Package 'R.temis'

June 17, 2019

Type Package

Title Integrated Text Mining Solution

Version 0.1.2

Imports stats, utils, graphics, testthat, wordcloud, igraph, stringi,
crayon, SnowballC, tm.plugin.factiva, tm.plugin.lexisnexis,
tm.plugin.europresse, tm.plugin.alceste

Depends tm (>= 0.6), NLP, slam, FactoMineR, explor

Description An integrated solution to perform
a series of text mining tasks such as importing and cleaning a corpus, and
analyses like terms and documents counts, lexical summary, terms
co-occurrences and documents similarity measures, graphs of terms,
correspondence analysis and hierarchical clustering. Corpora can be imported
from spreadsheet-like files, directories of raw text files,
as well as from 'Dow Jones Factiva', 'LexisNexis', 'Europresse' and 'Alceste' files.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

URL https://github.com/nalimilan/R.TeMiS

BugReports https://github.com/nalimilan/R.TeMiS/issues

NeedsCompilation no

Author Milan Bouchet-Valat [aut, cre],
Gilles Bastin [aut],
Antoine Chollet [aut]

Maintainer Milan Bouchet-Valat <nalimilan@club.fr>

Repository CRAN

Date/Publication 2019-06-17 15:10:03 UTC

R topics documented:

`add_clusters` *add_clusters*

Description

Add a meta-data variable to a corpus indicating the cluster to which each document belongs.

Usage

```
add_clusters(corpus, clust)
```

Arguments

`corpus`	A `Corpus` object.
`clust`	A `HCPC` object resulting from a call to `corpus_clustering`.

Value

A `Corpus` object with `meta(corpus, "cluster")` indicating the cluster of each document.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
res <- corpus_ca(corpus, dtm, ncp=2, sparsity=0.98)
clust <- corpus_clustering(res, 3)
corpus <- add_clusters(corpus, clust)
meta(corpus)
```

---

build_dtm *build_dtm*

Description

Compute document-term matrix from a corpus.

Usage

```
build_dtm(corpus, sparsity = 1, dictionary = NULL,
  remove_stopwords = FALSE, tolower = TRUE,
  remove_punctuation = TRUE, remove_numbers = TRUE, min_length = 2)
```

Arguments

`corpus` A `Corpus` object.

`sparsity` Value between 0 and 1 indicating the proportion of documents with no occurrences of a term above which that term should be dropped. By default all terms are kept (`sparsity=1`).

`dictionary` A vector of terms to which the matrix should be restricted. By default, all words with more than `min_length` characters are considered.

`remove_stopwords` Whether to remove stopwords appearing in a language-specific list (see `tm::stopwords`).

`tolower` Whether to convert all text to lower case.

`remove_punctuation` Whether to remove all punctuation from text before tokenizing terms.

`remove_numbers` Whether to remove all numbers from text before tokenizing terms.

`min_length` The minimal number of characters for a word to be retained.

Value

A `DocumentTermMatrix` object.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
build_dtm(corpus)
```

`characteristic_docs` *characteristic_docs*

Description

Print documents which are the most characteristic of each level of a variable, i.e. those with the lowest Chi-squared distance to the average vocabulary of documents belonging to that level.

Usage

```
characteristic_docs(corpus, dtm, variable, ndocs = 10, nterms = 25,
  p = 0.1)
```

Arguments

`corpus`	A `Corpus` object.
`dtm`	A `DocumentTermMatrix` object corresponding to `corpus`.
`variable`	A vector of values giving the groups for which most frequent terms should be reported.
`ndocs`	The number of (most characteristic) documents to print.
`nterms`	The number of terms to highlight in documents.
`p`	The maximum p-value up to which specific terms should be hightlighted.

Details

Occurrences of the `nterms` most specific terms for each level are highlighted. If stemming or other transformations have been applied to original words using `combine_terms`, all original words which have been transformed to the specified terms are highlighted.

Value

A list with one `Corpus` object for each level (invisibly).

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
characteristic_docs(corpus, dtm, meta(corpus)$Date)

# Also works when terms have been combined
dict <- dictionary(dtm)
dtm2 <- combine_terms(dtm, dict)
characteristic_docs(corpus, dtm2, meta(corpus)$Date)
```

---

combine_terms *combine_terms*

Description

Aggregate terms in a document-term matrix to according to groupings specified by a dictionary.

Usage

```
combine_terms(dtm, dict)
```

Arguments

dtm	A DocumentTermMatrix object.
dict	A data.frame with one row per term in dtm that should be retained. The row names must match names of rows in dtm, and the first column must give the term into which it should be transformed.

Details

If several terms use the same transformation, they will be aggregated together. Terms missing from dict will be dropped.

Value

An aggregated DocumentTermMatrix object.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
dict <- dictionary(dtm)
combine_terms(dtm, dict)
```

concordances *concordances*

Description

Print documents which contain one or more terms and return a sub-corpus with these documents.

Usage

```
concordances(corpus, dtm, terms, all = FALSE)
```

Arguments

corpus	A Corpus object.
dtm	A DocumentTermMatrix object corresponding to corpus.
terms	One of more terms appearing in dtm.
all	Whether only documents containing all terms should be printed. By default, documents need to contain at least one of the terms.

Details

Occurrences of the specified terms are highlighted. If stemming or other transformations have been applied to original words using combine_terms, all original words which have been transformed to the specified terms are highlighted.

Value

Corpus object (invisibly).

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
concordances(corpus, dtm, "oil")
concordances(corpus, dtm, c("oil", "opec"))
concordances(corpus, dtm, c("oil", "opec"), all=TRUE)

# Also works when terms have been combined
dict <- dictionary(dtm)
dtm2 <- combine_terms(dtm, dict)
concordances(corpus, dtm2, "product")
```

contributive_docs *contributive_docs*

Description

Print documents which contribute the most to an axis of correspondence analysis.

Usage

```
contributive_docs(corpus, ca, axis, ndocs = 10, nterms = 25)
```

Arguments

corpus	A Corpus object.
ca	A CA object.
axis	The CA axis to consider.
ndocs	The number of (most contributive) documents to print.
nterms	The number of terms to highlight in documents.

Details

Occurrences of the nterms most contributive terms are highlighted. If stemming or other transformations have been applied to original words using combine_terms, all original words which have been transformed to the specified terms are highlighted.

Value

Corpus object (invisibly).

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
ca <- corpus_ca(corpus, dtm)
contributive_docs(corpus, ca, 1)

# Also works when terms have been combined
dict <- dictionary(dtm)
dtm2 <- combine_terms(dtm, dict)
ca2 <- corpus_ca(corpus, dtm2)
contributive_docs(corpus, ca2, 1)
```

`cooc_terms` *cooc_terms*

Description

Show terms that are the most associated (positively or negatively) with a reference term.

Usage

```
cooc_terms(dtm, term, variable = NULL, p = 0.1, n = 25,
  sparsity = 1, min_occ = 2)
```

Arguments

`dtm` A `DocumentTermMatrix`.

`term` A reference term appearing in `dtm`.

`variable` An optional vector of values giving the groups for which most frequent terms should be reported.

`p` The maximum p-value up to which terms should be reported.

`n` The maximal number of terms to report (for each group, if applicable).

`sparsity` Value between 0 and 1 indicating the proportion of documents with no occurrences of a term above which that term should be dropped. By default all terms are kept (`sparsity=1`).

`min_occ` The minimum number of occurrences in the whole `dtm` below which terms should be skipped.

Details

Co-occurrent terms are those which are specific to documents which contain the given term. The output is the same as that returned by `specific_terms`.

Value

A list of matrices, one for each level of the variable, with columns:

- "% Term/Level": the percent of the term's occurrences in all terms occurrences in documents where the chosen term is also present.
- "% Level/Term": the percent of the term's occurrences that appear in documents where the chosen term is also present (rather than in documents where it does not appear), i.e. the percent of cooccurrences for the term..
- "Global %": the percent of the term's occurrences in all terms occurrences in the corpus (or in the subset of the corpus corresponding to the variable level).
- "Level": the number of cooccurrences of the term.
- "Global": the number of occurrences of the term in the corpus (or in the subset of the corpus corresponding to the variable level).

- "t value": the quantile of a normal distribution corresponding the probability "Prob.".
- "Prob.": the probability of observing such an extreme (high or low) number of occurrences of the term in documents where the chosen term is also present, under an hypergeometric distribution.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
cooc_terms(dtm, "barrel")
cooc_terms(dtm, "barrel", meta(corpus)$Date)
```

corpus_ca *corpus_ca*

Description

Run a correspondence analysis on a corpus.

Usage

```
corpus_ca(corpus, dtm, variables = NULL, ncp = 5, sparsity = 1, ...)
```

Arguments

corpus	A Corpus object.
dtm	A DocumentTermMatrix object corresponding to corpus with one row per document.
variables	An optional list of variables in meta(corpus) over which to aggregate dtm. If NULL (the default), the analysis is run on the unaggregated matrix.
ncp	The number of axes to compute (5 by default). Note that this determines the number of axes that will be used for clustering by HCPC. Pass Inf to compute all axes.
sparsity	Value between 0 and 1 indicating the proportion of documents with no occurrences of a term above which that term should be dropped. By default all terms are kept (sparsity=1).
...	Additional arguments passed to FactoMineR::CA.

Value

A CA object containing the correspondence analysis results.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
corpus_ca(corpus, dtm, ncp=3, sparsity=0.98)
```

---

corpus_clustering *corpus_clustering*

Description

Run a hierarchical clustering on documents of a corpus based on a correspondence analysis. The number of axes from ca which are used depends on the value of the n argument passed to corpus_ca.

Usage

```
corpus_clustering(ca, n = 0)
```

Arguments

ca	A CA object resulting from a call to corpus_ca.
n	Number of clusters to create. If 0 (the default), it is determined by clicking on the plot to choose the cut height.

Value

A HCPC object.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
res <- corpus_ca(corpus, dtm, ncp=2, sparsity=0.98)
corpus_clustering(res, 3)
```

dictionary *dictionary*

Description

Create a dictionary with information on all words in a corpus.

Usage

```
dictionary(dtm, remove_stopwords = FALSE)
```

Arguments

dtm	A DocumentTermMatrix object.
remove_stopwords	Whether stopwords should be removed from the dictionary.

Value

A data.frame with row names indicating the terms, and columns giving the stem, the number of occurrences, and whether the term is a stopword.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
dictionary(dtm)
```

extreme_docs *extreme_docs*

Description

Print documents which have the most extreme coordinations on an axis of correspondence analysis.

Usage

```
extreme_docs(corpus, ca, axis, ndocs = 10, nterms = 25)
```

Arguments

corpus	A Corpus object.
ca	A CA object.
axis	The CA axis to consider.
ndocs	The number of (most contributive) documents to print.
nterms	The number of terms to highlight in documents.

Details

Occurrences of the nterms most extreme terms are highlighted. If stemming or other transformations have been applied to original words using combine_terms, all original words which have been transformed to the specified terms are highlighted.

Value

Corpus object (invisibly).

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
ca <- corpus_ca(corpus, dtm)
contributive_docs(corpus, ca, 1)

# Also works when terms have been combined
dict <- dictionary(dtm)
dtm2 <- combine_terms(dtm, dict)
ca2 <- corpus_ca(corpus, dtm2)
extreme_docs(corpus, ca2, 1)
```

frequent_terms *frequent_terms*

Description

List terms with the highest number of occurrences in the document-term matrix of a corpus, possibly grouped by the levels of a variable.

Usage

```
frequent_terms(dtm, variable = NULL, n = 25)
```

Arguments

`dtm` A `DocumentTermMatrix`.

`variable` An optional vector of values giving the groups for which most frequent terms should be reported.

`n` The maximal number of terms to report (for each group, if applicable).

Value

A list of matrices, one for each level of the variable, with columns:

- "% Term/Level": the percent of the term's occurrences in all terms occurrences in the level.
- "% Level/Term": the percent of the term's occurrences that appear in the level (rather than in other levels).
- "Global %": the percent of the term's occurrences in all terms occurrences in the corpus.
- "Level": the number of occurrences of the term in the level ("internal").
- "Global": the number of occurrences of the term in the corpus.
- "t value": the quantile of a normal distribution corresponding the probability "Prob.".
- "Prob.": the probability of observing such an extreme (high or low) number of occurrences of the term in the level, under an hypergeometric distribution.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
frequent_terms(dtm)
frequent_terms(dtm, meta(corpus)$Date)
```

`import_corpus` *import_corpus*

Description

Import a corpus from a file.

Usage

```
import_corpus(paths, format, language, textcolumn = 1, encoding = NULL)
```

Arguments

`paths` Path to one of more files, or to a directory (if `format="txt"`) to import.

`format` File format: can be `"csv"`, `"txt"`, `"factiva"`, `"europresse"`, `"lexisnexis"` or `"alceste"`.

`language` The language name or code (preferably as IETF language tags, see `language`) to be used in particular for stopwords and stemming.

`textcolumn` When `format="csv"`, the column containing the text, either as a string or as a position

`encoding` The character encoding of the file, or `NULL` to attempt automatic detection.

Value

A `Corpus` object.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
import_corpus(file, "factiva", language="en")
```

`lexical_summary` *lexical_summary*

Description

Build a lexical summary table, optionally over a variable.

Usage

```
lexical_summary(dtm, corpus, variable = NULL, unit = c("document",
  "global"))
```

Arguments

`dtm` A `DocumentTermMatrix` containing the terms to summarize, which may have been stemmed.

`corpus` A `Corpus` object containing the original texts from which `dtm` was constructed.

`variable` An optional vector with one element per document indicating to which category it belongs. If ‘NULL, per-document measures are returned.

`unit` When `variable` is not `NULL`, defines the way measures are aggregated (see below).

Details

Words are defined as the forms of two or more characters present in the texts before stemming and stopword removal. On the contrary, unique *terms* are extracted from dtm, which means they do not include words that were removed from it, and that words different in the original text might become identical terms if stemming was performed. Please note that percentages for terms and words are computed with regard respectively to the total number of terms and of words, so the denominators are not the same for all measures.

When variable is not NULL, unit defines two different ways of aggregating per-document statistics into per-category measures:

- "document": values computed for each document are simply averaged for each category.
- "global": values are computed for each category taken as a whole: word counts are summed for each category, and ratios and averages are calculated for this level only, from the summed counts.

This distinction does not make sense when variable=NULL: in this case, "level" in the above explanation corresponds to "document", and two columns are provided about the whole corpus.

- "Corpus mean" is simply the average value of measures over all documents
- "Corpus total" is the sum of the number of terms, the percentage of terms (ratio of the summed numbers of terms) and the average word length in the corpus when taken as a single document.

Value

A table object with the following information for each document or each category of documents in the corpus:

- total number of terms
- number and percent of unique terms (i.e. appearing at least once) number and percent of hapax legomena (i.e. terms appearing once and only once)
- total number of words
- number and percent of long words (defined as at least seven characters)
- number and percent of very long words (defined as at least ten characters)
- average word length

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
lexical_summary(dtm, corpus)
```

set_corpus_variables *set_corpus_variables*

Description

Set corpus meta-data variables from a data frame.

Usage

```
set_corpus_variables(corpus, dset)
```

Arguments

corpus	A Corpus object.
dset	A data.frame containing meta-data variables, with one row per document in ‘corpus.

Value

A Corpus object with meta-data added.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
dset <- data.frame(x=1:length(corpus))
corpus <- set_corpus_variables(corpus, dset)
```

specific_terms *specific_terms*

Description

List terms most associated (positively or negatively) with each document or each of a variable’s levels.

Usage

```
specific_terms(dtm, variable = NULL, p = 0.1, n = 25, sparsity = 1,
  min_occ = 2)
```

Arguments

`dtm` A `DocumentTermMatrix`.

`variable` An optional vector of values giving the groups for which most frequent terms should be reported.

`p` The maximum p-value up to which terms should be reported.

`n` The maximal number of terms to report (for each group, if applicable).

`sparsity` Value between 0 and 1 indicating the proportion of documents with no occurrences of a term above which that term should be dropped. By default all terms are kept (`sparsity=1`).

`min_occ` The minimum number of occurrences in the whole `dtm` below which terms should be skipped.

Details

Specific terms reported here are those whose observed frequency in the document or level has the lowest probability under an hypergeometric distribution, based on their global frequencies in the corpus and on the number of occurrences of all terms in the document or variable level considered. The positive or negative character of the association is visible from the sign of the t value, or by comparing the value of the "% Term/Level" column with that of the "Global %" column.

Value

A list of matrices, one for each level of the variable, with columns:

- "% Term/Level": the percent of the term's occurrences in all terms occurrences in the level.
- "% Level/Term": the percent of the term's occurrences that appear in the level (rather than in other levels).
- "Global %": the percent of the term's occurrences in all terms occurrences in the corpus.
- "Level": the number of occurrences of the term in the level ("internal").
- "Global": the number of occurrences of the term in the corpus.
- "t value": the quantile of a normal distribution corresponding the probability "Prob.".
- "Prob.": the probability of observing such an extreme (high or low) number of occurrences of the term in the level, under an hypergeometric distribution.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
specific_terms(dtm)
specific_terms(dtm, meta(corpus)$Date)
```

split_documents *split_documents*

Description

Split documents in a corpus into documents of one of more paragraphs.

Usage

```
split_documents(corpus, chunksize, preserveMetadata = TRUE)
```

Arguments

`corpus` A `Corpus` object.

`chunksize` The number of paragraphs each new document should contain at most.

`preserveMetadata` Whether to preserve the meta-data of original documents.

Value

A `Corpus` object with split documents.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
split_documents(corpus, 3)
```

subset_corpus *subset_corpus*

Description

Select documents containing (or not containing) one or more terms.

Usage

```
subset_corpus(corpus, dtm, terms, exclude = FALSE, all = FALSE)
```

Arguments

- corpus — A Corpus object.
- dtm — A DocumentTermMatrix object corresponding to corpus.
- terms — One of more terms appearing in dtm.
- exclude — Whether documents containing the terms should be excluded rather than retained.
- all — Whether only documents containing all terms should be retained or excluded. By default, documents need to contain at least one of the terms.

Value

Corpus object.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
subset_corpus(corpus, dtm, "barrel")
subset_corpus(corpus, dtm, c("barrel", "opec"))
subset_corpus(corpus, dtm, c("barrel", "opec"), exclude=TRUE)
subset_corpus(corpus, dtm, c("barrel", "opec"), all=TRUE)
```

terms_graph *terms_graph*

Description

Plot a graph of terms.

Usage

```
terms_graph(dtm, n = 100, min_occ = 0,
  interactive = base::interactive(), vertex.label.cex = 1, ...)
```

Arguments

- dtm — A DocumentTermMatrix object.
- n — The maximum number of terms to represent.
- min_occ — The minimum number of occurrences for a term to be retained.
- interactive — If TRUE, show an interactive plot using tkplot. This is the case by default for interactive sessions.
- vertex.label.cex — The font size for vertex labels. It is interpreted as a multiplication factor of some device-dependent base font size.
- ... — Optional arguments passed to plot.igraph or tkplot.

Value

The ID of the plot returned by tkplot if `interactive=TRUE`, or `NULL` invisibly otherwise.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
terms_graph(dtm, 100, 3)
```

`term_freq` *term_freq*

Description

Study frequencies of chosen terms in the corpus, among documents, or among levels of

Usage

```
term_freq(dtm, terms, variable = NULL, by_term = FALSE)
```

Arguments

`dtm`	A `DocumentTermMatrix`.
`terms`	One or more reference term(s) appearing in `dtm`.
`variable`	An optional vector of values giving the groups for which most frequent terms should be reported.
`by_term`	Whether the third dimension of the array should be terms instead of levels.

Value

A list of matrices, one for each level of the variable, with columns:

- "% Term/Level": the percent of the term's occurrences in all terms occurrences in documents where the chosen term is also present.
- "% Level/Term": the percent of the term's occurrences that appear in documents where the chosen term is also present (rather than in documents where it does not appear), i.e. the percent of cooccurrences for the term..
- "Global %": the percent of the term's occurrences in all terms occurrences in the corpus (or in the subset of the corpus corresponding to the variable level).
- "Level": the number of cooccurrences of the term.
- "Global": the number of occurrences of the term in the corpus (or in the subset of the corpus corresponding to the variable level).

- "t value": the quantile of a normal distribution corresponding the probability "Prob.".
- "Prob.": the probability of observing such an extreme (high or low) number of occurrences of the term in documents where the chosen term is also present, under an hypergeometric distribution.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
term_freq(dtm, "barrel")
term_freq(dtm, "barrel", meta(corpus)$Date)
```

word_cloud *word_cloud*

Description

Plot a word cloud from a document-term matrix.

Usage

```
word_cloud(dtm, n = 50, remove_stopwords = TRUE, ...)
```

Arguments

`dtm`	A `DocumentTermMatrix` object.
`n`	The maximum number of words to plot.
`remove_stopwords`	Whether to remove stopwords appearing in a language-specific list (see `tm::stopwords`).
`...`	Additional arguments passed to `wordcloud`.

Examples

```
file <- system.file("texts", "reut21578-factiva.xml", package="tm.plugin.factiva")
corpus <- import_corpus(file, "factiva", language="en")
dtm <- build_dtm(corpus)
word_cloud(dtm)
```

Index