Markov Chain Algorithm Project

Animal Logic

Tony Li

Nov 2014

# Index

[Index 2](#_Toc404246826)

[1. Requirement 3](#_Toc404246827)

[1.1 Backend 3](#_Toc404246828)

[1.2 Frontend 3](#_Toc404246829)

[2. Project management 3](#_Toc404246830)

[2.1 Plan 3](#_Toc404246831)

[2.2 Milestone 3](#_Toc404246832)

[2.2.1 Complete first runnable version(10:30 19 Nov 2014) 3](#_Toc404246833)

[2.2.2 Modify the simple html page (12:30 19 Nov 2014) 4](#_Toc404246834)

[2.2.3 Integrate (11:10 20 Nov 2014) 4](#_Toc404246835)

[2.2.4 Make the web page runnable (11:30 20 Nov 2014) 4](#_Toc404246836)

[3. Design 4](#_Toc404246837)

[3.1 System Design 4](#_Toc404246838)

[3.1.1 Backend 4](#_Toc404246839)

[3.1.2 Frontend 4](#_Toc404246840)

[3.2 Algorithm Design 5](#_Toc404246841)

[3.2.1 Add words set 5](#_Toc404246842)

[3.2.2 Select teacher 5](#_Toc404246843)

[3.2.3 Write poem 5](#_Toc404246844)

[3.3 UI Design 6](#_Toc404246845)

[4. Reference 6](#_Toc404246846)

[4.1 Markov Chain 6](#_Toc404246847)

[4.2 Poems 6](#_Toc404246848)

[4.3 UI design 6](#_Toc404246849)

# Requirement

## Backend

Implement a Java application to allow a user to transform the contents of a text file using a Markov chain algorithm. The application should allow the selection of a source text file, and input parameters for the algorithm (typically prefix/suffix parameters).

## Frontend

Also, please design a simple Web UI that could be used for this application. This doesn’t need to be integrated with the Java code and could be sent to us as HTML, or just an image, or Photoshop file. (But if the frontend is a real one and can be integrated with the backend, it is a bonus, right?)

# Project management

## Plan

18 Nov morning learn Algorithm carefully.

18 Nov afternoon implement backend

19 Nov morning implement frontend

19 Nov afternoon integrate.

20 Nov morning test, summarise, refactor code, write doc in detail

20 Nov afternoon to make it more sense, buy [www.fonyli.com](http://www.fonyli.com) from godaddy. Start a new instance in AWS EC2, deploy system on.

## Milestone

https://github.com/FonyLi/tonyli/tree/master/AL/MarkovChain

### Complete first runnable version(10:30 19 Nov 2014)

<https://github.com/FonyLi/tonyli/commit/b62102eade335baddea16633e44d55dffbc17f36>

This version satisfies backend requirement, with a simple html.

Next step:

Add more comments in detail

Implement html more beautiful

### Modify the simple html page (12:30 19 Nov 2014)

<https://github.com/FonyLi/tonyli/commit/3b321bc94f7f9959bf41bcf1300a821b127a19b1>

the web page is reasonable

Next step:

Make the server run as a http server

### Integrate (11:10 20 Nov 2014)

### Make the web page runnable (11:30 20 Nov 2014)

<http://www.fonyli.com/index.htm> can be visited now!

# Design

## System Design

Simple html page + java servlet server

### Backend

Use glassfish grizzly to start a http server.

Handler:

select teacher handler(invoke upload file internally)

write poem handler ( user can input the begin and end of the poem)

### Frontend

Only one page of Simple html

Use jquery ajax to connectbackend.

Use select to select teachers(famous poemts)

## Algorithm Design

Markov Chain Algorithm

<http://books.google.com/books?id=to6M9_dbjosC&pg=PA61&source=gbs_toc_r&cad=4#v=onepage&q&f=false>

### Add words set

for each word

{

Add the next into the current’s next set;

Add the next into the “previous current”’s next set if previous is not null

}

### Select teacher

switch (teacher’s name, basically a famous poet)

{

Case some teacher:

Upload a local file of word set of that poet

}

Alternatively, user can upload file by invoking uploadFile(File), but this function doesn’t need to be supported by web page.

### Write poem

set max length as1000 (will be defined in config in later version)

start from begin,

while(not match end & total length < max length)

{

get next;

add it in the end of poem;

}

Return poem;

## UI Design

You will become a famous POET!

How?

select you poem teacher:\_\_\_\_(

options:

William Shakespeare,

Aleksandr Pushkin

Rabindranath Tagore

Walt Whitman

OK, input the beginning \_\_\_\_and the end\_\_\_ of your poem

(button: write a new poem)

Here is your masterpiece. Well done! You are a famous poet now!

# Reference

## Markov Chain

<http://books.google.com/books?id=to6M9_dbjosC&pg=PA61&source=gbs_toc_r&cad=4#v=onepage&q&f=false>

## Poems

List resources of all the sample txts

* + 1. William Shakespeare

<http://shakespeare.mit.edu/>

<http://shakespeare.mit.edu/romeo_juliet/romeo_juliet.1.1.html>

* + 1. Others

Just use the result from google

## UI design

Ajax doc