

# Package ‘discoverableresearch’

October 10, 2020

**Title** Checks Title, Abstract and Keywords to Optimise Discoverability

**Version** 0.0.1

## **Description**

A suite of tools are provided here to support authors in making their research more discoverable.

`check_keywords()` -

this function checks the keywords to assess whether they are already represented in the title and abstract. `check_fields()` -

this function compares terminology used across the title, abstract

and keywords to assess where terminological diversity (i.e. the use of synonyms) could increase the likelihood

of the record being identified in a search. The function looks for terms in the title and abstract that also

exist in other fields and highlights these as needing attention. `suggest_keywords()` -

this function takes a

full text document and produces a list of unigrams, bigrams and trigrams (1-, 2- or 2-word phrases) present in the full text after removing stop words (words with a low utility in natural language processing)

that do not occur in the title or abstract that may be suitable candidates for keywords. `suggest_title()` -

this function takes a full text document and produces a list of the most frequently used uni-

grams, bigrams

and trigrams after removing stop words that do not occur in the abstract or keywords that may be suitable

candidates for title words. `check_title()` -

this function carries out a number of sub tasks: 1) it compares

the length (number of words) of the title with the mean length of titles in major bibliographic databases to

assess whether the title is likely to be too short; 2) it assesses the proportion of stop words in the title

to highlight titles with low utility in search engines that strip out stop words; 3) it compares the title with a given sample of record titles from an .ris import and calculates a similarity score based on phrase

overlap. This highlights the level of uniqueness of the title. This version of the package also contains

functions currently in a non-

CRAN package called 'litsearchr' <<https://github.com/elizagrames/litsearchr>>.

**License** GPL-3

**Imports** dplyr, graphics, magrittr, ngram, readr, stats, stringdist,  
stringi, stopwords, synthesizr, tm

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 3.5.0)

**NeedsCompilation** no

**Author** Neal Haddaway [aut, cre] (<<https://orcid.org/0000-0003-3902-2234>>)

**Maintainer** Neal Haddaway <[nealhaddaway@gmail.com](mailto:nealhaddaway@gmail.com)>

**Repository** CRAN

**Date/Publication** 2020-10-10 10:10:05 UTC

## R topics documented:

check_fields . . . . .	2
check_keywords . . . . .	4
check_title . . . . .	5
check_title_length . . . . .	6
fakerake . . . . .	6
format_keywords . . . . .	7
get_ngrams . . . . .	7
get_stopwords . . . . .	8
get_tokens . . . . .	9
language_code . . . . .	10
possible_langs . . . . .	10
remove_punctuation . . . . .	11
suggest_keywords . . . . .	11
suggest_title . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

check_fields	<i>Check all field suitability</i>
--------------	------------------------------------

---

## Description

Check given fields (title, abstract and keywords) for an article to assess discoverability based on similarities across the fields

**Usage**

```
check_fields(title, abstract, keywords)
```

**Arguments**

title	The article title: a short string
abstract	The article abstract: a string
keywords	The article keywords: a vector of strings

**Value**

A dataframe displaying the presence of the terms across the title, abstract, and keywords

**Examples**

```
title <- "A methodology for systematic mapping in environmental sciences"
abstract <- "Systematic mapping was developed in social sciences in response to a lack of empirical
data when answering questions using systematic review methods, and a need for a method to describe
the literature across a broad subject of interest. Systematic mapping does not attempt to answer
a specific question as do systematic reviews, but instead collates, describes and catalogues
available evidence (e.g. primary, secondary, theoretical, economic) relating to a topic or
question of interest. The included studies can be used to identify evidence for policy-relevant
questions, knowledge gaps (to help direct future primary research) and knowledge clusters (sub-
sets of evidence that may be suitable for secondary research, for example systematic review).
Evidence synthesis in environmental sciences faces similar challenges to those found in social
sciences. Here we describe the translation of systematic mapping methodology from social sciences
for use in environmental sciences. We provide the first process-based methodology for systematic
maps, describing the stages involved: establishing the review team and engaging stakeholders;
setting the scope and question; setting inclusion criteria for studies; scoping stage; protocol
development and publication; searching for evidence; screening evidence; coding; production of a
systematic map database; critical appraisal (optional); describing and visualising the findings;
report production and supporting information. We discuss the similarities and differences in
methodology between systematic review and systematic mapping and provide guidance for those
choosing which type of synthesis is most suitable for their requirements. Furthermore, we discuss
the merits and uses of systematic mapping and make recommendations for improving this evolving
methodology in environmental sciences."
keywords <- c("Systematic mapping",
"Evidence-based environmental management",
"Systematic evidence synthesis",
"Evidence review",
"Knowledge gaps",
"Knowledge clusters")
check <- check_fields(title, abstract, keywords)
check$df
check$tit_terms
check$abs_terms
check$key_terms
check$report;
```

---

check_keywords	<i>Check keyword suitability</i>
----------------	----------------------------------

---

### Description

Check given keywords for an article to assess whether they are already represented in the title and abstract

### Usage

```
check_keywords(title, abstract, keywords)
```

### Arguments

title	The article title: a short string
abstract	The article abstract: a string
keywords	The article keywords: a vector of strings

### Value

A dataframe displaying the presence of the keywords in the title and abstract

### Examples

```
title <- "A methodology for systematic mapping in environmental sciences"
abstract <- "Systematic mapping was developed in social sciences in response to a lack of empirical data when answering questions using systematic review methods, and a need for a method to describe the literature across a broad subject of interest. Systematic mapping does not attempt to answer a specific question as do systematic reviews, but instead collates, describes and catalogues available evidence (e.g. primary, secondary, theoretical, economic) relating to a topic or question of interest. The included studies can be used to identify evidence for policy-relevant questions, knowledge gaps (to help direct future primary research) and knowledge clusters (sub-sets of evidence that may be suitable for secondary research, for example systematic review). Evidence synthesis in environmental sciences faces similar challenges to those found in social sciences. Here we describe the translation of systematic mapping methodology from social sciences for use in environmental sciences. We provide the first process-based methodology for systematic maps, describing the stages involved: establishing the review team and engaging stakeholders; setting the scope and question; setting inclusion criteria for studies; scoping stage; protocol development and publication; searching for evidence; screening evidence; coding; production of a systematic map database; critical appraisal (optional); describing and visualising the findings; report production and supporting information. We discuss the similarities and differences in methodology between systematic review and systematic mapping and provide guidance for those choosing which type of synthesis is most suitable for their requirements. Furthermore, we discuss the merits and uses of systematic mapping and make recommendations for improving this evolving methodology in environmental sciences."
keywords <- c("Systematic mapping",
  "Evidence-based environmental management",
  "Systematic evidence synthesis",
  "Evidence review",
```

```

"Knowledge gaps",
"Knowledge clusters")
check <- check_keywords(title, abstract, keywords)
check;

```

---

check\_title

*Check title with those from a test set*


---

## Description

Check given title for an article to assess how discoverable it is

## Usage

```
check_title(title, testset, threshold = 0.6, matches = FALSE, plot = TRUE)
```

## Arguments

title	The article title: a short string
testset	A provided sample set of representative titles to compare with, entered as a .bib or .ris file (or using the RIS .txt file in data as specified in the example below)
threshold	A threshold between 0 and 1 for the similarity score of titles in the sample set relative to the title provided, above which matching titles will be printed out in 'matches'. Default threshold set to 0.6 (arbitrarily)
matches	Logical argument TRUE or FALSE. If TRUE, the matches with a similarity score above the threshold are printed to a data frame ('matches'). If FALSE, no output is provided.
plot	Logical argument TRUE or FALSE. If TRUE, a histogram of the similarity scores of test set titles compared to the title is plotted.

## Value

A report describing the suitability of the title for research discovery based on a comparison with the test set. If 'matches = TRUE', a list containing a report describing the suitability of the title for research discovery based on a comparison with the test set and a database containing matches with a similarity score above the threshold value.

## Examples

```

title <- "A methodology for systematic mapping in environmental sciences"
testset <- system.file("extdata", "sample_titles.txt", package="discoverableresearch")
check <- check_title(title, testset = testset, threshold = 0.7, matches = TRUE, plot = TRUE)
check$output
check$dat;

```

check\_title\_length      *Check title suitability*

---

**Description**

Check given title for an article to assess how discoverable it is based on its length and proportion of words that are non-stop words

**Usage**

```
check_title_length(title)
```

**Arguments**

title                      The article title: a short string

**Value**

An output describing the suitability of the title for research discovery based on its length and the number of non-stop words

**Examples**

```
title <- "A methodology for systematic mapping in environmental sciences"  
check <- check_title_length(title)  
check;
```

---

fakerake                      *Functions from litsearchr (not yet on CRAN) Quick keyword extraction*

---

**Description**

Extracts potential keywords from text separated by stop words

**Usage**

```
fakerake(text, stopwords, min_n = 2, max_n = 5)
```

**Arguments**

text                      A string object to extract terms from  
stopwords                A character vector of stop words to remove  
min\_n                    Numeric: the minimum length ngram to consider  
max\_n                    Numeric: the maximum length ngram to consider

**Value**

A character vector of potential keywords

---

format_keywords	<i>Format input keywords</i>
-----------------	------------------------------

---

**Description**

Convert string of keywords with separator into a vector

**Usage**

```
format_keywords(keywords, sep = ";")
```

**Arguments**

keywords	The article keywords: a vector of strings
sep	Character that separates keywords in a single string

**Value**

A vector of lowercase keywords

**Examples**

```
keywords <- c("Systematic mapping;  
Evidence-based environmental management;  
Systematic evidence synthesis;  
Evidence review;  
Knowledge gaps;  
Knowledge clusters")  
newkeywords <- format_keywords(keywords, sep = ";")  
newkeywords;
```

---

get_ngrams	<i>Extract n-grams from text</i>
------------	----------------------------------

---

**Description**

This function extracts n-grams from text.

**Usage**

```
get_ngrams(  
  x,  
  n = 2,  
  min_freq = 1,  
  ngram_quantile = NULL,  
  stop_words,
```

```

    rm_punctuation = FALSE,
    preserve_chars = c("-", "_"),
    language = "English"
  )

```

### Arguments

x	A character vector from which to extract n-grams.
n	Numeric: the minimum number of terms in an n-gram.
min_freq	Numeric: the minimum number of times an n-gram must occur to be returned.
ngram_quantile	Numeric: what quantile of ngrams should be retained. Defaults to 0.8; i.e. the 80th percentile of ngram frequencies.
stop_words	A character vector of stopwords to ignore.
rm_punctuation	Logical: should punctuation be removed before selecting ngrams?
preserve_chars	A character vector of punctuation marks to be retained if rm_punctuation is TRUE.
language	A string indicating the language to use for removing stopwords.

### Value

A character vector of n-grams.

### Examples

```
get_ngrams("On the Origin of Species By Means of Natural Selection")
```

---

get_stopwords	<i>Retrieve stop words for a given language</i>
---------------	---

---

### Description

This function retrieves stop words to use for a specified language.

### Usage

```
get_stopwords(language = "English")
```

### Arguments

language	A character vector containing the name of the language for which to retrieve stop words. Defaults to "English"
----------	--

### Value

Returns a character vector of stop words.



**Examples**

```
get_stopwords("English")
```

---

get_tokens	<i>Remove stopwords from text</i>
------------	-----------------------------------

---

**Description**

Removes stopwords from text in whichever language is specified.

Removes stop words from a text string (adapted from 'litsearchr' <https://github.com/elizagrames/litsearchr/>) and returns the remaining words as a vector of strings

**Usage**

```
get_tokens(text, language = "English")
```

```
get_tokens(text, language = "English")
```

**Arguments**

text            An input string

language        The language used to look up stop words (default is "English")

**Value**

Returns the input text with stopwords removed.

A vector of strings consisting of the non-stop words from the 'text' input

**Examples**

```
get_tokens("On the Origin of Species", language="English")
text <- "A methodology for systematic mapping in environmental sciences"
tokens <- get_tokens(text)
tokens;
```

---

language_code	<i>Get short language codes</i>
---------------	---------------------------------

---

**Description**

This is a lookup function that returns the two-letter language code for specified language.

**Usage**

```
language_code(language)
```

**Arguments**

language      A character vector containing the name of a language.

**Value**

Returns a character vector containing a two-letter language code.

**Examples**

```
language_code("French")
```

---

possible_langs	<i>Languages codes synthesizr can recognize</i>
----------------	---

---

**Description**

A dataset of the languages that can be recognized by synthesizr along with their short form, character encoding, and whether a scientific journal indexed in 'ulrich' uses them.

**Usage**

```
possible_langs
```

**Format**

A database with 53 rows of 4 variables:

**Short** the short form language code

**Language** the name of the language

**Encoding** which character encoding to use for a language

**Used** whether or not the language is used by a scientific journal

**Source**

'litsearchr' package on 'Github'

**Examples**

```
possible_langs
```

---

```
remove_punctuation      Remove punctuation from text
```

---

**Description**

Removes common punctuation marks from a text.

**Usage**

```
remove_punctuation(text, preserve_punctuation = NULL)
```

**Arguments**

`text`                    A character vector from which to remove punctuation.  
`preserve_punctuation`    A string or vector of punctuation to retain

**Value**

Returns the input text with punctuation removed.

**Examples**

```
remove_punctuation("#s<<<//<y>!&^n$$t/>h%e&s$sis#!++r!//")
```

---

```
suggest_keywords      Suggest keywords
```

---

**Description**

Suggests possible keywords by extracting uni-, bi-, and tri-grams from a long text (e.g. article full text), having removed punctuation and stop words. Returns the remaining words as a vector of strings and assesses whether they are already present in the abstract or title

**Usage**

```
suggest_keywords(title, abstract, fulltext, suggest = FALSE)
```

**Arguments**

title	An article title
abstract	An article abstract
fulltext	An article full text
suggest	A logical argument of TRUE or FALSE. If TRUE, the output data frame returned is a subset that only includes potential keywords (i.e. those not already in the title or abstract)

**Value**

A data frame consisting of potential candidate keywords and their suitability. If suggest = FALSE, only good candidates are returned.

**Examples**

```

title <- "A methodology for systematic mapping in environmental sciences"
abstract <- "Systematic mapping was developed in social sciences in response to a lack of empirical
data when answering questions using systematic review methods, and a need for a method to describe
the literature across a broad subject of interest. Systematic mapping does not attempt to answer
a specific question as do systematic reviews, but instead collates, describes and catalogues
available evidence (e.g. primary, secondary, theoretical, economic) relating to a topic or
question of interest. The included studies can be used to identify evidence for policy-relevant
questions, knowledge gaps (to help direct future primary research) and knowledge clusters (sub-
sets of evidence that may be suitable for secondary research, for example systematic review).
Evidence synthesis in environmental sciences faces similar challenges to those found in social
sciences. Here we describe the translation of systematic mapping methodology from social sciences
for use in environmental sciences. We provide the first process-based methodology for systematic
maps, describing the stages involved: establishing the review team and engaging stakeholders;
setting the scope and question; setting inclusion criteria for studies; scoping stage; protocol
development and publication; searching for evidence; screening evidence; coding; production of a
systematic map database; critical appraisal (optional); describing and visualising the findings;
report production and supporting information. We discuss the similarities and differences in
methodology between systematic review and systematic mapping and provide guidance for those
choosing which type of synthesis is most suitable for their requirements. Furthermore, we discuss
the merits and uses of systematic mapping and make recommendations for improving this evolving
methodology in environmental sciences."
filepath <- system.file("extdata", "fulltext.rds", package="discoverableresearch")
fulltext <- readRDS(filepath)
fulltext <- gsub("\n", " ", fulltext)
fulltext <- gsub("\s+", " ", fulltext)
poss_keywords <- suggest_keywords(title, abstract, fulltext)
poss_keywords;

```

**Description**

Suggests possible title words by extracting uni-, 'bi-, and tri-grams from a long text (e.g. article full text), having removed punctuation and stop words. Returns the remaining words as a vector of strings and assesses whether they are already present in the title or abstract

**Usage**

```
suggest_title(abstract, keywords, fulltext, suggest = FALSE)
```

**Arguments**

abstract	An article abstract
keywords	An article keywords, supplied as a vector
fulltext	An article full text
suggest	A logical argument of TRUE or FALSE. If TRUE, the output data frame returned is sub-setting to only include potential keywords (i.e. those not already in the abstract or keywords)

**Value**

A data frame consisting of potential candidate title words and their suitability. If suggest = FALSE, only good candidates are returned.

**Examples**

```
abstract <- "Systematic mapping was developed in social sciences in response to a lack of empirical
data when answering questions using systematic review methods, and a need for a method to describe
the literature across a broad subject of interest. Systematic mapping does not attempt to answer
a specific question as do systematic reviews, but instead collates, describes and catalogues
available evidence (e.g. primary, secondary, theoretical, economic) relating to a topic or
question of interest. The included studies can be used to identify evidence for policy-relevant
questions, knowledge gaps (to help direct future primary research) and knowledge clusters (sub-
sets of evidence that may be suitable for secondary research, for example systematic review).
Evidence synthesis in environmental sciences faces similar challenges to those found in social
sciences. Here we describe the translation of systematic mapping methodology from social sciences
for use in environmental sciences. We provide the first process-based methodology for systematic
maps, describing the stages involved: establishing the review team and engaging stakeholders;
setting the scope and question; setting inclusion criteria for studies; scoping stage; protocol
development and publication; searching for evidence; screening evidence; coding; production of a
systematic map database; critical appraisal (optional); describing and visualising the findings;
report production and supporting information. We discuss the similarities and differences in
methodology between systematic review and systematic mapping and provide guidance for those
choosing which type of synthesis is most suitable for their requirements. Furthermore, we discuss
the merits and uses of systematic mapping and make recommendations for improving this evolving
methodology in environmental sciences."
keywords <- c("Systematic mapping",
"Evidence-based environmental management",
"Systematic evidence synthesis",
"Evidence review",
"Knowledge gaps",
```

```
"Knowledge clusters")
filepath <- system.file("extdata", "fulltext.rds", package="discoverableresearch")
fulltext <- readRDS(filepath)
fulltext <- gsub("\n", " ", fulltext)
fulltext <- gsub("\s+", " ", fulltext)
poss_titlewords <- suggest_title(abstract, keywords, fulltext)
poss_titlewords;
```

# Index

## \* datasets

possible\_langs, [10](#)

check\_fields, [2](#)

check\_keywords, [4](#)

check\_title, [5](#)

check\_title\_length, [6](#)

fakerake, [6](#)

format\_keywords, [7](#)

get\_ngrams, [7](#)

get\_stopwords, [8](#)

get\_tokens, [9](#)

language\_code, [10](#)

possible\_langs, [10](#)

remove\_punctuation, [11](#)

suggest\_keywords, [11](#)

suggest\_title, [12](#)