Package 'phrasemachine'

May 29, 2017

Type Package

Title Simple Phrase Extraction

Version 1.1.2

Date 2017-05-29

Author Matthew J. Denny, Abram Handler, Brendan O'Connor

Maintainer Matthew J. Denny <mdenny@psu.edu>

Description Simple noun phrase extraction using part-of-speech information. Takes a collection of un-processed documents as input and returns a set of noun phrases associated with those documents.

URL http://slanglab.cs.umass.edu/phrases/

License GPL-3

Imports NLP, openNLP, stringr

LazyData TRUE

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, quanteda

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-29 18:00:46 UTC

R topics documented:

coarsen_POS_tags 2
extract_ngram_filter 2
extract_phrases 3
phrasemachine 4
POS_tag_documents 5

Index **7**

coarsen_POS_tags *Coarsen POS tags*

Description

Coarsens PTB or Petrov/Gimpel coarse tags into one of eight categories: 'A' = adjective, 'D' = determiner, 'P' = preposition, 'N' = common/proper noun, 'M' = verb modifiers, 'V' = verbs, 'C' = coordinating conjunction, 'O' = all else NOTE: 'M', 'C', and 'V' tags are currently only compatible with the PTB tagset.

Usage

```
coarsen_POS_tags(tag_vector)
```

Arguments

tag_vector	A vector of POS tags.

Value

A vector of coarse tags.

Examples

```
pos_tags <- c("VB", "JJ", "NN", "NN")
coarsen_POS_tags(pos_tags)
```

extract_ngram_filter *Extract phrase spans*

Description

Takes a sequences of POS tags and a regex and returns spans which match regex.

Usage

```
extract_ngram_filter(pos_tags, regex, maximum_ngram_length,
  minimum_ngram_length)
```

Arguments

pos_tags	A character vector of Penn TreeBank or Petrov/Gimpel style tags.
regex	The regular expression (or vector of regular expressions) used to find phrases.
maximum_ngram_length	The maximum length phrases returned.
minimum_ngram_length	The minimum length phrases returned.

Value

A numeric matrix with two columns and rows equal to number of spans matched. First column is span start, second is span end.

Examples

```
pos_tags <- c("VB", "JJ", "NN", "NN")
spans <- extract_ngram_filter(pos_tags,
                              regex = "(A|N)*N(PD*(A|N)*N)*",
                              maximum_ngram_length = 8,
                              minimum_ngram_length = 1)
```

---

extract_phrases *Extract Phrases*

Description

Extracts phrases from a list of POS tagged document using the "FilterFSA" method in Handler et al. 2016.

Usage

```
extract_phrases(POS_tagged_documents, regex = "(A|N)*N(PD*(A|N)*N)*",
  maximum_ngram_length = 8, minimum_ngram_length = 2,
  return_phrase_vectors = TRUE, return_tag_sequences = FALSE)
```

Arguments

`POS_tagged_documents`
: A list object of the form produced by the 'POS_tag_documents()' function, with either Penn TreeBank or Petrov/Gimpel style tags.

`regex`
: The regular expression used to find phrases. Defaults to "(A|N)*N(PD*(A|N)*N)*", the "SimpleNP" grammar in Handler et al. 2016. A vector of regular expressions may also be provided if the user wishes to match more than one.

`maximum_ngram_length`
: The maximum length phrases returned. Defaults to 8. Increasing this number can greatly increase runtime.

`minimum_ngram_length`
: The minimum length phrases returned. Defaults to 2. Can be increased to remove shorter phrases, or decreased to include unigrams.

`return_phrase_vectors`
: Logical indicating whether a list of phrase vectors (with each entry contain a vector of phrases in one document) should be returned, or whether phrases should combined into a single space separated string.

`return_tag_sequences`
: Logical indicating whether tag sequences should be returned along with phrases. Defaults to FALSE.

Value

A list object.

Examples

```
## Not run:
# make sure quanteda is installed
requireNamespace("quanteda", quietly = TRUE)
# load in U.S. presidential inaugural speeches from Quanteda example data.
documents <- quanteda::data_corpus_inaugural
# use first 10 documents for example
documents <- documents[1:10,]

# run tagger
tagged_documents <- POS_tag_documents(documents)

phrases <- extract_phrases(tagged_documents,
                           regex = "(A|N)*N(PD*(A|N)*N)*",
                           maximum_ngram_length = 8,
                           minimum_ngram_length = 1)

## End(Not run)
```

---

phrasemachine *POS tag and extract phrases from a collection of documents*

Description

Extracts phrases from a set of documents using the "FilterFSA" method in Handler et al. 2016.

Usage

```
phrasemachine(documents, regex = "(A|N)*N(PD*(A|N)*N)*",
  maximum_ngram_length = 8, minimum_ngram_length = 2,
  return_phrase_vectors = TRUE, return_tag_sequences = FALSE,
  memory = "-Xmx512M")
```

Arguments

documents	A vector of strings (one per document).
regex	The regular expression used to find phrases. Defaults to "(A|N)*N(PD*(A|N)*N)*", the "SimpleNP" grammar in Handler et al. 2016. A vector of regular expressions may also be provided if the user wishes to match more than one.
maximum_ngram_length	The maximum length phrases returned. Defaults to 8. Increasing this number can greatly increase runtime.

`minimum_ngram_length`
: The minimum length phrases returned. Defaults to 2. Can be increased to remove shorter phrases, or decreased to include unigrams.

`return_phrase_vectors`
: Logical indicating whether a list of phrase vectors (with each entry contain a vector of phrases in one document) should be returned, or whether phrases should combined into a single space separated string.

`return_tag_sequences`
: Logical indicating whether tag sequences should be returned along with phrases. Defaults to FALSE.

`memory`
: The default amount of memory (512MB) assigned to the NLP package to POS tag documents is often not enough for large documents, which can lead to a "java.lang.OutOfMemoryError". The memory argument defaults to "-Xmx512M" (512MB) in this package, and can be increased if necessary to accommodate very large documents.

Value

A list object.

Examples

```
phrasemachine("Hello there my red good cat.")
```

`POS_tag_documents` *POS tag documents*

Description

Annotates documents (provided as a character vector with one entry per document) with pars-of-speech (POS) tags using the openNLP POS tagger

Usage

```
POS_tag_documents(documents, memory = "-Xmx512M")
```

Arguments

`documents`
: A vector of strings (one per document).

`memory`
: The default amount of memory (512MB) assigned to the NLP package to POS tag documents is often not enough for large documents, which can lead to a "java.lang.OutOfMemoryError". The memory argument defaults to "-Xmx512M" (512MB) in this package, and can be increased if necessary to accommodate very large documents.

Value

A list object.

Examples

```
## Not run:
# make sure quanteda is installed
requireNamespace("quanteda", quietly = TRUE)
# load some example data:
documents <- quanteda::data_corpus_inaugural

# run tagger
tagged_documents <- POS_tag_documents(documents)

## End(Not run)
```

Index

coarsen_POS_tags, 2

extract_ngram_filter, 2
extract_phrases, 3

phrasemachine, 4
POS_tag_documents, 5