

Markovman

Generated by Doxygen 1.8.13

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	src/include/statemach.h File Reference	3
2.1.1	Detailed Description	3
2.2	src/lib/statemach.c File Reference	3
2.2.1	Detailed Description	4
2.3	src/markovman.c File Reference	4
2.3.1	Detailed Description	4
	Index	7

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

src/ markovman.c	
Implementation of markov chains for random text generation	4
src/include/ statemach.h	
Header file for state machines	3
src/lib/ statemach.c	
File implementing state machines	3

Chapter 2

File Documentation

2.1 `src/include/statemach.h` File Reference

Header file for state machines.

2.1.1 Detailed Description

Header file for state machines.

Author

Ian G. Tayler

Date

5 May 2017 (creation)

This exports the names from [lib/statemach.c](#) that we will need in `src/main.c`.

See also

<https://github.com/IanTayler/markovman.git>

2.2 `src/lib/statemach.c` File Reference

File implementing state machines.

2.2.1 Detailed Description

File implementing state machines.

Author

Ian G. Tayler

Date

5 May 2017 (creation)

This is the file where all the action happens. We define the struct 'Word' and a few functions for handling it. That covers most of the program's logic.

See also

<https://github.com/IanTayler/markovman.git>

2.3 src/markovman.c File Reference

Implementation of markov chains for random text generation.

```
#include <stdio.h>
#include "statemach.h"
```

Functions

- int **main** (void)

2.3.1 Detailed Description

Implementation of markov chains for random text generation.

Author

Ian G. Tayler

Date

5 May 2017 (creation)

DESCRIPTION:

Markovman is a program for random text generation based on markov chains. The generator is trained from a corpus. The only supported format for the corpus is as a text file, with dots '.' separating sentences.

This is the main file fo the program, where the (command line) interface is implemented.

USAGE:

The following is the interface as I plan to implement it, although it hasn't been written yet. The easiest way to use Markovman is to call it together with a corpus-file.

```
markovman path/to/corpus.txt
```

That will put the program in a loop, reading from stdin. You can pass the following commands:

```
gen N
```

will generate N sentences one after the other based on the corpus.

```
kill X
```

will make the word X disappear from the corpus.

```
exit
```

will exit the program

Another possibility is running the program like the following, which will generate N sentences and close immediately.

```
markovman path/to/corpus.txt -n N
```

See also

<https://github.com/IanTayler/markovman.git>

Index

src/include/statemach.h, [3](#)
src/lib/statemach.c, [3](#)
src/markovman.c, [4](#)