Text Analysis with R, part 2

Michele Claibourn, UVA Library's StatLab

October 30, 2018

Roadmap for today

Today we'll focus on unsupervised approaches for analyzing text

- review of key concepts: document-feature matrix, reducing dimensionality
- review of implementation: dfms, frequencies, relative frequencies and weighting, keyness
- feature co-occurrence
- document similarity or distance
- document clustering – hierarchical, kmeans
- topic modeling - lda, stm

Still using Quanteda (mostly)

Corpora, Documents, Features

- Corpus: a library of documents along with corpus-levvel meta-data and document-level variables
- Document: unit of analysis, selected by researcher
- Feature: word/word stem, linguistic features (POS), multi-word expressions (ngrams), extracted metrics (e.g., readability)
- Document-feature matrix (dfm): a matrix of documents by tabulated features

Today's Text

1. Comments submitted to Ours to Shape
   - scraped with `acquire_comments.R` (in materials)
2. Tweets from @realDonaldTrump from January 20, 2017 to September 30, 2018
   - pulled from Brendan Brown's Trump Twitter Archive
   - as part of the broader Public Presidency Project

R Script Supplement | Similarity and Distance

With documents presented as vectors (e.g., vectors of word frequencies), measure the degree of similarity of two documents as the correlation between their corresponding vectors, for example, the cosine of the angle between the two vectors, or *cosine similarity*

$$cos(x_i, x_{ii'}) = \frac{x_{ij} * x_{i'j}}{|x_{ij}||x_{i'j}|} = \frac{\sum_{j=1}^{p} x_{ij}x_{i'j}}{\sqrt{\sum_{j=1}^{p} x_{ij}^2}\sqrt{\sum_{j=1}^{p} x_{i'j}^2}}$$

Or measure the distance between documents, for example the *Euclidean distance*

$$d(x_i, x_{i'}) = |x_i - x_{i'}| = \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2}$$

Distance measures are frequently used in clustering algorithms which also provide lots of choices.

R Script Supplement | Hierarchical Clustering

Produces hierarchical, or nested, groupings, but without pre-specifying number of groups, k

- Agglomerative: Begins with n partitions and successivly combinines observations
- Operates on a proxmity matrix and attempts to find the optimal next step
  - Start with each point in a cluster of its own
  - Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar. **Ward's method** joins the clusters that produce the minimim increase in the squared error. Fuse these
  - Compute the new pairwise inter-cluster dissimilarities among the $i - 1$ remaining clusters, then repeat above step until there is only one cluster
- Once two observations are fused they cannot be separated

R Script Supplement | Hierarchical Clustering, Dendrogram

The Dendrogram represents the sequence of fusions.

- The vertical height represents the distance between clusters when joined
- Earlier fusions occur lower in the tree and represent more similarity among groups/observations
- Don't get thrown by horizontal proximity
- Place a line at some height of the dendrogram; the number of vertical intersections equals the number of clusters.
- One way of choosing k – cut where the gap between combinations is largest.

R Script Supplement | K-means Clustering

Aka iterative partitioning. The general k-means algorithm is

1. Randomly pick *k*-centroids
2. Assign each data point to the nearest centroid
3. Update each centroid to be the average of the data points assigned to it
4. Repeat steps 2 and 3 until it stops changing

The initial centroids can affect the outcome, so try a lot of starting points.

R Script Supplement | K-means Clustering

Each centroid is calculated as the mean vector of all observations belonging to that cluster:

$$\bar{\mathbf{x}}_m = \frac{1}{n_m} \sum_{p=1}^{P} x_{ip}$$

And distance is the Euclidean distance from the mean vector:

$$WSS = \sum_{m=1}^{g} \sum_{l=1}^{n_m} (\mathbf{x}_{ml} - \bar{\mathbf{x}}_m)(\mathbf{x}_{ml} - \bar{\mathbf{x}}_m)'$$

Minimize the within sum of squares.

R Script Supplement | Topic Model, LDA

To aid in automatic discovery of thematic content: We observe documents and words; topics are part of latent (hidden) structure we wish to infer. Given a number of topics:

- Across a fixed vocabulary, each topic is a distribution over terms
- Each document is a distribution over topics
- To generate a document, randomly choose a distribution over topics. For each word in the document
  - Probabilistically draw one of the k topics from the distribution over topics
  - Given the topic, draw a word from the distribution over terms
  - Rinse and repeat

The model estimation "reverses" this stylized stochastic process to infer estimates of the originating topic and term distributions.

R Script Supplement | Topic Model, STM

The basic LDA assumes documents are exchangeable (independent). Several extensions are intended to relax this:

- Expressed Agenda Model (Grimmer 2010): allows for differences in topic probabilities across authors
- Dynamic Topic Model (Blei and Lafferty 2006): parameters change using an evolution model
- Correlated Topic Model (Blei and Lafferty 2007): Dirichlet prior is replaced with a logistic Normal distribution
- Structural Topic Model (Airoldi, Roberts, and Stewart 2011): for topic modeling with document-level covariate information
  - Topics can be correlated (logistic normal);
  - Each document has its own prior distribution over topics (β);
  - Word use within a topic can vary by group (κ)

R Script Supplement | Topic Model, Term metrics

- Highest probability terms:

$$\beta_{w,k} = exp(\mu_{w,k})$$

- Term exclusivity:

$$exclusivity_{w,k} = \frac{\mu_{w,k}}{\sum_{l=1}^{K} \mu_{w,l}}$$

- FREX: the harmonic mean of frequency and exclusivity

$$FREX_{w,k} = \left(\frac{\omega}{ECDF(exclusivity_{w,k})} + \frac{1-\omega}{ECDF(\mu_{w,k})} \right)^{-1}$$

- Lift: mean frequency for a term in topic k divided by the average frequency across all documents

Concluding Thoughts

1. All quantitative models of language are wrong, but some are useful
2. Validate, validate, validate