What should we hope to gain from visual analysis of text?

**Goals of Visual Text Analysis**

- **Understanding** - get the “gist” of documents
- **Grouping** - cluster for overview or classification
- **Compare** - compare document collections, or inspect evolution of collection over time
- **Correlate** - compare patterns in text to those in other data, e.g., correlate with social network

**Some Text Analysis Topics**

- Vector Space Model
- Descriptive Phrases
- Named Entity Recognition
- Clustering & Topic Modeling
Vector Space Model

Words are (not) nominal?
- High dimensional (10,000+)
- More than equality tests
- Words have meanings and relations
  - Correlations: Hong Kong, San Francisco, Bay Area
  - Order: April, February, January, June, March, May
  - Membership: Tennis, Running, Swimming, Hiking, Piano
  - Hierarchy, antonyms & synonyms, entities, ...

Text Processing Pipeline

Tokenization: segment text into terms
- Special cases? e.g., "San Francisco", "L'ensemble", "U.S.A."
- Remove stop words? e.g., "a", "an", "the", "to", "be"?

Stemming: one means of normalizing terms
- Reduce terms to their "root"; Porter's algorithm for English
  - e.g., automate(s), automatic, automation all map to automat
- For visualization, want to reverse stemming for labels
  - Simple solution: map from stem to the most frequent word

Result: ordered stream of terms

The Bag of Words Model

Ignore ordering relationships within the text
A document = vector of term weights
- Each dimension corresponds to a term (10,000+)
- Each value represents the relevance
  - For example, simple term counts

Aggregate into a document x term matrix
- Document vector space model
Document x Term matrix
Each document is a vector of term weights
Simplest weighting is to just count occurrences

<table>
<thead>
<tr>
<th></th>
<th>Antony and Cleopatra</th>
<th>Julius Caesar</th>
<th>The Tempest</th>
<th>Hamlet</th>
<th>Othello</th>
<th>Macbeth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antony</td>
<td>157</td>
<td>73</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Brutus</td>
<td>4</td>
<td>157</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Caesar</td>
<td>232</td>
<td>227</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Calpurnia</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cleopatra</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mercy</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>worser</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Keyword Weighting

Term Frequency
\[ tf_{id} = \text{count}(t) \text{ in } d \]

TF.IDF: Term Freq by Inverse Document Freq
\[
\begin{align*}
  tf_{id} &= \log(1 + tf_{id}) \times \log(N/df_i) \\
  df_i &= \# \text{ docs containing } t; \quad N = \# \text{ of docs}
\end{align*}
\]

G^2: Probability of different word frequency
\[
\begin{align*}
  E_1 &= |d| \times (tf_{id} + tf_{(C-d)}) / |C| \\
  E_2 &= |C-d| \times (tf_{id} + tf_{(C-d)}) / |C| \\
  G^2 &= 2 \times (tf_{id} \log(tf_{id}/E_1) + tf_{(C-d)} \log(tf_{(C-d)}/E_2))
\end{align*}
\]

Useful Libraries

Apache Lucene
Tokenization, stemming, indexing & search

Stanford Core NLP
Tokenization, stemming, POS tagging, entity recognition, parsing & coreference resolution
Descriptive Phrases

Limitations of Frequency Statistics?

Typically focus on unigrams (single terms)

Often favors frequent (TF) or rare (IDF) terms
  - Not clear that these provide best description

A “bag of words” ignores additional information
  - Grammar / part-of-speech
  - Position within document
  - Recognizable entities

How do people describe text?

We asked 69 subjects (all Ph.D. students) to read and describe dissertation abstracts.

Students were given 3 documents in sequence, they then described the collection as a whole.

Students were matched to both familiar and unfamiliar topics; topical diversity within a collection was varied systematically.

[Chuang, Manning & Heer, 2010]

Bigrams (phrases of 2 words) are the most common.
Phrase length declines with more docs & more diversity.

Term Commonness

\[
\frac{\log(tf_w)}{\log(tf_{the})}
\]

The normalized term frequency relative to the most frequent n-gram, e.g., the word “the”. Measured across an entire corpus or across the entire English language (using Google n-grams)

Selected descriptive terms have medium commonness. Judges avoid both rare and common words.

Commonness increases with more docs & more diversity.
Grammar: Technical Term Patterns

Technical Term \( T = (A|N)^* (N|C) / N \)

Compound Tech. Term \( X = (A/N)^* N \) of \( T \)

Regular expressions over part-of-speech tags.
\( A = \) adjective, \( N = \) noun, \( C = \) cardinal number.

Prior work suggests these patterns can be used to identify important terms in text.

**Over 4/5 of selected terms match pattern!**

Method - Part 2

**Build a statistical model of keyphrase quality**

Train a logistic regression model
Positive examples: selected phrases
Negative examples: randomly sampled phrases

Assess contributions of four classes of features:
Freq stats, commonness, grammar & position

Evaluate the phrases selected by our model using precision & recall measures
Adding Grammatical Features

Scoring Terms with Freq, Grammar & Position

<table>
<thead>
<tr>
<th>Model Feature</th>
<th>Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>constant</td>
<td>-2.3550***</td>
</tr>
<tr>
<td>log(tf)</td>
<td>0.9390***</td>
</tr>
<tr>
<td>WC ∈ (0%, 20%)</td>
<td>0.1770</td>
</tr>
<tr>
<td>WC ∈ (20%, 40%)</td>
<td>0.2304*</td>
</tr>
<tr>
<td>WC ∈ (40%, 60%)</td>
<td>0.0158</td>
</tr>
<tr>
<td>WC ∈ (60%, 80%)</td>
<td>-0.6205***</td>
</tr>
<tr>
<td>WC ∈ (80%, 100%)</td>
<td>-1.9081***</td>
</tr>
<tr>
<td>relative first occurrence</td>
<td>0.4800**</td>
</tr>
<tr>
<td>first sentence</td>
<td>0.9386***</td>
</tr>
<tr>
<td>full tech. term</td>
<td>-0.5015</td>
</tr>
<tr>
<td>partial tech. term</td>
<td>1.4461**</td>
</tr>
<tr>
<td>full compound tech. term</td>
<td>1.1373</td>
</tr>
<tr>
<td>partial compound tech. term</td>
<td>1.1806*</td>
</tr>
</tbody>
</table>

Fitted Parameters for a Corpus-Independent Model

WC: web commonness bins, *: p < 0.05, **: p < 0.01, ***: p < 0.001

Compare to User-Selected Phrases

At low recall (fewer phrases) our models select phrases with higher precision.
Automatic Keyphrase Extraction

Phase 1  Score candidate terms using our keyphrase quality regression model

Phase 2  Eliminate redundancy by grouping similar terms based on word overlap plus entity and acronym resolution.
- “analysis”, “data analysis”, ...
- “Barack Obama”, “Obama”, ...

\[ G^2 \]

Regression Model

<table>
<thead>
<tr>
<th>Term</th>
<th>Keyphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>fighter</td>
<td>Super Hornet</td>
</tr>
<tr>
<td>F/A</td>
<td>F/A-18</td>
</tr>
<tr>
<td>Hornet</td>
<td>fighter jet</td>
</tr>
<tr>
<td>Super</td>
<td>Boeing engineers</td>
</tr>
<tr>
<td>Boeing</td>
<td>special process</td>
</tr>
<tr>
<td>-18</td>
<td>rain check</td>
</tr>
<tr>
<td>rain</td>
<td>electronics suite</td>
</tr>
<tr>
<td>St.</td>
<td>Program experts</td>
</tr>
<tr>
<td>jet</td>
<td>simulated rain</td>
</tr>
<tr>
<td>Louis</td>
<td>ultimate customers</td>
</tr>
<tr>
<td>15-inch-per-hour</td>
<td>enormous hangar</td>
</tr>
<tr>
<td>close</td>
<td>water-tight integrity</td>
</tr>
<tr>
<td>hangar</td>
<td>Rich Barber</td>
</tr>
<tr>
<td>check</td>
<td>15-inch-per-hour rate</td>
</tr>
<tr>
<td>Check</td>
<td>video story</td>
</tr>
<tr>
<td>Barber</td>
<td>aircraft</td>
</tr>
<tr>
<td>van</td>
<td>U.S. Navy fighter pilots</td>
</tr>
<tr>
<td>seaplane</td>
<td>Super Hornet final assembly manager</td>
</tr>
</tbody>
</table>

Named Entity Recognition
Named Entity Recognition

Input:
Jim bought 300 shares of ACME Corp. in 2006.

Output:
<PERSON>Jim</PERSON> bought <QUANTITY>300</QUANTITY> shares of <ORG>ACME Corp.</ORG> in <YEAR>2006</YEAR>.

Current state-of-the-art uses statistical methods: conditional random fields (CRF).

Entity Recognition Uses/Issues

Extract structured data from unstructured text.
Analyze documents w.r.t. constituent entities.
>> subject of lecture on Jigsaw (see also Palantir)

Challenges
- Entity resolution (which entities are the same?)
- Entity linkage (how are entities connected? Can one label the relationship types?)
- Most analysis tools require human-in-the-loop

Clustering / Topic Modeling
Cluster & Compare Documents

STRATEGY 1: Cluster term vectors directly
- K-means clustering of document vectors
- Hierarchical agglomerative clustering

STRATEGY 2: Map documents to lower-dim space
- Latent Semantic Analysis (LSA)
- Statistical Methods (e.g., Latent Dirichlet Allocation or LDA)
- Use topics as clusters or compare in new space

Latent Semantic Analysis (LSA)

IDEA: Construct a low-rank approximation of the document-term matrix. Map the individual terms (10,000+) to a smaller set of dimensions (typically 100-300) representing “concepts.”

Roughly, the idea is to find the eigenvectors and use the top-K vectors to represent the data. BUT, doc-term matrices aren’t square. Thus a different way to find principal components is used: singular value decomposition (SVD).

Overlap of articles in categories Biochemistry (blue) and Medicine (red).

Statistical Topic Modeling

Discover latent topics in a collection by finding words that co-occur in the same documents.

Popular model: Latent Dirichlet Allocation (LDA)

Assumes documents are mixtures of topics, and the words are drawn from these topics.

Given K (# topics) and a set of documents D with words from a vocabulary V, determine:

\[ \beta_k \sim \text{Multinomial}(V) \] - topics as dist. over words

\[ \theta_d \sim \text{Multinomial}(K) \] - per-doc topic mixtures
Example LDA Output

<table>
<thead>
<tr>
<th>Topics</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physics</td>
<td>optical, quantum, frequency, laser, high,</td>
</tr>
<tr>
<td></td>
<td>electron, single</td>
</tr>
<tr>
<td>Biology</td>
<td>dna, replication, rna, repair, complex,</td>
</tr>
<tr>
<td></td>
<td>interaction, base</td>
</tr>
<tr>
<td>Generic</td>
<td>results, show, finally, present, provide,</td>
</tr>
<tr>
<td></td>
<td>demonstrate</td>
</tr>
</tbody>
</table>

Topic Identification

LDA is an example of a latent topic model: topics are inferred from the data. The learned topics have no names: each is exactly and only defined by its distribution over words.

What if we already have label metadata?

Labeled LDA: assume each label is a topic, and infer the probability of the words associated with each label.

Document Similarity

Compute distance between document vectors
Common: cosine of angle b/w term vectors (just the dot product if vectors are normalized)
Similarity affected by term weighting (e.g., tfidf)

With LSA or LDA models, can compute distance in lower dimensional space, e.g., \( \cos(\theta_d, \theta_{d'}) \)

Example: Thesis Explorer
MIMIR / Jason Chuang

9,000+ Thesis Abstracts

TF.IDF Cosine Distance

LDA Topic Distance

"Word-Borrowing" among Departments (similar to Labeled LDA)
LDA Overclusters the Humanities?

Implications for Vis Design

Standard InfoVis Reference Model:
Input -> Data -> Visualization <-> Analyst

Model-Driven Visualization?
Input -> Data -> Model -> Visualization <-> Analyst

What considerations guide the visualization process for increasingly abstracted data?

Interpretation & Trust

Interpretation: the facility with which an analyst makes inferences about the underlying data, which is affected by all stages in the process (choice of model, choice of visual encodings...)

Trust: the actual and perceived validity of an analysts’s inferences. Are the interpretations valid and actionable?

Strategy 1: Alignment

Understand the domain and organize the display around the primary entities of interest. Define the model accordingly. Anticipate sources of error to reduce false inferences.

In the Thesis Explorer this led to:
- Focus on similarity between departments
- Choice of word-borrowing model
- Abandonment of PCA projection overview
Strategy 2: Progressive Disclosure

Enable analysts to shift among levels of abstraction on-demand, by drilling down from overview, to intermediate abstractions, and eventually to the underlying data itself.

In the Thesis Explorer this led to:
- Department level view
- Thesis level drill-down view, access to text
- Needed future work to assess word overlap

Strategy 3: Social Verification (?)

Observe and record domain experts’ met and unmet expectations as they explore the data. We used this information to drive modeling decisions for the thesis explorer.

In the future, visualizations might enable a larger, crowd-sourced verification of the model’s ability to quantify and discover facts known to distinct subsets of a community.