# CS691CL: Final Project

## `README`

Monath, Schulze, Zaporojets

March 31, 2014

# 1 Preliminaries

- Download & Install the Anaconda Distribution of Python from here:
  - https://store.continuum.io/cshop/anaconda/
- Set Anaconda to be your default version of Python

# 2 Data Format Specifications

Following the format specified in the `Scikit-Learn Dataset`, we can use the following data structure format for our data sets:

- Suppose we define the following function for loading the data from files on disk to the data structure `dataset` in memory.

```
dataset = load_data("DirectoryName")
```

- `dataset` will be a **bunch** datatype, which is a **dictionary** that supports dot (.) style field-/attribute access.
- `dataset` has four fields:
  - `data` - A **list** such that each element is the contents of one of the text files in the data set. Let $N$ be the length of the list.
  - `filenames` - A $N$-by-1 **ndarray** such that each entry of `filenames` is the filename of the corresponding text entry in `data`
  - `target` - A $N$-by-1 **ndarray** of integers such that each entry of `target` is the class label (an integer) of the corresponding text entry in `data`. Let there be $C$ distinct class labels.
  - `target_names` - A $C$-element **list** such that the integer class label `i` has the name `target_names[i]`

# 3 Feature Format Specifications

By *feature* or *feature vector*, I mean the *document representation*, e.g. unigram bag-of-words or bag of dependency pairs, etc. This section defines the format in which we will store these representations in `Python`.

- Following the format of the bag-of-words feature extractor provided by `Scikit-Learn`, we can define a method which extracts the features out of all the documents in a `dataset`.

```
dataset = load_data("DirectoryName")
features = extract_features(dataset, "feature type")
```

- `features` will be a $N$-by-$M$ matrix, with $N$ as the number of documents in the dataset and $M$ is the length of the feature vectors (e.g. the number of words or dependency pairs in all documents). `features` will be a sparse matrix in the format **csr_matrix**

# 4 Reuters-21578

- Which documents are in the training set and which are in the testing set is defined by the ModApte split. See Appendix A for a full explanation of this division.
- The ten most frequent classes, defined by Natase et al are: *acq, corn, crude, earn, grain, interest, money-fx, ship, trade, wheat.* **The category *money-fx* seems to be the combination of two categories in the dataset. We will have to be careful. It's unclear if it means the union or intersection of the two**
- Filter out documents that contain no text
- Filter out documents that do not belong to one of the 10 most frequent classes
- Filter out features that do not appear in at least 2 documents
- Filter out documents that have an all 0 representation vector. (How could you have this? A document of entirely stop words?)
- Important note: Documents can have more than 1 class label!!!!

# A ModApte Split - from Reuters-21578 ReadMe

```
VIII.B. The Modified Apte ("ModApte") Split :

 Training Set (9,603 docs): LEWISSPLIT="TRAIN";  TOPICS="YES"
 Test Set (3,299 docs): LEWISSPLIT="TEST"; TOPICS="YES"
 Unused (8,676 docs):   LEWISSPLIT="NOT-USED"; TOPICS="YES"
                      or TOPICS="NO"
                      or TOPICS="BYPASS"

This replaces the 10645/3672 split (7,856 not used) of the
Reuters-22173 collection.  These are our best approximation to the
training and test splits used in APTE94 and APTE94b. Note the
following:

      1. As with the ModLewis, those documents removed in forming
Reuters-21578 are not present, and BYPASS documents are not used.
      2. The intent in APTE94 and APTE94b was to use the Lewis split,
but restrict it to documents with at least one TOPICS categories.
However, but it was not clear exactly what Apte, et al meant by having
at least one TOPICS category (e.g. how was "bypass" treated, whether
this was before or after any fixing of typographical errors, etc.). We
have encoded our interpretation in the TOPICS attribute.  ***Note
that, as discussed above, some TOPICS="YES" stories have no TOPICS
categories, and a few TOPICS="NO" stories have TOPICS
categories. These facts are irrelevant to the definition of the
split.*** If you are using a learning algorithm that requires each
training document to have at least TOPICS category, you can screen out
the training documents with no TOPICS categories. Please do NOT screen
out any of the 3,299 documents - that will make your results
incomparable with other studies.

      3. As with ModLewis, it may be desirable to use the 8,676 Unused
documents for gathering statistical information about feature
distribution.

As with ModLewis, this split assigns documents from April 7, 1987 and
before to the training set, and documents from April 8, 1987 and after
to the test set.  The difference is that only documents with at least
one TOPICS category are used.  The rationale for this restriction is
that while some documents lack TOPICS categories because no TOPICS
apply (i.e. the document is a true negative example for all TOPICS
categories), it appears that others simply were never assigned TOPICS
categories by the indexers. (Unfortunately, the amount of time that
has passed since the collection was created has made it difficult to
establish exactly what went on during the indexing.)

WARNING: Given the many changes in going from Reuters-22173 to
Reuters-21578, including correction of many typographical errors in
category labels, results on the ModApte split cannot be compared
with any published results on the Reuters-22173 collection!
```