Constellate notebooks workshop notes – Steve Baskauf – 2022-09-21

Key points:
Constellate datasets are JSONL files containing metadata, n-grams, and sometimes full text. They get loaded in the notebook by ID number.

By default a sample of 1500 documents is used in notebooks. Full datasets must be downloaded, see “working with dataset files” lesson for details.

gzip (GNU zip) is a Linux file compression format that can be used in streaming. Compressed files have the .gz extension. Constellate JSONL files are gzip compressed. The dataset_reader() method is used to uncompress and load a single row (i.e. one document). See the "Working with Dataset Files" lesson for details.

Important terms:
corpus (plural: *corpora*) – a text or set of texts used for study
token – a sequence of characters to be treated as a group (often a word)
tuple – immutable data structure containing a sequence of values, e.g. (2,5) or ('cat', True, -5)
counter – a Python data structure like a dictionary, but ordered and whose values are counts. Differences in behavior:

- returns 0 if key doesn’t exist
- sortable
- referenceable by position
- faster

gensim dictionary - unique tokens (key) and unique IDs for them (value). It's NOT a Python dictionary. Rather it's a special data structure. The point of this is to make it possible to refer to the tokens as only numbers and take advantage of increased speed for processing numeric data. Note: name comes from the genism package.
bow (bag of words) corpus – a list of tuples where the first number is the genism token ID and the second is the word frequency.

Packages used:
From the standard library:
collections – alternative objects to built-in ones like list, dict, etc. See lesson on “Counter Objects” for details.

Constellate-specific
constellate – client library for loading and manipulating Constellate datasets

Commonly used data science packages:
numpy – for numerical arrays
pandas – for manipulating data frames (i.e. tables)
matplotlib – for graphics
sklearn – scikit-learn machine learning library.

Text analysis packages:
nltk – Natural Language Tool Kit
spacy – open source natural language processing (NLP) library (using memory managed Cython) for large-scale information extraction
gensim – information retrieval and NLP on large corpora
vaderSentiment – used to generate a VADER (Valence Aware Dictionary for sEntiment Reasoning) lexicon

Notebooks:
Common pattern at the beginning of the notebooks:

1. Download the Constellate dataset into the notebook environment.
2. Apply pre-processing filters
3. Sometimes load stopwords

The following are functional (non-remedial) notebooks:

Exploring Metadata and Pre-Processing – basically wrangling of various forms using pandas. Creates a pre-processing filter to reduce the size of datasets and speed up analysis.

Creating a Stopwords List – can use built-in ones from NLTK, spaCy, or Gensim. Basically this short lesson puts a stopword list into a CSV.

Exploring Word Frequencies – get word counts after filtering with stop words, then visualizing

Finding Significant Words using TF/IDF (term frequency-inverse document frequency) – The TF-IDF score for a word in a document is larger if word is found frequently in a certain document and larger if it's rarely found in other documents. So it looks for words that are characteristic to the particular document. Can be useful for finding names particular to a document, as well as to detect bad OCR (optical character recognition).

Sentiment Analysis with VADER – uses a rule-based system (vs. machine learning) developed for social media. Uses more than unigrams, so can distinguish "very good" and "good". Note: the VADER example is embedded in the text and the sklearn example downloads a dataset, so neither uses the Constellate ones. They are really trained on social media posts, so probably wouldn't work that well on the articles.

LDA Topic modeling – trains a Latent Dirichlet Allocation (LDA) modeling to find topics (groups of words that occur together). Screens out words that are in 90% of docs or fewer than 50 docs to reduce processing. Can set number of topics and number of training bouts ("passes"). Perplexity score measures how successfully the trained model predicts new data (?). Goal is to get lowest perplexity. The topics will change as the number of topics specified changes. NOTE: the methodology is kinda weird. The unigrams are just listed the number of times they were counted, so the analysis makes no use of the nearness of one word to another.