naivebayescpp

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 NewsItem Struct Reference

Contains information for a single news item.

`#include <NewsItem.h>`

Public Attributes

- std::string **path**
- std::string collection

  holds the path to the information, whether a file path or web path
- std::map< std::string, size_t > word_count

  The classification of the news item set as empty or unknown when unknown.
- std::string contents

  The count (per word) of the contents of the news item.

2.1.1 Detailed Description

Contains information for a single news item.

The documentation for this struct was generated from the following file:

- /home/aherrera/CLionProjects/naivebayescpp/src/NewsItem.h

2.2 Parser Class Reference

The Parser that parses newsgroup data.

`#include <Parser.h>`

Public Member Functions

- Parser (const std::string &path)
- Parser (std::vector< NewsItem > itms)
- WordMatrix getMatrix ()
- std::vector< std::string > get_classes ()
- std::vector< NewsItem > get_items ()
- void prune_per_class (size_t max_per_classes)
- void prune_per_class (uint_fast32_t random_seed, size_t max_per_classes)
- void prune_per_class (const std::map< std::string, size_t > &class_count)
- void prune_per_class (uint_fast32_t random_seed, const std::map< std::string, size_t > &class_count)
- Parser get_items_of_classes (const std::vector< std::string > &classes)
- Parser prune_classes (size_t n)
- void shuffle ()

2.2.1 Detailed Description

The Parser that parses newsgroup data.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 Parser() [1/2]

```
Parser::Parser (
            const std::string & path )  [explicit]
```

Constructs the parser, parses the data into an intermediary format.

Parameters

path	The path to the newgroup root path

2.2.2.2 Parser() [2/2]

```
Parser::Parser (
            std::vector< NewsItem > itms )  [explicit]
```

Creates a parser from already parsed items

Parameters

itms	The items

2.2.3 Member Function Documentation

2.2.3.1 get_classes()

```
vector< string > Parser::get_classes ( )
```

Gets the classes available in the data

Returns

The classes in a vector of strings

2.2.3.2 get_items()

```
vector< NewsItem > Parser::get_items ( )
```

Get the NewsItems from the parser

Returns

A vector of NewsItem objects

2.2.3.3 get_items_of_classes()

```
Parser Parser::get_items_of_classes (
            const std::vector< std::string > & classes )
```

This function returns a Parser object with only those items of classes provided

Parameters

classes	The classes as a vector of strings

Returns

The resulting Parser with only the classes provided

2.2.3.4 getMatrix()

```
WordMatrix Parser::getMatrix ( )
```

Returns the data in a word matrix, useful for analysis and classification

Returns

A WordMatrix object

2.2.3.5 prune_classes()

```
Parser Parser::prune_classes (
            size_t n )
```

Prunes the number of classes randomly to n

Parameters

n	The number of classes in the resulting WordMatrix

Returns

A WordMatrix with n classes

2.2.3.6 prune_per_class() [1/4]

```
void Parser::prune_per_class (
            size_t max_per_classes )
```

Gets the maximum on a per class basis

Parameters

max_per_classes	The max

2.2.3.7 prune_per_class() [2/4]

```
void Parser::prune_per_class (
            uint_fast32_t random_seed,
            size_t max_per_classes )
```

Gets the maximum on a per class basis with a preset random seed

Parameters

random_seed	The random seed
max_per_class	The max

2.2.3.8 prune_per_class() [3/4]

```
void Parser::prune_per_class (
            const std::map< std::string, size_t > & class_count )
```

Removes the number of items per class

Parameters

class_count	A map where the key is the class and the value is the number of items in that class to keep

2.2.3.9 prune_per_class() [4/4]

```
void Parser::prune_per_class (
            uint_fast32_t random_seed,
            const std::map< std::string, size_t > & class_count )
```

Removes a number of items per class

Parameters

random_seed	The random seed
class_count	A map where the key is the class and the value is the number of items in that class to keep

2.2.3.10 shuffle()

```
void Parser::shuffle ( )
```

Randomly shuffles NewsItems

The documentation for this class was generated from the following files:

- /home/aherrera/CLionProjects/naivebayescpp/src/Parser.h
- /home/aherrera/CLionProjects/naivebayescpp/src/Parser.cpp

2.3 WebHandler Class Reference

Uses the libcurl library to interface with newsapi.org to obtain the leading news.

`#include <WebHandler.h>`

Public Member Functions

- WebHandler ()
- WebHandler (std::string key)
- std::vector< NewsItem > sendQuery (const std::string &collection)
- std::vector< NewsItem > sendQueries (const std::vector< std::string > &collections)
- void setKey (const std::string &key)
- std::string getKey ()

2.3.1 Detailed Description

Uses the libcurl library to interface with newsapi.org to obtain the leading news.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 WebHandler() [1/2]

```
WebHandler::WebHandler ( )
```

Default constructor, gets a json config file with key

2.3.2.2 WebHandler() [2/2]

```
WebHandler::WebHandler (
            std::string key )  [explicit]
```

Constructs the WebHandler class

Parameters

key	The api key for the newsapi.org site

2.3.3 Member Function Documentation

2.3.3.1 getKey()

```
string WebHandler::getKey ( )
```

Gets the newsapi.org api key

Returns

The news api key

2.3.3.2 sendQueries()

```
std::vector< NewsItem > WebHandler::sendQueries (
            const std::vector< std::string > & collections )
```

Send queries to newsapi for the top 20 per class in vector of classes collection

Parameters

collections	The classes

Returns

A vector of NewsItem objects

2.3.3.3 sendQuery()

```
vector< NewsItem > WebHandler::sendQuery (
            const std::string & collection )
```

Sends query to newsapi for the top 20 of class collection

Parameters

collection	The class

Returns

A vector of NewsItem objects all of class collection

2.3.3.4 setKey()

```
void WebHandler::setKey (
            const std::string & key )
```

Set the newapi.org api key

Parameters

key	newsapi key

The documentation for this class was generated from the following files:

- /home/aherrera/CLionProjects/naivebayescpp/src/WebHandler.h
- /home/aherrera/CLionProjects/naivebayescpp/src/WebHandler.cpp

2.4 WordMatrix Class Reference

Uses an Eigen matrix to store a words count per class and do arithmetic with.

`#include <WordMatrix.h>`

Public Member Functions

- WordMatrix (const std::vector< NewsItem > &items)
- WordMatrix (MatrixXi word_count, std::map< std::string, size_t > classes, std::map< std::string, size_t > words)
- size_t getTotalWords ()
- size_t getClassTotal (const std::string &cls)
- size_t getWordTotal (const std::string &word)
- size_t getClassTotal (const std::vector< std::string > &clss)
- size_t getWordTotal (const std::vector< std::string > &words)
- size_t & getCount (const std::string &clss, const std::string &word)
- size_t & getCount (size_t i, size_t j)
- size_t getCount (const std::vector< std::string > &cls, const std::string &word)
- size_t getCount (const std::string &cls, const std::vector< std::string > &wrds)
- size_t getCount (const std::vector< std::string > &clss, const std::vector< std::string > &wrds)
- WordMatrix block (const std::vector< std::string > &clss, const std::vector< std::string > &wrds)
- WordMatrix block (const std::vector< std::string > &clss)
- WordMatrix prune_classes (size_t n)
- void printProbabilities (std::ostream &ostr)
- void printFrequency (std::ostream &ostr)
- void printProbabilities ()
- void printFrequency ()
- void printLatexProbabilities (std::ostream &ostr)
- void printLatexFrequency (std::ostream &ostr)
- void printLatexProbabilities ()
- void printLatexFrequency ()
- WordMatrix getMostFrequent (size_t n)
- std::string predict (const NewsItem &itm)
- std::vector< std::string > getWords ()
- std::vector< std::string > getClasses ()
- size_t word_index (const std::string &word)
- size_t class_index (const std::string &cls)

2.4.1 Detailed Description

Uses an Eigen matrix to store a words count per class and do arithmetic with.

2.4.2 Constructor & Destructor Documentation

2.4.2.1 WordMatrix() [1/2]

```
WordMatrix::WordMatrix (
            const std::vector< NewsItem > & items )  [explicit]
```

Constructs the word matrix from a vector of NewsItem objects

Parameters

items	A vector of NewsItems

2.4.2.2 WordMatrix() [2/2]

```
WordMatrix::WordMatrix (
            MatrixXi word_count,
            std::map< std::string, size_t > classes,
            std::map< std::string, size_t > words )
```

Constructs the word matrix from raw data

Parameters

word_count	An Eigen Matrix representing the word count
classes	A map of classes where the key is the class and the value is the column index of the matrix
words	A map of words where the key is the word and the value is the row index of the matrix

2.4.3 Member Function Documentation

2.4.3.1 block() [1/2]

```
WordMatrix WordMatrix::block (
            const std::vector< std::string > & clss,
            const std::vector< std::string > & wrds )
```

Gets a submatrix of the matrix with the specified classes and words

Parameters

clss	The set of classes in a vector of strings
wrds	The set of words in a vector of strings

Returns

A sub-WordMatrix

2.4.3.2 block() [2/2]

```
WordMatrix WordMatrix::block (
            const std::vector< std::string > & clss )
```

Gets a submatrix of the matrix with the specified classes

Parameters

clss	The set of classes in a vector of strings

Returns

A sub-WordMatrix

2.4.3.3 class_index()

```
size_t WordMatrix::class_index (
            const std::string & cls )
```

Gets the index, in the WordMatrix, of a class

Parameters

cls	The class

Returns

The WordMatrix's index

2.4.3.4 getClasses()

```
vector< string > WordMatrix::getClasses ( )
```

Gets the classes in the WordMatrix

Returns

The classes in the WordMatrix as a vector of strings

2.4.3.5 getClassTotal() [1/2]

```
size_t WordMatrix::getClassTotal (
            const std::string & cls )
```

Gets the total words in a class

Parameters

cls	The class to sum the words for

Returns

The total words with class cls

2.4.3.6 getClassTotal() [2/2]

```
size_t WordMatrix::getClassTotal (
            const std::vector< std::string > & clss )
```

Gets the total words for a set of classes

Parameters

clss	A vector of the classes to sum for

Returns

The total words present in the classes provided

2.4.3.7 getCount() [1/5]

```
size_t & WordMatrix::getCount (
            const std::string & clss,
            const std::string & word )
```

Gets the number of occurrences of a word in a class

Parameters

clss	The class
word	The word

Returns

The count

2.4.3.8 getCount() [2/5]

```
size_t& WordMatrix::getCount (
            size_t i,
            size_t j )
```

Gets the number of occurrences of a word in class

Parameters

i	The ith column in the WordMatrx
j	The jth row in the WordMatrix

Returns

The count

2.4.3.9 getCount() [3/5]

```
size_t WordMatrix::getCount (
            const std::vector< std::string > & cls,
            const std::string & word )
```

Gets the number of occurences of a word in a set of classes

Parameters

cls	The set of classes in a vector of strings
word	The word

Returns

The count

2.4.3.10 getCount() [4/5]

```
size_t WordMatrix::getCount (
            const std::string & cls,
            const std::vector< std::string > & wrds )
```

Gets the number of occurences of a set of words in a class

Parameters

cls	The class
wrds	The set of words in a vector of strings

Returns

The count

2.4.3.11 getCount() [5/5]

```
size_t WordMatrix::getCount (
            const std::vector< std::string > & clss,
            const std::vector< std::string > & wrds )
```

Gets the number of occurences of a set of words in a set of classes

Parameters

clss	The set of classes in a vector of strings
wrds	The set of words in a vector of strings

Returns

The count

2.4.3.12 getMostFrequent()

```
WordMatrix WordMatrix::getMostFrequent (
            size_t n )
```

Gets the n/n_classes most frequent words in each class that results in n total words

Parameters

n	The total words per class

Returns

A sub-matrix with only the n most frequent words as described above

2.4.3.13 getTotalWords()

```
size_t WordMatrix::getTotalWords ( )
```

Gets the total words in the data

Returns

The total words

2.4.3.14 getWords()

```
vector< string > WordMatrix::getWords ( )
```

Gets the words in the WordMatrix

Returns

The words in the WordMatrix as a vector of strings

2.4.3.15 getWordTotal() [1/2]

```
size_t WordMatrix::getWordTotal (
            const std::string & word )
```

Gets the total number of occurrences of a word in all classes

Parameters

word	The word to sum for

Returns

The total number of occurrences of the word

2.4.3.16 getWordTotal() [2/2]

```
size_t WordMatrix::getWordTotal (
            const std::vector< std::string > & words )
```

Gets the total number of occurrences of a set of words in all classes

Parameters

words	The words in vector of strings

Returns

The total number of occurrences for the set of words

2.4.3.17 predict()

```
std::string WordMatrix::predict (
            const NewsItem & itm )
```

Predict the class of a NewsItem

Parameters

itm	The NewsItem to inference

Returns

The class' string

2.4.3.18 printFrequency() [1/2]

```
void WordMatrix::printFrequency (
            std::ostream & ostr )
```

Prints the frequencies of the words in an output stream

Parameters

ostr	The output stream

2.4.3.19 printFrequency() [2/2]

```
void WordMatrix::printFrequency ( )
```

Prints the frequencies of the words to the terminal

2.4.3.20 printLatexFrequency() [1/2]

```
void WordMatrix::printLatexFrequency (
            std::ostream & ostr )
```

Prints the frequencies of the words in an output stream in latex format

Parameters

ostr	The output stream

2.4.3.21 printLatexFrequency() [2/2]

```
void WordMatrix::printLatexFrequency ( )
```

Prints the frequencies of the words to the terminal in latex format

2.4.3.22 printLatexProbabilities() [1/2]

```
void WordMatrix::printLatexProbabilities (
            std::ostream & ostr )
```

Prints the probability of the words in an output stream in latex format

Parameters

ostr	The output stream

2.4.3.23 printLatexProbabilities() [2/2]

```
void WordMatrix::printLatexProbabilities ( )
```

Prints the probabilities of the words to the terminal in latex format

2.4.3.24 printProbabilities() [1/2]

```
void WordMatrix::printProbabilities (
            std::ostream & ostr )
```

Prints the probabilities of the words in an output stream

Parameters

ostr	The output stream

2.4.3.25 printProbabilities() [2/2]

```
void WordMatrix::printProbabilities ( )
```

Prints the probabilities of the words to the terminal

2.4.3.26 prune_classes()

```
WordMatrix WordMatrix::prune_classes (
            size_t n )
```

Prunes the number of classes randomly to n

Parameters

n	The number of classes in the resulting WordMatrix

Returns

A WordMatrix with n classes

2.4.3.27 word_index()

```
size_t WordMatrix::word_index (
            const std::string & word )
```

Gets the index, in the WordMatrix, of a word

Parameters

word	The word

Returns

The WordMatrix's index

The documentation for this class was generated from the following files:

- /home/aherrera/CLionProjects/naivebayescpp/src/WordMatrix.h
- /home/aherrera/CLionProjects/naivebayescpp/src/WordMatrix.cpp

Index