Package 'gmining'

January 19, 2017

Type Package

Title Gutenberg Mining

Version 0.1.0

Author Andrea Duque

Maintainer Andrea Duque <abd@cin.ufpe.br>

Description Contains utility functions for text analysis in project Gutenberg books.

License MIT <http://opensource.org/licenses/MIT>

Depends igraph, stringr, tm, NLP, tokenizers, rowr, dplyr

Suggests SnowballC, wordcloud

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

R topics documented:

corpus_to_graphs — corpus_to_graphs *returns a list of co-ocurrence book graphs from a book Corpus. It's not optimized, so it might take a while for a large Corpus.*

Description

corpus_to_graphs returns a list of co-ocurrence book graphs from a book Corpus. It's not optimized, so it might take a while for a large Corpus.

Usage

```
corpus_to_graphs(myCorpus)
```

Arguments

myCorpus	a tm VCorpus

Value

A list of igraph graphs

Examples

```
myCorpus <- Corpus(VectorSource("three men wait door say holmes oh indeed
                                seem do thing completely must compliment holmes answer"))
book_graphs <- corpus_to_graphs(myCorpus)
```

create_edge — create_edge *returns an edge for a pair of words in a book graph.*

Description

create_edge returns an edge for a pair of words in a book graph.

Usage

```
create_edge(pair, node_labels)
```

Arguments

pair	A character vector of length 2.
node_labels	A character vector of generic length, containing all unique tokens extracted from a text.

Value

A list of integer vectors of length 2 each, containing the positions of the pair in the token vector.

Examples

```
create_edge(c("three", "men"), c("three", "men", "are", "waiting"))
create_edge(c("three", "are"), c("three", "men", "are", "waiting"))
```

---

create_edge_list *create_edge_list returns all edges of a book graph.*

Description

create_edge_list returns all edges of a book graph.

Usage

```
create_edge_list(corpus_element)
```

Arguments

book_content	A character vector containing all words from a text book.

Value

A list of edge pairs (integer vector of length 2) of the co-ocurrence graph book text.

Examples

```
create_edge_list("three men wait door say holmes oh indeed seem do thing completely must compliment holmes answer
```

---

create_graph *create_graph creates a graph from an edge list list*

Description

create_graph creates a graph from an edge list list

Usage

```
create_graph(edge_list)
```

Arguments

edge_list	A list of integer vectors of length 2

Value

An igraph graph.

Examples

```
create_graph(list(c(1,2), c(2,3), c(3,1)))
```

`create_nodes` `create_nodes` *returns all nodes for a book graph.*

Description

`create_nodes` returns all nodes for a book graph.

Usage

```
create_nodes(book_content)
```

Arguments

`book_content`	A character vector containing all words from a text book.

Value

A character vector containing unique tokens from book.

Examples

```
create_nodes("three men wait door say holmes oh indeed seem do thing completely must compliment holmes answer")
```

`get_book_content` `get_book_content` *returns the content from a book from a VCorpus*

Description

`get_book_content` returns the content from a book from a VCorpus

Usage

```
get_book_content(corpus_element)
```

Arguments

`corpus_element`	An element from a Vcorpus

Value

A character vector containing the book content

Examples

```
myCorpus <- Corpus(VectorSource("three men wait door say holmes oh indeed seem do thing completely must complimen
#get content from first book in VCorpus
get_book_content(myCorpus[[1]])
```

---

get_nodes_labels *returns a list of nodes labels for each book in a Corpus*

Description

get_nodes_labels returns a list of nodes labels for each book in a Corpus

Usage

```
get_nodes_labels(myCorpus)
```

Arguments

myCorpus	a tm VCorpus

Value

A list of tokens

Examples

```
myCorpus <- Corpus(VectorSource("three men wait door say holmes oh indeed
                                seem do thing completely must compliment holmes answer"))
nodes_labels <- get_nodes_labels(myCorpus)
```

---

gmining *gmining: A package for making data analysis in Project Gutenberg easy.*

Description

The foo package provides three categories of important functions: foo, bar and baz.

gmining functions

The gmining functions ...

Index