codecademy

Text Preprocessing

Text Preprocessing

In natural language processing, text preprocessing is the practice of cleaning and preparing text data. NLTK and `re` are common Python libraries used to handle many text preprocessing tasks.

Noise Removal

In natural language processing, *noise removal* is a text preprocessing task devoted to stripping text of formatting.

```
import re

text = "Five fantastic fish flew off to
find faraway functions. Maybe find another
five fantastic fish? Find my fish with a
function please!"

# remove punctuation
result = re.sub(r'[\.\?\!\,\:\;\"]', '',
text)

print(result)
# Five fantastic fish flew off to find
faraway functions Maybe find another five
fantastic fish Find my fish with a
function please
```

Tokenization

In natural language processing, *tokenization* is the text preprocessing task of breaking up text into smaller components of text (known as tokens).

```
from nltk.tokenize import word_tokenize

text = "This is a text to tokenize"
tokenized = word_tokenize(text)

print(tokenized)
# ["This", "is", "a", "text", "to",
"tokenize"]
```

Text Normalization

In natural language processing, *normalization* encompasses many text preprocessing tasks including stemming, lemmatization, upper or lowercasing, and stopwords removal.

Stemming

In natural language processing, *stemming* is the text preprocessing normalization task concerned with bluntly removing word affixes (prefixes and suffixes).

```
from nltk.stem import PorterStemmer

tokenized = ["So", "many", "squids",
"are", "jumping"]

stemmer = PorterStemmer()
stemmed = [stemmer.stem(token) for token
in tokenized]

print(stemmed)
# ['So', 'mani', 'squid', 'are', 'jump']
```

Lemmatization

In natural language processing, *lemmatization* is the text preprocessing normalization task concerned with bringing words down to their root forms.

```
from nltk.stem import WordNetLemmatizer

tokenized = ["So", "many", "squids",
"are", "jumping"]

lemmatizer = WordNetLemmatizer()
lemmatized = [lemmatizer.lemmatize(token)
for token in tokenized]

print(stemmed)
# ['So', 'many', 'squid', 'be', 'jump']
```

Stopword Removal

In natural language processing, *stopword removal* is the process of removing words from a string that don't provide any information about the tone of a statement.

```
from nltk.corpus import stopwords

# define set of English stopwords
stop_words =
set(stopwords.words('english'))

# remove stopwords from tokens in dataset
statement_no_stop = [word for word in
word_tokens if word not in stop_words]
```

Part-of-Speech Tagging

In natural language processing, *part-of-speech tagging* is the process of assigning a part of speech to every word in a string. Using the part of speech can improve the results of lemmatization.