_result.utf8      2410    8781    11684   22875   104372  98001   0.804   0.85
6       0.829   0.058   0.391   0.829
keith@MySurface:/mnt/c/Users/无粡/Documents/计算导论/ref/ProblemB_2017_datasets/scripts$
```

Figure 15


keith@MySurface: /mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2...

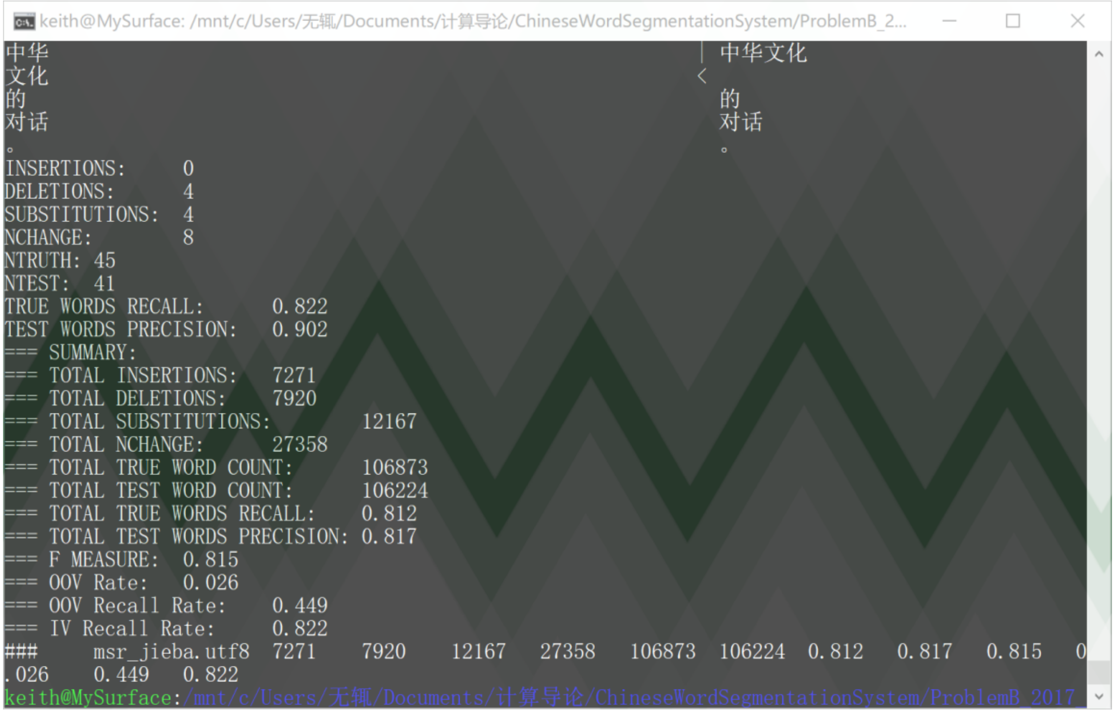
```
文化                                                            文化
的                                                              的
对话                                                            对话
。                                                              。
INSERTIONS:     2
DELETIONS:      3
SUBSTITUTIONS:  5
NCHANGE:        10
NTRUTH: 45
NTEST:  44
TRUE WORDS RECALL:      0.822
TEST WORDS PRECISION:   0.841
=== SUMMARY:
=== TOTAL INSERTIONS:   7005
=== TOTAL DELETIONS:    5477
=== TOTAL SUBSTITUTIONS:        16040
=== TOTAL NCHANGE:      28522
=== TOTAL TRUE WORD COUNT:      106873
=== TOTAL TEST WORD COUNT:      108401
=== TOTAL TRUE WORDS RECALL:    0.799
=== TOTAL TEST WORDS PRECISION: 0.787
=== F MEASURE:  0.793
=== OOV Rate:   0.026
=== OOV Recall Rate:    0.355
=== IV Recall Rate:     0.811
###     msr_test.utf8.result    7005    5477    16040   28522   106873  108401  0.799   0.787   0
.793    0.026   0.355   0.811
keith@MySurface:/mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2017_
datasets/scripts$
```

Figure 16


keith@MySurface: /mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2...

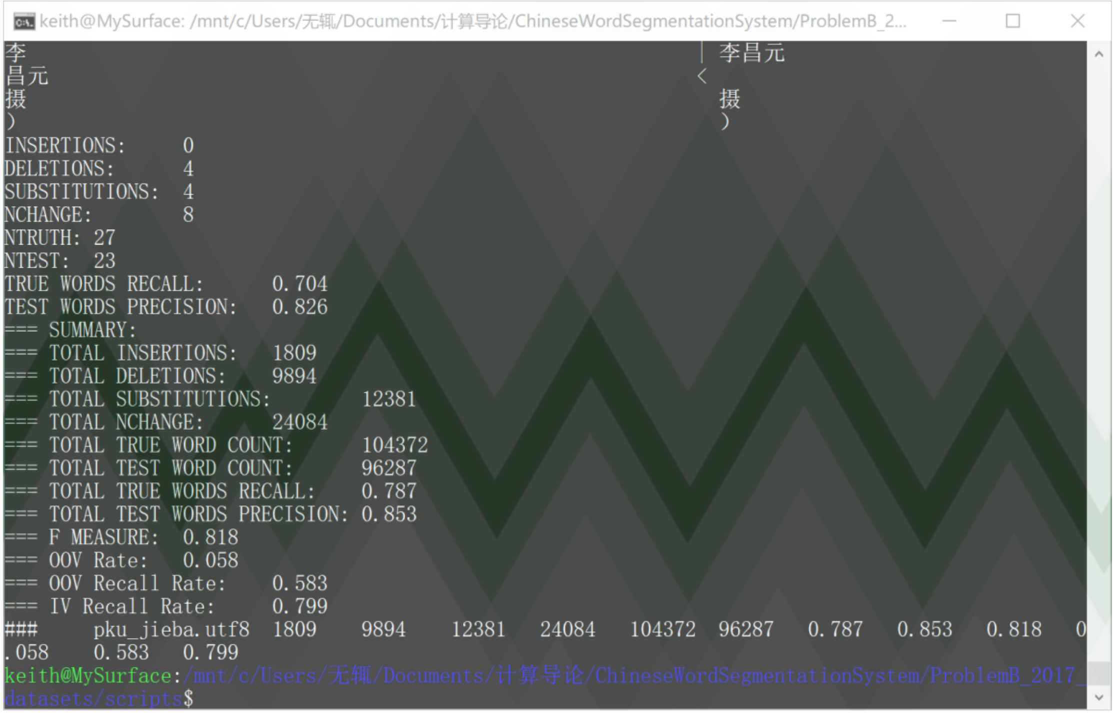
```
李                                                            | 李昌元
昌元                                                          <
摄                                                              摄
)                                                               )
INSERTIONS:     1
DELETIONS:      4
SUBSTITUTIONS:  9
NCHANGE:        14
NTRUTH: 27
NTEST:  24
TRUE WORDS RECALL:      0.519
TEST WORDS PRECISION:   0.583
=== SUMMARY:
=== TOTAL INSERTIONS:   5609
=== TOTAL DELETIONS:    8774
=== TOTAL SUBSTITUTIONS:        17205
=== TOTAL NCHANGE:      31588
=== TOTAL TRUE WORD COUNT:      104372
=== TOTAL TEST WORD COUNT:      101207
=== TOTAL TRUE WORDS RECALL:    0.751
=== TOTAL TEST WORDS PRECISION: 0.775
=== F MEASURE:  0.763
=== OOV Rate:   0.058
=== OOV Recall Rate:    0.305
=== IV Recall Rate:     0.778
###     pku_test.utf8.result    5609    8774    17205   31588   104372  101207  0.751   0.775   0
.763    0.058   0.305   0.778
keith@MySurface:/mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2017_
datasets/scripts$
```

Figure 17


keith@MySurface: /mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2...

```
中华                                                              | 中华文化
文化                                                              <
的                                                                  的
对话                                                                对话
。                                                                  。
INSERTIONS:     0
DELETIONS:      4
SUBSTITUTIONS:  4
NCHANGE:        8
NTRUTH: 45
NTEST:  41
TRUE WORDS RECALL:      0.822
TEST WORDS PRECISION:   0.902
=== SUMMARY:
=== TOTAL INSERTIONS:   7271
=== TOTAL DELETIONS:    7920
=== TOTAL SUBSTITUTIONS:        12167
=== TOTAL NCHANGE:      27358
=== TOTAL TRUE WORD COUNT:      106873
=== TOTAL TEST WORD COUNT:      106224
=== TOTAL TRUE WORDS RECALL:    0.812
=== TOTAL TEST WORDS PRECISION: 0.817
=== F MEASURE:  0.815
=== OOV Rate:   0.026
=== OOV Recall Rate:    0.449
=== IV Recall Rate:     0.822
###     msr_jieba.utf8  7271    7920    12167   27358   106873  106224  0.812   0.817   0.815   0
.026    0.449   0.822
keith@MySurface:/mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2017
```

Figure 18


keith@MySurface: /mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2...

```
李                                                                | 李昌元
昌元                                                              <
摄                                                                  摄
)                                                                   )
INSERTIONS:     0
DELETIONS:      4
SUBSTITUTIONS:  4
NCHANGE:        8
NTRUTH: 27
NTEST:  23
TRUE WORDS RECALL:      0.704
TEST WORDS PRECISION:   0.826
=== SUMMARY:
=== TOTAL INSERTIONS:   1809
=== TOTAL DELETIONS:    9894
=== TOTAL SUBSTITUTIONS:        12381
=== TOTAL NCHANGE:      24084
=== TOTAL TRUE WORD COUNT:      104372
=== TOTAL TEST WORD COUNT:      96287
=== TOTAL TRUE WORDS RECALL:    0.787
=== TOTAL TEST WORDS PRECISION: 0.853
=== F MEASURE:  0.818
=== OOV Rate:   0.058
=== OOV Recall Rate:    0.583
=== IV Recall Rate:     0.799
###     pku_jieba.utf8  1809    9894    12381   24084   104372  96287   0.787   0.853   0.818   0
.058    0.583   0.799
keith@MySurface:/mnt/c/Users/无辄/Documents/计算导论/ChineseWordSegmentationSystem/ProblemB_2017_
datasets/scripts$
```

Figure 19

6. Conclusion

In comparison with the F1 Score of Jieba, it’s evident that this program with only HMM is not good enough in terms of the accuracy of segmentation, not to mention other effective algorithms. To solve this problem, I thought and tried a lot. In the process, I have a much deeper insight into different algorithms about NLP and Chinese Word Segmentation.

In fact, when doing this project, I was ‘forced’ to learn plenty of new things, and that’s exactly the meaning of this project. For example, when dealing with the segmentation, I spent lots of time learning the HMM and Viterbi Algorithm.

And as for the part of User Interface, I learned a completely new skill: Python Web Development with Flask. To implement a relatively beautiful website, I also learned the basic knowledge of HTML, CSS and the web frame Twitter Bootstrap, which took me more than a week’s time in total.

In conclusion, doing this project improves my coding ability greatly and provides me with opportunities to learn new things.

7. Reference

When doing this project, I refer to quite a few articles on the Internet and some books. Some of them are listed as follows:

- 中文分词的 python 实现-基于 HMM 算法 - CSDN 博客
- 中文分词之 HMM 模型详解 - CSDN 博客
- Flask Web Development: Developing Web Applications with Python (Miguel Grinberg)
- Flask Documentation (0.12)
- Bootstrap 教程 | 菜鸟教程
- 中文分词入门之最大匹配法 | 我爱自然语言处理

With my sincere gratitude!