Final Report

Team Name:Pyson

Team Leader:ZhouBen

Date: 2015-12-24

1.Prototype System Introduction

1.1 Functions

1.2 Running Environment

Windows 7 or later

Mac OSX or later

1.3 Developing Environment

PyScripter 2.6.0

PyCharm 4.5.4

Github

2.Task Allocation

Yuan Meng:

GUI design and implementation

Hu Guo Shen:

Core algorithm

Zhou Ben:

Overall design

User Input processing

Testing

In the view of file:

Yuan Meng:

View.py

Hu Guo Shen:

package kernel

Zhou Ben:  
 test.py

Controller.py

User input processing part of View.py

In the view of MVC:

Yuan Meng: View

Hu Guo Shen: Model

Zhou Ben: Controller

3.System Architecture

3.1 Introduction

We develop this prototype system under the guiding ideology of MVC.

MVC is a classic design pattern firstly used in user interface design,but now its idea has been used in a much wider range.MVC design pattern decouples different functions in the system.Its clear separation not only improves the readability and maintainability of the system but also forms a duty division naturally.

MVC decomposes a typical system into three parts:model,view and controller.

3.2 Model

Model is the core of the system encapsulating the essential data and service logic.

In our case,data and service logic denote the lexicon and the algorithm used to perform segmentation.

In the view of code,the package *kernel* in our source code is responsible for the model part including:

1.*\_\_init\_\_.py:*

Basic classes definition.

2.*segmentation\_by\_retrieve.py:*

Where segmentation involves special cases where high-priority is required, including quotation identification,terminology and particular cases.

3.*mi.py,dts\_calculate.py,judge.py:*

Where typical steps of word segmentation is done.

*4*.*test.py*

Test logic and implementation

5.some txt files

Includes lexicon,corpora,test answers and so on.

The detail of algorithm will be discussed later.

3.3 View

View is the manager of GUI which enables user to input as well as demonstrate result to them.

In our code,the file *view.py* contains this part,in which we use python standard library tkinter to construct interface.

User permission:

Interface design:

Snapshots:(暂缺，等全做完了补上)

3.4 Controller

Controller is the bridge between view and model.It interprets the inputs from view from human-compatible content into machine-compatible content which is a call to a particular function of model in most cases.When model returns a value,it need to reformat it into human-compatible stuff.

In our program,*controller.py* implements main of the controller.But because of the character of tkinter,there is a tight coupling between the user input processing and interface constructing,resulting in that a part of *view.py* also works for controller.

Controller is a rather important part which controls the path on which the system works.

But in our case,the necessary communication between view and model is relatively simple,so it only serves as a formatter and a middle station between view and model.

3.5 Workflow

(暂缺)

4.Algorithm Description

5.Demo and Testing Result

5.1 Screenshots

5.2 Testing Procedure,Data and Result

6.Conclusion

Masterpiece!