Documentation Classification Lab

- Target: Given a bunch of documents, try to classify the topic of the document.
- Steps:
  - Step 1, crawl, parse the documents. Prepare data
  - Step 2, convert the documents into features vectors
  - Step 3, using a classification algorithm to classify the data.

Prepare knowledge

Text classification lib

- Install lib and dependencies
  - nltk

```
pip3 install nltk
```

Run within python

```
import nltk
nltk.download('punkt')
```

In [ ]:
```
import nltk
sentence = u"Chúng ta thường nói đến Rau sạch, Rau an toàn để phân biệt với các rau bình thường bán ngoài chợ."
nltk.word_tokenize(sentence)
```

Vietnamese classification lib

- Another classifier is `underthesea` lib

```
pip3 install underthesea
pip3 install Cython
pip3 install future scipy numpy scikit-learn
pip3 install -U fasttext --no-cache-dir --no-deps --force-reinstall
```

Get the model via

```
underthesea data
```

Note: If you install on macOs, and face the error `option clang: error: unsupported option '-fopenmp'` Then you can try to point the GCC compiler to the brew's one.

```
brew install gcc g++
export CXX=/usr/local/bin/g++-7
export CC=/usr/local/bin/gcc-7
pip3 install underthesea -U
```

In [ ]:
```
from underthesea.classification import classify
classify(sentence)
```

Prepare the data

Sample data

- Sample data with newspaper

In [ ]:
```
from sklearn.datasets import fetch_20newsgroups
sample_data_train = fetch_20newsgroups(subset='train', shuffle=True)
print(f"Keys: {sample_data_train.keys()}")
print(f"Description: {sample_data_train.description}")
print(f"Data size: {len(sample_data_train.data)}")
print(f"Target name:\n{sample_data_train.target_names}")
```

In [ ]:
```
# Quick view sample data
sample_data_train.data[:10]
```

In [ ]:
```
# Quick view sample target
[sample_data_train.target_names[x] for x in sample_data_train.target[:10]]
```

Converts documents to features vectors

In this step we will use tf-idf (https://en.wikipedia.org/wiki/Tf%E2%80%93idf) to create features vectors.

Create Tokens

In [ ]:
```
# Translate text to tokens
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(sample_data_train.data)
print(X_train_counts)
```

Create tf-idf vector

In [ ]:
```
"""
Manual calc tf-idf
"""

def tf(word, blob):
    return blob.words.count(word) / len(blob.words)

def n_containing(word, bloblist):
    return sum(1 for blob in bloblist if word in blob)

def idf(word, bloblist):
    return math.log(len(bloblist) / (1 + n_containing(word, bloblist)))

def tfidf(word, blob, bloblist):
    return tf(word, blob) * idf(word, bloblist)
```

In [ ]:
```
"""
Using scikit library
"""
from sklearn.feature_extraction.text import TfidfTransformer
tf_transformer = TfidfTransformer(use_idf=False).fit(X_train_counts)
X_train_tf = tf_transformer.transform(X_train_counts)
X_train_tf.shape
```

Classify data

In [ ]:
```
from sklearn import neighbors
n_neighbors = 5

clf = neighbors.KNeighborsClassifier(n_neighbors)
clf.fit(X_train_tf,sample_data_train.target)
```

In [ ]:
```
test_data = ["hello is this me you looking for", "Theresa May is on the verge of pu
blicly blaming Russia for the attempted murder of Sergei and Yulia Skripal and orde
ring expulsions and sanctions against President Putin’s regime. An announcement cou
ld come as early as today after a meeting of the government’s National Security Cou
ncil"]
clf.predict(test_data)
```

In [ ]:
```
from sklearn.pipeline import Pipeline
text_clf = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer()),
                     ('clf', neighbors.KNeighborsClassifier(n_neighbors))
                    ])
text_clf.fit(sample_data_train.data, sample_data_train.target)
```

In [ ]:
```
test_predict_result=text_clf.predict(test_data)
test_predict_result
```

In [ ]:
```
[sample_data_train.target_names[x] for x in test_predict_result]
```

Exercises

Give the data file with records come from TuoiTre newspaper. Your task is to build a classifier which can predict the document topics Note:

- File `data-trim.csv` is the data sample.
- File tuoitre.csv (https://drive.google.com/file/d/0ByBWHzMQ2OtGd3VackZ3T1V4eVk/view?usp=sharing) is the full data.