Simple text preprocessing

Puteaux, Fall/Winter 2020-2021

```
##############################################
##                                          ##
##  Natural Language Processing in Python   ##
##                                          ##
##############################################
```

```
§1 Introduction to Natural Language Processing in Python
```

```
§1.2 Simple topic identification
```

1 Simple text preprocessing

1.1 Why preprocess?

- When performing machine learning or other statistical methods, it could help make for better input data.
- Examples:
  - *tokenization to create a bag of words*
  - *lowercasing words*
- Lemmatization/Stemming:
  - shorten words to their root stems
- Remove stop words, punctuation, or unwanted tokens.
- Good to experiment with different approaches.

1.2 Code of text preprocessing with Python:

[1]:
```
from nltk.tokenize import word_tokenize
from collections import Counter
```

[2]:
```
from nltk.corpus import stopwords

text = """The cat is in the box. The cat likes the box.
The box is over the cat."""
tokens = [w for w in word_tokenize(text.lower()) if w.isalpha()]
no_stops = [t for t in tokens if t not in stopwords.words('english')]
```

```
Counter(no_stops).most_common(2)
```

[2]:
```
[('cat', 3), ('box', 3)]
```

[3]:
```
from nltk.stem import WordNetLemmatizer

text = """Cats, dogs and birds are common pets. So are fish."""
tokens = [w for w in word_tokenize(text.lower()) if w.isalpha()]
no_stops = [t for t in tokens if t not in stopwords.words('english')]
wordnet_lemmatizer = WordNetLemmatizer()
lemmatized = [wordnet_lemmatizer.lemmatize(t) for t in no_stops]
print(lemmatized)
```

```
['cat', 'dog', 'bird', 'common', 'pet', 'fish']
```

1.3 Practice question for text preprocessing steps:

- Which of the following are useful text preprocessing steps?

  ☐ Stems, spelling corrections, lowercase.

  ☒ Lemmatization, lowercasing, removing unwanted tokens.

  ☐ Removing stop words, leaving in capital words.

  ☐ Strip stop words, word endings and digits.

1.4 Practice exercises for simple text preprocessing:

▶ Package pre-loading:

[4]:
```
from nltk import word_tokenize
from collections import Counter
```

▶ Data pre-loading:

[5]:
```
article = open('ref1. Wikipedia article - Debugging.txt').read()
tokens = word_tokenize(article)
lower_tokens = [t.lower() for t in tokens]
stopwords = open('ref2. English stopwords.txt').read()
english_stops = word_tokenize(stopwords)
```

▶ Text preprocessing practice:

[6]:
```
# Import WordNetLemmatizer
from nltk.stem import WordNetLemmatizer

# Retain alphabetic words: alpha_only
alpha_only = [t for t in lower_tokens if t.isalpha()]

# Remove all stop words: no_stops
```

```
no_stops = [t for t in alpha_only if t not in english_stops]

# Instantiate the WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()

# Lemmatize all tokens into a new list: lemmatized
lemmatized = [wordnet_lemmatizer.lemmatize(t) for t in no_stops]

# Create the bag-of-words: bow
bow = Counter(lemmatized)

# Print the 10 most common tokens
print(bow.most_common(10))
```

```
[('debugging', 40), ('system', 25), ('bug', 17), ('software', 16), ('problem',
15), ('tool', 15), ('computer', 14), ('process', 13), ('term', 13), ('debugger',
13)]
```