

# Package: wordpiece (via r-universe)

October 27, 2024

**Type** Package

**Title** R Implementation of Wordpiece Tokenization

**Version** 2.1.3

**Description** Apply 'Wordpiece' (<[arXiv:1609.08144](#)>) tokenization to input text, given an appropriate vocabulary. The 'BERT' (<[arXiv:1810.04805](#)>) tokenization conventions are used by default.

**Encoding** UTF-8

**URL** <https://github.com/macmillancontentscience/wordpiece>

**BugReports** <https://github.com/macmillancontentscience/wordpiece/issues>

**Depends** R (>= 3.3.0)

**License** Apache License (>= 2)

**RoxygenNote** 7.1.2

**Roxygen** list(markdown = TRUE)

**Imports** dlr (>= 1.0.0), fastmatch (>= 1.1), memoise (>= 2.0.0),  
piecemaker (>= 1.0.0), rlang, stringi (>= 1.0), wordpiece.data  
(>= 1.0.2)

**Suggests** covr, knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Repository** <https://jonthegeek.r-universe.dev>

**RemoteUrl** <https://github.com/macmillancontentscience/wordpiece>

**RemoteRef** HEAD

**RemoteSha** 3eb92c759556e89d235202c45dec2dc859e661d

## Contents

load_or_retrieve_vocab . . . . .	2
load_vocab . . . . .	2

prepare_vocab . . . . .	3
set_wordpiece_cache_dir . . . . .	4
wordpiece_cache_dir . . . . .	4
wordpiece_tokenize . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

load\_or\_retrieve\_vocab

*Load a vocabulary file, or retrieve from cache*

---

### Description

Load a vocabulary file, or retrieve from cache

### 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 corresponding to the index of that token in the vocabulary.

### Value

The vocab as a character vector of tokens. The casedness of the vocabulary is inferred and attached as the "is\_cased" attribute. The vocabulary indices are taken to be the positions of the tokens, *starting at zero* for historical consistency.

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 (the order of the tokens), it would break any pre-trained models.

---

load\_vocab

*Load a vocabulary file*

---

### Description

Load a vocabulary 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 corresponding to the index of that token in the vocabulary.

**Value**

The vocab as a character vector of tokens. The casedness of the vocabulary is inferred and attached as the "is\_cased" attribute. The vocabulary indices are taken to be the positions of the tokens, *starting at zero* for historical consistency.

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 (the order of the tokens), it would break any pre-trained models.

**Examples**

```
# Get path to sample vocabulary included with package.
vocab_path <- system.file("extdata", "tiny_vocab.txt", package = "wordpiece")
vocab <- load_vocab(vocab_file = vocab_path)
```

---

```
prepare_vocab
```

```
Format a Token List as a Vocabulary
```

---

**Description**

We use a special named integer vector with class `wordpiece_vocabulary` to provide information about tokens used in `wordpiece_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 character vector of tokens. The casedness of the vocabulary is inferred and attached as the "is\_cased" attribute. The vocabulary indices are taken to be the positions of the tokens, *starting at zero* for historical consistency.

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 (the order of the tokens), it would break any pre-trained models.

## Examples

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

---

set\_wordpiece\_cache\_dir

*Set a Cache Directory for wordpiece*

---

## Description

Use this function to override the cache path used by wordpiece for the current session. Set the WORDPIECE\_CACHE\_DIR environment variable for a more permanent change.

## Usage

```
set_wordpiece_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.

---

wordpiece\_cache\_dir      *Retrieve Directory for wordpiece Cache*

---

## Description

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

- Option: `wordpiece.dir` Use [set\\_wordpiece\\_cache\\_dir](#) to set a specific cache directory for this session
- Environment: `WORDPIECE_CACHE_DIR` Set this environment variable to specify a wordpiece 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

```
wordpiece_cache_dir()
```

**Value**

A character vector with the normalized path to the cache.

---

wordpiece\_tokenize      *Tokenize Sequence with Word Pieces*

---

**Description**

Given a sequence of text and a wordpiece vocabulary, tokenizes the text.

**Usage**

```
wordpiece_tokenize(  
  text,  
  vocab = wordpiece_vocab(),  
  unk_token = "[UNK]",  
  max_chars = 100  
)
```

**Arguments**

text	Character; text to tokenize.
vocab	Character vector of vocabulary tokens. The tokens are assumed to be in order of index, with the first index taken as zero to be compatible with Python implementations.
unk_token	Token to represent unknown words.
max_chars	Maximum length of word recognized.

**Value**

A list of named integer vectors, giving the tokenization of the input sequences. The integer values are the token ids, and the names are the tokens.

**Examples**

```
tokens <- wordpiece_tokenize(  
  text = c(  
    "I love tacos!",  
    "I also kinda like apples."  
  )  
)
```

# Index

load\_or\_retrieve\_vocab, 2  
load\_vocab, 2

prepare\_vocab, 3

set\_wordpiece\_cache\_dir, 4, 4

wordpiece\_cache\_dir, 4  
wordpiece\_tokenize, 3, 5