Package 'morphemepiece'

September 17, 2021

Type Package

Title Morpheme Tokenization

Version 1.0.1

Description Tokenize text into morphemes. The morphemepiece algorithm uses a lookup table to determine the morpheme breakdown of words, and falls back on a modified wordpiece tokenization algorithm for words not found in the lookup table.

URL https://github.com/macmillancontentscience/morphemepiece

BugReports https://github.com/macmillancontentscience/morphemepiece/issues

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.1.1

Imports dlr (>= 1.0.0), magrittr, morphemepiece.data, piecemaker (>= 1.0.0), purrr (>= 0.3.4), rlang, stringr (>= 1.4.0)

Suggests dplyr, fs, ggplot2, here, knitr, remotes, rmarkdown, testthat (>= 3.0.0), utils

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Jonathan Bratt [aut, cre] (<https://orcid.org/0000-0003-2859-0076>), Jon Harmon [aut] (<https://orcid.org/0000-0003-4781-4346>), Bedford Freeman & Worth Pub Grp LLC DBA Macmillan Learning [cph]

Maintainer Jonathan Bratt <jonathan.bratt@macmillan.com>

Repository CRAN

Date/Publication 2021-09-17 08:20:02 UTC

R topics documented:

`morphemepiece-package` *morphemepiece: Morpheme Tokenization*

Description

Tokenize words into morphemes (the smallest unit of meaning).

`load_lookup` *Load a morphemepiece lookup file*

Description

Usually you will want to use the included lookup that can be accessed via `morphemepiece_lookup()`. This function can be used to load a different lookup from a file.

Usage

```
load_lookup(lookup_file)
```

Arguments

`lookup_file`	path to lookup file. File is assumed to be a text file, with one word per line. The lookup value, if different from the word, follows the word on the same line, after a space.

Value

The lookup as a named list. Names are words in lookup.

load_or_retrieve_lookup

Load a lookup file, or retrieve from cache

Description

Usually you will want to use the included lookup that can be accessed via `morphemepiece_lookup()`. This function can be used to load (and cache) a different lookup from a file.

Usage

```
load_or_retrieve_lookup(lookup_file)
```

Arguments

`lookup_file`	path to lookup file. File is assumed to be a text file, with one word per line. The lookup value, if different from the word, follows the word on the same line, after a space.

Value

The lookup table as a named character vector.

load_or_retrieve_vocab

Load a vocabulary file, or retrieve from cache

Description

Usually you will want to use the included vocabulary that can be accessed via `morphemepiece_vocab()`. This function can be used to load (and cache) a different vocabulary from a file.

Usage

```
load_or_retrieve_vocab(vocab_file)
```

Arguments

`vocab_file`	path to vocabulary file. File is assumed to be a text file, with one token per line, with the line number (starting at zero) corresponding to the index of that token in the vocabulary.

Value

The vocab as a list of named integer vectors. Names are tokens in vocabulary, values are integer indices. The casedness of the vocabulary is inferred and attached as the "is_cased" attribute.

Note that from the perspective of a neural net, the numeric indices *are* the tokens, and the mapping from token to index is fixed. If we changed the indexing, it would break any pre-trained models. This is why the vocabulary is stored as a named integer vector, and why it starts with index zero.

`load_vocab` *Load a vocabulary file*

Description

Usually you will want to use the included vocabulary that can be accessed via `morphemepiece_vocab()`. This function can be used to load a different vocabulary from a file.

Usage

```
load_vocab(vocab_file)
```

Arguments

`vocab_file`	path to vocabulary file. File is assumed to be a text file, with one token per line, with the line number (starting at zero) corresponding to the index of that token in the vocabulary.

Value

The vocab as a named integer vector. Names are tokens in vocabulary, values are integer indices. The casedness of the vocabulary is inferred and attached as the "is_cased" attribute.

Note that from the perspective of a neural net, the numeric indices *are* the tokens, and the mapping from token to index is fixed. If we changed the indexing, it would break any pre-trained models using that vocabulary. This is why the vocabulary is stored as a named integer vector, and why it starts with index zero.

`morphemepiece_cache_dir`

Retrieve Directory for Morphemepiece Cache

Description

The morphemepiece cache directory is a platform- and user-specific path where morphemepiece saves caches (such as a downloaded lookup). You can override the default location in a few ways:

- Option: `morphemepiece.dir`Use `set_morphemepiece_cache_dir` to set a specific cache directory for this session
- Environment: `MORPHEMEPIECE_CACHE_DIR`Set this environment variable to specify a morphemepiece cache directory for all sessions.
- Environment: `R_USER_CACHE_DIR`Set this environment variable to specify a cache directory root for all packages that use the caching system.

Usage

```
morphemepiece_cache_dir()
```

Value

A character vector with the normalized path to the cache.

`morphemepiece_tokenize` *Tokenize Sequence with Morpheme Pieces*

Description

Given a single sequence of text and a morphemepiece vocabulary, tokenizes the text.

Usage

```
morphemepiece_tokenize(
  text,
  vocab = morphemepiece_vocab(),
  lookup = morphemepiece_lookup(),
  unk_token = "[UNK]",
  max_chars = 100
)
```

Arguments

`text`	Character scalar; text to tokenize.
`vocab`	Named integer vector containing vocabulary words. Should have "vocab_split" attribute, with components named "prefixes", "words", "suffixes".
`lookup`	A morphemepiece lookup table.
`unk_token`	Token to represent unknown words.
`max_chars`	Maximum length of word recognized.

Value

A character vector of tokenized text (later, this should be a named integer vector, as in the wordpiece package.)

`prepare_vocab` *Format a Token List as a Vocabulary*

Description

We use a special named integer vector with class morphemepiece_vocabulary to provide information about tokens used in `morphemepiece_tokenize`. This function takes a character vector of tokens and puts it into that format.

Usage

```
prepare_vocab(token_list)
```

Arguments

`token_list`	A character vector of tokens.

Value

The vocab as a named integer vector. Names are tokens in vocabulary, values are integer indices. The casedness of the vocabulary is inferred and attached as the "is_cased" attribute.

Note that from the perspective of a neural net, the numeric indices *are* the tokens, and the mapping from token to index is fixed. If we changed the indexing, it would break any pre-trained models using that vocabulary. This is why the vocabulary is stored as a named integer vector, and why it starts with index zero.

Examples

```
my_vocab <- prepare_vocab(c("some", "example", "tokens"))
class(my_vocab)
attr(my_vocab, "is_cased")
```

`set_morphemepiece_cache_dir` *Set a Cache Directory for Morphemepiece*

Description

Use this function to override the cache path used by morphemepiece for the current session. Set the `MORPHEMEPIECE_CACHE_DIR` environment variable for a more permanent change.

Usage

```
set_morphemepiece_cache_dir(cache_dir = NULL)
```

Arguments

`cache_dir`	Character scalar; a path to a cache directory.

Value

A normalized path to a cache directory. The directory is created if the user has write access and the directory does not exist.

Index