The 20 newsgroups text dataset

The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date.

This module contains two loaders. The first one, :func:`sklearn.datasets.fetch_20newsgroups`, returns a list of the raw texts that can be fed to text feature extractors such as :class:`~sklearn.feature_extraction.text.CountVectorizer` with custom parameters so as to extract feature vectors. The second one, :func:`sklearn.datasets.fetch_20newsgroups_vectorized`, returns ready-to-use features, i.e., it is not necessary to use a feature extractor.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 12); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 12); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 12); *backlink*
>
> Unknown interpreted text role "func".

Data Set Characteristics:

Classes	20
Samples total	18846
Dimensionality	1
Features	text

Usage

The :func:`sklearn.datasets.fetch_20newsgroups` function is a data fetching / caching functions that downloads the data archive from the original 20 newsgroups website, extracts the archive contents in the `~/scikit_learn_data/20news_home` folder and calls the :func:`sklearn.datasets.load_files` on either the training or testing set folder, or both of them:

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 33); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 33); *backlink*
>
> Unknown interpreted text role "func".

```
>>> from sklearn.datasets import fetch_20newsgroups
>>> newsgroups_train = fetch_20newsgroups(subset='train')

>>> from pprint import pprint
>>> pprint(list(newsgroups_train.target_names))
['alt.atheism',
 'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware',
 'comp.sys.mac.hardware',
 'comp.windows.x',
 'misc.forsale',
```

```
 'rec.autos',
 'rec.motorcycles',
 'rec.sport.baseball',
 'rec.sport.hockey',
 'sci.crypt',
 'sci.electronics',
 'sci.med',
 'sci.space',
 'soc.religion.christian',
 'talk.politics.guns',
 'talk.politics.mideast',
 'talk.politics.misc',
 'talk.religion.misc']
```

The real data lies in the `filenames` and `target` attributes. The target attribute is the integer index of the category:

```
>>> newsgroups_train.filenames.shape
(11314,)
>>> newsgroups_train.target.shape
(11314,)
>>> newsgroups_train.target[:10]
array([ 7,  4,  4,  1, 14, 16, 13,  3,  2,  4])
```

It is possible to load only a sub-selection of the categories by passing the list of the categories to load to the :func:`sklearn.datasets.fetch_20newsgroups` function:

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 76); *backlink*
>
> Unknown interpreted text role "func".

```
>>> cats = ['alt.atheism', 'sci.space']
>>> newsgroups_train = fetch_20newsgroups(subset='train', categories=cats)

>>> list(newsgroups_train.target_names)
['alt.atheism', 'sci.space']
>>> newsgroups_train.filenames.shape
(1073,)
>>> newsgroups_train.target.shape
(1073,)
>>> newsgroups_train.target[:10]
array([0, 1, 1, 1, 0, 1, 1, 0, 0, 0])
```

Converting text to vectors

In order to feed predictive or clustering models with the text data, one first need to turn the text into vectors of numerical values suitable for statistical analysis. This can be achieved with the utilities of the `sklearn.feature_extraction.text` as demonstrated in the following example that extract TF-IDF vectors of unigram tokens from a subset of 20news:

```
>>> from sklearn.feature_extraction.text import TfidfVectorizer
>>> categories = ['alt.atheism', 'talk.religion.misc',
...               'comp.graphics', 'sci.space']
>>> newsgroups_train = fetch_20newsgroups(subset='train',
...                                       categories=categories)
>>> vectorizer = TfidfVectorizer()
>>> vectors = vectorizer.fit_transform(newsgroups_train.data)
>>> vectors.shape
(2034, 34118)
```

The extracted TF-IDF vectors are very sparse, with an average of 159 non-zero components by sample in a more than 30000-dimensional space (less than .5% non-zero features):

```
>>> vectors.nnz / float(vectors.shape[0])
159.01327...
```

:func:`sklearn.datasets.fetch_20newsgroups_vectorized` is a function which returns ready-to-use token counts features instead of file names.

> **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 119); *backlink*
>
> Unknown interpreted text role "func".

Filtering text for more realistic training

It is easy for a classifier to overfit on particular things that appear in the 20 Newsgroups data, such as newsgroup headers. Many classifiers achieve very high F-scores, but their results would not generalize to other documents that aren't from this window of time.

For example, let's look at the results of a multinomial Naive Bayes classifier, which is fast to train and achieves a decent F-score:

```
>>> from sklearn.naive_bayes import MultinomialNB
>>> from sklearn import metrics
>>> newsgroups_test = fetch_20newsgroups(subset='test',
...                                      categories=categories)
>>> vectors_test = vectorizer.transform(newsgroups_test.data)
>>> clf = MultinomialNB(alpha=.01)
>>> clf.fit(vectors, newsgroups_train.target)
MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True)

>>> pred = clf.predict(vectors_test)
>>> metrics.f1_score(newsgroups_test.target, pred, average='macro')
0.88213...
```

(The example :ref:`sphx_glr_auto_examples_text_plot_document_classification_20newsgroups.py` shuffles the training and test data, instead of segmenting by time, and in that case multinomial Naive Bayes gets a much higher F-score of 0.88. Are you suspicious yet of what's going on inside this classifier?)

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 150); *backlink***
>
> Unknown interpreted text role "ref".

Let's take a look at what the most informative features are:

```
>>> import numpy as np
>>> def show_top10(classifier, vectorizer, categories):
...     feature_names = vectorizer.get_feature_names_out()
...     for i, category in enumerate(categories):
...         top10 = np.argsort(classifier.coef_[i])[-10:]
...         print("%s: %s" % (category, " ".join(feature_names[top10])))
...
>>> show_top10(clf, vectorizer, newsgroups_train.target_names)
alt.atheism: edu it and in you that is of to the
comp.graphics: edu in graphics it is for and of to the
sci.space: edu it that is in and space to of the
talk.religion.misc: not it you in is that and to of the
```

You can now see many things that these features have overfit to:

- Almost every group is distinguished by whether headers such as `NNTP-Posting-Host:` and `Distribution:` appear more or less often.
- Another significant feature involves whether the sender is affiliated with a university, as indicated either by their headers or their signature.
- The word "article" is a significant feature, based on how often people quote previous posts like this: "In article [article ID], [name] <[e-mail address]> wrote:"
- Other features match the names and e-mail addresses of particular people who were posting at the time.

With such an abundance of clues that distinguish newsgroups, the classifiers barely have to identify topics from text at all, and they all perform at the same high level.

For this reason, the functions that load 20 Newsgroups data provide a parameter called **remove**, telling it what kinds of information to strip out of each file. **remove** should be a tuple containing any subset of `('headers', 'footers', 'quotes')`, telling it to remove headers, signature blocks, and quotation blocks respectively.

```
>>> newsgroups_test = fetch_20newsgroups(subset='test',
...                                      remove=('headers', 'footers', 'quotes'),
...                                      categories=categories)
>>> vectors_test = vectorizer.transform(newsgroups_test.data)
>>> pred = clf.predict(vectors_test)
>>> metrics.f1_score(pred, newsgroups_test.target, average='macro')
0.77310...
```

This classifier lost over a lot of its F-score, just because we removed metadata that has little to do with topic classification. It loses even more if we also strip this metadata from the training data:

```
>>> newsgroups_train = fetch_20newsgroups(subset='train',
...                                       remove=('headers', 'footers', 'quotes'),
```

```
...                                        categories=categories)
>>> vectors = vectorizer.fit_transform(newsgroups_train.data)
>>> clf = MultinomialNB(alpha=.01)
>>> clf.fit(vectors, newsgroups_train.target)
MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True)
```

```
>>> vectors_test = vectorizer.transform(newsgroups_test.data)
>>> pred = clf.predict(vectors_test)
>>> metrics.f1_score(newsgroups_test.target, pred, average='macro')
0.76995...
```

Some other classifiers cope better with this harder version of the task. Try running :ref:`sphx_glr_auto_examples_model_selection_grid_search_text_feature_extraction.py` with and without the `--filter` option to compare the results.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 218); *backlink*

Unknown interpreted text role "ref".

Data Considerations

The Cleveland Indians is a major league baseball team based in Cleveland, Ohio, USA. In December 2020, it was reported that "After several months of discussion sparked by the death of George Floyd and a national reckoning over race and colonialism, the Cleveland Indians have decided to change their name." Team owner Paul Dolan "did make it clear that the team will not make its informal nickname -- the Tribe -- its new team name." "Itâ€™s not going to be a half-step away from the Indians," Dolan said."We will not have a Native American-themed name."

https://www.mlb.com/news/cleveland-indians-team-name-change

Recommendation

- When evaluating text classifiers on the 20 Newsgroups data, you should strip newsgroup-related metadata. In scikit-learn, you can do this by setting `remove=('headers', 'footers', 'quotes')`. The F-score will be lower because it is more realistic.
- This text dataset contains data which may be inappropriate for certain NLP applications. An example is listed in the "Data Considerations" section above. The challenge with using current text datasets in NLP for tasks such as sentence completion, clustering, and other applications is that text that is culturally biased and inflammatory will propagate biases. This should be taken into consideration when using the dataset, reviewing the output, and the bias should be documented.

Examples

- :ref:`sphx_glr_auto_examples_model_selection_grid_search_text_feature_extraction.py`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 251); *backlink*

  Unknown interpreted text role "ref".

- :ref:`sphx_glr_auto_examples_text_plot_document_classification_20newsgroups.py`

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\scikit-learn-main\sklearn\datasets\descr\(scikit-learn-main)(sklearn)(datasets)(descr)twenty_newsgroups.rst, line 253); *backlink*

  Unknown interpreted text role "ref".