Foundations of Text Analysis

Description

The purpose of this 5-week workshop is to increase the knowledge of text mining principles among participants. By the end of the workshop, students will be able to describe the range of basic text mining techniques (everything from the creation of a corpus, to the counting/tabulating of words, to classification and clustering, and visualizing the results of text analysis) and have garnered hands-on experience with all of them. All the materials for this workshop are available online. [0]

Prerequisites and Recommendation

There are three prerequisites: 1) a sincere willingness to learn, 2) a willingness to work at a computer's command-line interface, and 3) a commitment to attending every session. Students are strongly encouraged to bring their own computers to class.

Instructor Information and Office Hours

The instructor is Eric Lease Morgan <emorgan@nd.edu>, a Digital Initiatives Librarian in the Hesburgh Libraries at the University of Notre Dame. He conducted his first text mining investigation in 2010 when he measured the "greatness" of *The Great Books Of The Western World*. Eric has been writing software since 1976. If you have any questions during the five weeks of the workshop, then just drop Eric a line, and impromptu office hours will be created.

Readings

Readings will come from a book entitled Natural Language Processing With Python, and it is freely available online under a Creative Commons license. [1, 2] Please see the weekly session summaries below for reading assignments.

Outline and learning objectives

The workshop series is divided into the following sections, one per week:

1. Overview of text mining and working from the command line
2. Building a corpus
3. Word and phrase frequencies
4. Extracting meaning with dictionaries, parts-of-speech analysis, and named entity recognition

5. Classification and topic modeling

Session Summaries

Week 1 - Overview of text mining and working from the command line

Text mining is a process enabling the researcher to analyze arbitrarily large volumes of text for the purposes of discovering patterns and anomalies. These large volumes of text called corpora might be a single novel, a collection of novels, a set of academic journal articles, the qualitative results of a survey, an entire website, sets of "tweets", etc. By counting and tabulating the words in a corpus, the reader can begin to learn of the corpus's "aboutness". Since language follows sets of loosely articulated rules, the meaning of the words can be inferred through the use of lexical tools like dictionaries and parts-of-speech parsers. All of these things are the building blocks for answering a set of upper-level questions such as but not limited to: what is the size of a given writer's vocabulary, what things (people, places, ideas, etc.) predominate a corpus, how has the meaning of a given word or phrase changed over time, what sorts of actions takes place in a corpus, to what degree is the overall tone of a corpus positive or negative. Text mining is a "distant reading" process. It both supplements and complements the "close reading" process. Text mining is not a replacement for the traditional reading process we have all come to know and love.

With the exception of Voyant Tools and to a lesser extent GATE, there are few holistic applications for doing text mining. [3, 4] Instead a combination of database applications, programming languages, and command-line utilities make up the bulk of text mining software, and thus, it is all but imperative to be familiar with a computer's command-line interface.

Exercise #1 Use Voyant Tools and familiarize yourself with the command line

- Use Voyant Tools' Voyeur to do a bit of text mining on your text of choice, or, on The Adventures of Sherlock Holmes (http://voyant-tools.org/)
- Show also http://hermeneuti.ca/voyeur/tools (Emphasize that there is more
- Use SSH to log into the workshop host computer
- Use a set of Linux commands to interact with the computer's file system: ls, touch, rm, mkdir, rmdir, mv, cp, cat, more, less, man, etc.
- Use a number of different methods to read Linux help texts, such as:
    - `man -k keyword | more`
    - `man ftp`
    - `info cp | more`
- Use grep and wc - on a text of your choice or Sherlock - to do simple text analysis

Week 2 - Building a corpus & Preprocessing Files

The process of text mining is not possible without a corpus to analyze. Corpora are sets of "plain text" documents, meaning files containing nothing but words, numbers, and symbols. These documents are "plain" in that they contain no formatting. No bolding. No italics. No fonts. Plain text files are often saved using the .txt filename extension.

If you are starting out with word processor files (like Word), then you might want to save those files as text files before any work is done against them. PDF is a common format for many documents, and many PDF documents are not necessarily facsimiles, but rather many PDF documents have the text embedded in them. Web pages are often the basis of corpora, but it will be necessary to remove any HTML markup before analysis can be done. Some documents are simply scans of their originals, and these files will need have OCR done against them. OCR (optical character recognition) converts image files (JPEG, GIF, TIFF, etc.) depicting words into plain text files. OCR can be done against PDF documents too. The OCR process is often not perfect. Images of older printed pages are less likely to be recognizable by the OCR application because the fonts used to depict letters are often more stylized.

Saving many plain text files in one or more directories is not always the best way to organize a corpus. Instead, it is often a good idea to manifest your corpus as an organized list a database. Fields in the database might include but are not limited to: title, author, date, source, URL, and filename. This sort of "metadata" will facilitate finer grained analysis.

Exercise #2 - Create a corpus

- Use wget to build a corpus of plain text documents from the Web
- Programmatically harvest articles from JSTOR (bin/harvest.pl)
- Use tika to extract the plain text from PDF documents (http://tika.apache.org)

Week 3 - Word and phrase frequencies

The whole of text mining is based on the counting of words. Once sets of words are counted the same words can be tabulated, thus becoming frequencies. Once words are tabulated measurements have taken place, and sets of measurements lead to observations. Observations can be charted, graphed, and visualized. From here patterns and anomalies become apparent. Once patterns and anomalies are articulated so do overarching descriptions and possibly predictions. And finally, descriptions and predictions lead to knowledge, not mere information. It all starts with the counting and tabulating of words.

Developed during the late Middle Ages, concordances are probably one of the oldest of text mining tools. They first count the number of times words (or phrases) occur in a corpus and

display the word in the context of the other words surrounding them. The modern-day term for concordance is key word in context (KWIK).

Exercise #3 - Learn about a corpus through frequencies

- Use Python's Natural Language Toolkit (NLTK) to read a file and report on its length,most frequently used words, and most frequently used ngrams (bin/frequency.py, bin/ngrams.py)
- Use the NLTK and its' concordance feature to search a file and display the result (bin/concordance.py)
- Use the NLTK to list "similar" words. (bin/similar.py)

Week 4 - Extracting meaning with dictionaries, parts-of-speech analysis, and named entity recognition

The previously discussed topics are limited to the counting and display of words, and the meanings of the words are only implied. By looking up words in dictionaries, by performing parts-of-speech (POS) analysis, and through named entity recognition, higher orders of investigation can be done, but these types of investigations are limited to specific languages.

Generally speaking (no puns intended) human languages are made up of a number of parts, such as but not limited to: nouns, verbs, adjectives, etc. These are called parts-of-speech, and they can be extracted from texts for further analysis. For example, all the adjectives could be tabulated from a corpus to determine peoples' feelings or sentiment. Personal pronouns can be listed and tabulated to determine the types of people in a novel. Verbs can be enumerated in order to figure out what actions are depicted in a text. Named entities are a subset of nouns. They include things like the people, places, or organizations. They also include dates, times, and money amounts. Named entity recognition is the process of denoting named entities in a particular text or corpus. Once this is done, a person can count and tabulate the entities to do things like determine the "aboutness" of a text, plot points on a map, or generate a timeline.

Exercise #4 - Explore meaning using dictionaries and POS analysis

- Explore this web-based interface (http://wordnet.princeton.edu)
- Use the command-line interface to explore WordNet (wn)
- Use Python's NLTK to explore WordNet (bin/define.py, bin/explain.py)
- Use the command line interface to parse a document for POS (tree-tagger-english)
- Use Python's NLTK to do POS analysis (bin/file2pos.py, bin/summarize-pos.py, bin/pos.py)
- Use Standford's NLP Java-based tool to do named entity recognition (bin/ner.sh)

Week 5 - Classification and clustering

It is often desirable to divide a large textual corpus into smaller subsets, and there are two approaches to the problem: 1) classification, and 2) clustering. Classification is similar to the process librarians use to organize books. A list of subject terms is articulated, and as new items are added to the collection they are associated with one or more of terms. Clustering is the complementary process allowing the person to first denote the number of “subject terms” (often called “topics”), and the computer program takes text as input doing its best to divide them into the given number of topics. Spam filters, authorship attribution (or "stylometrics"), and to some degree information retrieval processes all use classification and clustering techniques.

Exercise #5 - Classify and topic model documents

- Use Mallet to classify a set of documents (bin/classify.sh)
- Use Mallet to topic model a set of documents (https://code.google.com/p/topicmodelingtool/)

Links

[0] Workshop Materials / Course Site - http://dh.crc.nd.edu/sandbox/text-analysis-workshop/
[1] Natural Language Processing With Python - http://bit.ly/1yzbDBf
[2] Textbook Creative Commons License Detail - http://bit.ly/1AOEBMg
[3] Voyant Tools - http://voyant-tools.org/
[4] GATE - https://gate.ac.uk/

Eric Lease Morgan <emorgan@nd.edu>
February 26, 2015