Package 'phrasemachine'

October 24, 2016

Type Package

Title Simple Phrase Extraction

Version 1.0.0

Date 2016-10-23

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, quanteda

LazyData TRUE

RoxygenNote 5.0.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2016-10-24 17:14:25

R topics documented:

coarsen_POS_tags *Coarsen POS tags*

Description

Coarsens PTB or Petrov/Gimpel coarse tags into one of five categories: 'A' = adjective, 'D' = determiner, 'P' = preposition, 'N' = common/proper noun, 'O' = all else

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 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.

`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:
# load data
corp <- quanteda::corpus(quanteda::inaugTexts)
documents <- quanteda::texts(corp)[1:5]

# 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)
```

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.
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

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

POS_tag_documents *POS tag documents*

Description

Annotates documents (proived 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)
```

Arguments

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

Value

A list object.

Examples

```
## Not run:
# load data
corp <- quanteda::corpus(quanteda::inaugTexts)
documents <- quanteda::texts(corp)[1:5]

# run tagger
tagged_documents <- POS_tag_documents(documents)

## End(Not run)
```

Index