Exploratory Text Analytics Final Project

Overview

The goal of your final project is to apply and integrate what you have learned in this course to create **a digital critical edition of a corpus** that will support exploration of the cultural contents of that corpus. Cultural content is broadly conceived—it may be about language use, social events, cultural categories, sentiments, identity, taste, etc.

Specifically, you will acquire a collection of long-form texts and perform the following operations:

1. **Convert** the collection from their source formats (F0) into a set of tables that conform to the Standard Text Analytic Data Model (F2) and to the Machine Learning Corpus Format (F1).
2. **Annotate** these tables with statistical and linguistic features using NLP libraries such as NLTK (F3).
3. **Create** a vector representation of the corpus to generate TFIDF values to add to the TOKEN and VOCAB tables (F4).
4. **Extend** the annotated and vectorized model with tables and features derived from the application of various unsupervised methods, including PCA, LDA, and word2vec (F5).
5. **Explore** your results using statistical and visualization methods.
6. **Present** conclusions about cultural patterns observed in the corpus by means of these operations.

Deliverables

To receive full credit for the assignment, you must produce **a digital analytical edition** of a corpus. This edition should include the following deliverables to be uploaded or linked to the Assignment for the Final Project in Collab.

- A **collection of source files** compressed in an archive (e.g., zip or tar.gz) and hosted on your UVA Box account.
  - A **manifest** file describing those sources files, including their:
    - **Provenance**: Where did they come from? Describe the website or other source and provide relevant URLs.
    - **Location**: Provide a link to the source files in UVA Box.
    - **Description**: What is the general subject matter of the corpus?
    - **Format**: A description of both the file formats of the source files, e.g., plaintext, XML, CSV, etc., and the internal structure where applicable. For example, if XML then specify document type (e.g., TEI or XHTML).
- A collection of **data files**, in either CSV or SQLite format, containing the F2 through F5 data you extracted from the corpus.

  - These files should include, at a minimum, the following core tables:
    - LIBRARY.csv — Metadata for the source files
    - TOKEN.csv — Annotated with statistical and linguistic features
    - VOCAB.csv — Annotated with statistical and linguistic features
  - In addition, you should include the following data in your files, either as features in the appropriate core table or as separate tables. Note that all tables should have an appropriate index and, where appropriate, an OCHO index.
    - **Principal Components**
      - Table of documents and components.
      - Table of components and word counts (i.e., the “loadings”), either added to the VOCAB table or as a separate table with a shared index with the VOCAB table.
    - **Topic Models (LDA)**
      - Table of document and topic concentrations.
      - Table of topics and term counts, either added to the VOCAB table or as a separate table with a shared index with the VOCAB table.
    - **Word Embeddings (word2vec)**
      - Terms and embeddings, either added to the VOCAB table or as a separate table with a shared index with the VOCAB table.
    - **Sentiment Analysis**
      - Sentiment and emotion values as features in VOCAB or as a separate table with a shared index with the VOCAB table.
      - Sentiment polarity and emotions for each document.
- The **Jupyter notebooks** used to perform all operations that produced the data in your tables.
- One or more **Jupyter notebooks** that explore, visualize, and interpret the data. You should use at least three of the following visualization types (beyond simple bar and pie charts):
    - Hierarchical cluster diagrams
    - Heatmaps showing correlations
    - Scatter plots
    - KDE plots
    - Dispersion plots
    - t-SNE plots
- Any **Python files** written (e.g., `.py` files) written to support your work.

- Any **other assets–**e.g., images, stylesheets, JavaScript libraries, etc.–required by your notebooks.
- A **two- to four-page final report** presenting and interpreting the results of your work**.** This document can contain as many visualizations as necessary to make your points, but the page count refers to text only.

Format and Style Guides

Any nondata files you produce, such as a Jupyter notebook, a manifest, or a Python program, should contain a header stating your name and email address, the name of this class (DS 5001), and the date. It should look something like this (depending on the document):

> Rafael Alvarado (rca2t@virginia.edu)
> DS 5001
> 26 January 2020

Jupyter notebooks should be properly outlined with headers and explanatory text where necessary to follow what is happening.

The final report may be written using any word processor (e.g., Word or Google Docs), but please upload it as a PDF. The document should have 12-point font and 1-inch margins, and should contain images. The images, however, do not count toward the final page count.

Group Work

Students may work in groups. Ideally, these will be composed of three students, but may contain two or four if necessary. In these groups, students may collaborate on the work of acquiring, consolidating, and modifying the source data. In addition, students in groups are free to share code and ideas. However, each student is responsible for **their own deliverables**. In particular, the observations made in the final report must be unique to each student. Of course, there will be some overlap of ideas, but students must demonstrate that they have engaged individually with the material by writing up their own conclusions and expressing them in their own language.

Grading Rubric

Given that the focus of this course is on method and not domain knowledge per se (although we have covered a good bit of that), you may be liberal in your interpretations. That is, do not worry about whether they will meet high scholarly and scientific standards. Remember, the purpose of ETA is to open up texts so that you may *explore* them and extract possibly significant patterns from them. However, you are expected to present your conclusions in a coherent and compelling manner. And, if you do find that you have discovered something interesting about your data beyond the requirements of the assignment, by all means consider pursuing it beyond this course.

● Deliverables	50%
● Legibility and meeting of format standards	10%
● Quality of Final Report	40%

Appendix: Forms of Text Data

F0 **Source Format**. The initial source format of a text, which varies by collection, e.g. XML (e.g. TEI and RSS), HTML, plain text (e.g. Gutenberg), JSON, and CSV.

F1 **Machine Learning Corpus Format (MLCF)**. Ideally a table of minimum discursive units indexed by document content hierarchy.

F2 **Standard Text Analytic Data Model (STADM)**. A normalized set of tables including DOC, TOKEN, and TERM tables. Produced by the tokenization of F1 data.

F3 **NLP Annotated STADM**. STADM with annotations added to token and term records indicating stopwords, parts-of-speech, stems and lemmas, named entities, grammatical dependencies, sentiments, etc.

F4 **STADM with Vector Space models**. Vector space representations of TOKEN data and resulting statistical data, such as term frequency and TFIDF.

F5 **STADM with analytical models**. STADM with columns and tables added for outputs of fitting and transforming models with the data.

F6 **STADM converted into interactive visualization**. STADM represented as a database-driven application with interactive visualization, .e.g. Jupyter notebooks and web applications.