cnf

0.1

Generated by Doxygen 1.7.1

Sat Jul 24 2010 22:12:36

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 AllocMemdiscard Class Reference

`#include <allocmd.h>`

Public Member Functions

- AllocMemdiscard (size_t)
- void * **alloc** (size_t)
- void **reset** ()

2.1.1 Detailed Description

written by kei.uchiumi implementated Memory Discard Strategy Copyright (C) 2010- Kei Uchiumi

2.1.2 Constructor & Destructor Documentation

2.1.2.1 AllocMemdiscard::AllocMemdiscard (size_t *heapsize*)

Copyright (C) 2010- Kei Uchiumi

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

- lib/allocmd.h
- lib/allocmd.cpp

2.2 Learner< T > Class Template Reference

Public Member Functions

- Learner (const char *tmpl, const char *corpus, unsigned int poolsize)
- void learn (unsigned int iter, unsigned int reg)
- void save (const char *save)

- bool init ()
- void setcache (unsigned int cachesize)
- void setpenalty (float w, float u, float t)
- void setpenalty (float bs, float us, float bf, float uf, float t)
- void setlabelcol (unsigned int labelcol)
- void setsqcol (unsigned int sqcolsize)
- void setbound (unsigned int bound)
- void setfbound (unsigned int fbound)
- void setsbound (unsigned int sbound)
- void setlambda (float lambda)
- void setalpha (float alpha)
- template<>
  void **setpenalty** (float w, float u, float t)
- template<>
  void **setpenalty** (float bs, float us, float bf, float uf, float t)
- template<>
  void **setbound** (unsigned int bound)
- template<>
  void **setfbound** (unsigned int fbound)
- template<>
  void **setsbound** (unsigned int sbound)
- template<>
  void **setalpha** (float alpha)

template<class T> class Learner< T >

2.2.1 Constructor & Destructor Documentation

2.2.1.1 template<class T > Learner< T >::Learner (const char * *tmpl*, const char * *corpus*, unsigned int *poolsize*)

Construct Learner

Parameters

tmpl template

corpus training-corpus

poolsize poolsize for allocator

2.2.2 Member Function Documentation

2.2.2.1 template<class T > bool Learner< T >::init ( )

Initialize

2.2.2.2 template<class T > void Learner< T >::learn (unsigned int *iter*, unsigned int *reg*)

Learn Model Parameters

Parameters

iter iteration

regularize 0:L1-regularization 1:L2-regularization

2.2.2.3 template<class T > void Learner< T >::save (const char * *save*)

Save Model Parameters

Parameters

save modelfile

2.2.2.4 template<class T > void Learner< T >::setalpha (float *alpha*)

Set Alpha parameter for detecting learning-rate eta

Parameters

alpha alpha

2.2.2.5 template<class T > void Learner< T >::setbound (unsigned int *bound*)

Set threshold of cut-off for feature

Parameters

bound NUM

2.2.2.6 template<class T > void Learner< T >::setcache (unsigned int *cachesize*)

Set cachesize

Parameters

cache cachesize

2.2.2.7 template<class T > void Learner< T >::setfbound (unsigned int *fbound*)

Set threshold of cut-off for token feature

Parameters

bound NUM

2.2.2.8 template<class T > void Learner< T >::setlabelcol (unsigned int *labelcol*)

Set label-col in sequence

Parameters

labelcol labelcol

2.2.2.9 template<class T > void Learner< T >::setlambda (float *lambda*)

Set Lambda parameter for detecting learning-rate eta

Parameters

lambda lambda

2.2.2.10 template<class T > void Learner< T >::setpenalty (float *bs,* float *us,* float *bf,* float *uf,* float *t*)

Set penalty parameters

Parameters

bs penalty parameter for transition-feature of segment

us penalty parameter for observed-feature of segment

bf penalty parameter for transition-feauture of token

uf penalty parameter for observed-feauture of token

t penalty parameter for gate-function

2.2.2.11 template<class T > void Learner< T >::setpenalty (float *w,* float *u,* float *t*)

Set penalty parameters

Parameters

w penalty parameter for observed-feature

u penalty parameter for transition-feauture

t penalty parameter for gate-function

2.2.2.12 template<class T > void Learner< T >::setsbound (unsigned int *sbound*)

Set threshold of cut-off for segment feature

Parameters

bound NUM

2.2.2.13 template<class T > void Learner< T >::setsqcol (unsigned int *sqcolsize*)

Set colsize of sequence

Parameters

colsize colsize of sequence

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

- src/learner.hpp

2.3 MyUtil Class Reference

`#include <myutil.h>`

Static Public Member Functions

- static bool **IsEOS** (const char *str)
- static bool **IsEOS** (std::string &s)
- static void **chomp** (char *str)
- static void **chomp** (std::string &s)
- static unsigned int **getByteUtf8** (const char *p)
- static void **push** (int c, char **s)
- static void **itoalter** (int n, char **s)
- static void **itoa** (int n, char *s)
- static bool **IsCommentOut** (const char *str)
- static void sqread (FILE *fp, Sequence *s, unsigned int bufsize)
- static void sqread (std::ifstream &in, Sequence *s)

2.3.1 Detailed Description

written by kuchiumi util class for folos crf

2.3.2 Member Function Documentation

2.3.2.1 static void MyUtil::sqread (std::ifstream & *in*, Sequence * *s*) [inline, static]

read sequence function

Parameters

in ifstream

sq Sequence

2.3.2.2 static void MyUtil::sqread (FILE * *fp*, Sequence * *s*, unsigned int *bufsize*) [inline, static]

read sequence function

Parameters

fp FILE pointer

sq Sequence

bufsize buffer size

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

- lib/myutil.h

2.4 Sequence Class Reference

Public Member Functions

- int **dump** ()
- int push (const char *str)
- int **clear** ()
- unsigned int getRowSize ()
- char * getToken (int row, int col)
- int setDelimit (const char *delimit)
- int **init** ()
- int setAllocSize (unsigned int allocsize)
- int setColSize (unsigned int colsize)

2.4.1 Member Function Documentation

2.4.1.1 unsigned int Sequence::getRowSize ( )

Parameters

[out] ***rowsize***

2.4.1.2 char * Sequence::getToken (int *row*, int *col*)

Parameters

[in] ***row***

[in] ***col***

[out] ***char****

2.4.1.3 int Sequence::push (const char * *str*)

Parameters

[in] ***str***

2.4.1.4 int Sequence::setAllocSize (unsigned int *allocsize*)

Parameters

[in] ***allocsize***

2.4.1.5 int Sequence::setColSize (unsigned int *colsize*)

Parameters

[in] ***colsize***

2.4.1.6 int Sequence::setDelimit (const char * *delimit*)

Parameters

[in] ***delimit***

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

- lib/sequence.h
- lib/sequence.cpp

2.5 Tagger< T > Class Template Reference

Public Member Functions

- Tagger (const char *tmpl, unsigned int poolsize)
- void read (const char *model)
- void tagging (const char *corpus)
- void viterbi (Sequence *s, AllocMemdiscard *cache, std::vector< int > &labels)
- void output (Sequence *s, std::vector< int > &labels)
- void setcache (unsigned int cachesize)
- void setsqcol (unsigned int sqcolsize)
- void clear ()
- template<>
  void **clear** ()

template<class T> class Tagger< T >

2.5.1 Constructor & Destructor Documentation

2.5.1.1 template<class T > Tagger< T >::Tagger (const char * *tmpl*, unsigned int *poolsize*)

Construct Tagger

Parameters

tmpl template

poolsize poolsize for allocator

2.5.2 Member Function Documentation

2.5.2.1 template<class T > void Tagger< T >::clear ( )

Clear model parameter

2.5.2.2 template<class T > void Tagger< T >::output (Sequence * *s*, std::vector< int > & *labels*)

Output labeled-sequence

Parameters

s sequence

labels vector stored labels

2.5.2.3 template<class T > void Tagger< T >::read (const char * *model*)

Read Modelfile

Parameters

model modelfile

2.5.2.4 template<class T > void Tagger< T >::setcache (unsigned int *cachesize*)

Set cachesize

Parameters

cache cachesize

2.5.2.5 template<class T > void Tagger< T >::setsqcol (unsigned int *sqcolsize*)

Set colsize of sequence

Parameters

colsize colsize of sequence

2.5.2.6 template<class T > void Tagger< T >::tagging (const char * *corpus*)

Tagging to corpus

Parameters

corpus test-corpus

2.5.2.7 template<class T > void Tagger< T >::viterbi (Sequence * *s*, AllocMemdiscard * *cache*, std::vector< int > & *labels*)

Predict Tags and set to labels

Parameters

s target sequence

cache cache

labels vector to store predicted labels

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

- src/tagger.hpp

Index