Package 'fastTextR'

May 12, 2017

Type Package

Title An Interface to the 'fastText' Library

Version 1.0

Date 2016-09-22

Author Florian Schwendinger [aut, cre]

Maintainer Florian Schwendinger <FlorianSchwendinger@gmx.at>

Description An interface to the 'fastText' library
<https://github.com/facebookresearch/fastText>. The package
can be used for text classification and to learn word vectors.
The install folder contains the 'PATENTS' file.
An example how to use 'fastTextR' can be found in the 'README' file.

License BSD_3_clause + file LICENSE

Imports stats, graphics, Rcpp (>= 0.12.4)

LinkingTo Rcpp

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-05-12 06:11:25 UTC

R topics documented:

fasttext *Train a Model*

Description

Train a new word representation model or supervised classification model.

Usage

```
fasttext(input, method = c("supervised", "cbow", "skipgram"),
  control = ft.control())
```

Arguments

input	a character string giving the location of the input file.
method	a character string giving the method, possible values are 'supervised', 'cbow' and 'skipgram'.
control	a list giving the control variables, for more information see ft.control.

Examples

```
## Not run:
model <- fasttext("my_data.txt", method="supervised",
                  control = ft.control(nthreads = 1L))

## End(Not run)
```

ft.control *Default Control Settings*

Description

A auxiliary function for defining the control variables.

Usage

```
ft.control(loss = c("softmax", "hs", "ns"), learning_rate = 0.05,
  learn_update = 100L, word_vec_size = 5L, window_size = 5L, epoch = 5L,
  min_count = 5L, min_count_label = 0L, neg = 5L, max_len_ngram = 1L,
  nbuckets = 2000000L, min_ngram = 3L, max_ngram = 6L, nthreads = 1L,
  threshold = 1e-04, label = "__label__", verbose = 0,
  pretrained_vectors = "")
```

Arguments

loss	a character string giving the name of the loss function allowed values are 'softmax', 'hs' and 'ns'.
learning_rate	a numeric giving the learning rate, the default value is 0.05.
learn_update	an integer giving after how many tokens the learning rate should be updated. The default value is 100L, which means the learning rate is updated every 100 tokens.
word_vec_size	an integer giving the length (size) of the word vectors.
window_size	an integer giving the size of the context window.
epoch	an integer giving the number of epochs.
min_count	an integer giving the minimal number of word occurences.
min_count_label	and integer giving the minimal number of label occurences.
neg	an integer giving how many negatives are sampled (only used if loss is "ns").
max_len_ngram	an integer giving the maximum length of ngrams used.
nbuckets	an integer giving the number of buckets.
min_ngram	an integer giving the minimal ngram length.
max_ngram	an integer giving the maximal ngram length.
nthreads	an integer giving the number of threads.
threshold	a numeric giving the sampling threshold.
label	a character string specifying the label prefix (default is '__label__').
verbose	an integer giving the verbosity level, the default value is 0L and shouldn't be changed since Rcpp::Rcout cann't handle the traffic.
pretrained_vectors	a character string giving the file path to the pretrained word vectors which are used for the supervised learning.

Value

a list with the control variables.

Examples

```
ft.control(learning_rate=0.1)
```

get_words *Get Words*

Description

Obtain all the words from a previously trained model.

Usage

```
get_words(model)
```

Arguments

model	an object inheriting from "fasttext".

Value

a character vector.

Examples

```
## Not run:
get_words(model)

## End(Not run)
```

get_word_vectors *Get Word Vectors*

Description

Obtain word vectors from a previously trained model.

Usage

```
get_word_vectors(model, words)
```

Arguments

model	an object inheriting from "fasttext".
words	a character vector giving the words.

Value

a matrix containing the word vectors.

Examples

```
## Not run:
get_word_vectors(model, c("word", "vector"))

## End(Not run)
```

`normalize` *Normalize*

Description

Applies normalization to a given text.

Usage

```
normalize(txt)
```

Arguments

`txt`	a character vector to be normalized.

Value

a character vector.

Examples

```
## Not run:
normalize(some_text)

## End(Not run)
```

`predict.supervised_model`
Predict using a Previously Trained Model

Description

Predict values based on a previously trained model.

Usage

```
## S3 method for class 'supervised_model'
predict(object, newdata = character(),
  newdata_file = "", result_file = "", k = 1L, prob = FALSE, ...)
```

Arguments

object	an object inheriting from 'fasttext'.
newdata	a character vector giving the new data.
newdata_file	a character string giving the location of to the new data.
result_file	a character string naming a file.
k	an integer giving the number of labels to be returned.
prob	a logical if true the probabilities are also returned.
...	currently not used.

Value

NULL if a 'result_file' is given otherwise if 'prob' is true a data.frame with the predicted labels and the corresponding probabilities, if 'prob' is false a character vector with the predicted labels.

Examples

```
## Not run:
predict(object, newdata)

## End(Not run)
```

---

read.fasttext *Read Model*

Description

Read a previously saved model from file.

Usage

```
read.fasttext(file)
```

Arguments

file	a character string giving the name of the file to be read in.

Value

an object inheriting from "fasttext".

Examples

```
## Not run:
model <- read.fasttext( "dbpedia.bin" )

## End(Not run)
```

`save.fasttext` *Save Model*

Description

Save the model to a file.

Usage

```
save.fasttext(model, file)
```

Arguments

`model`	an object inheriting from `"fasttext"`.
`file`	a character string giving the name of the file.

Examples

```
## Not run:
save.fasttext(model = m, file = "data.model")

## End(Not run)
```

Index