Using Text Mining Techniques to Help Bring Electronic Discovery Under Control
HELLO, MY NAME IS...

Bruce

---ASK ME ABOUT CLUSTERING
1. Background of eDiscovery environment

2. Search analytics through result side clustering

3. Unsupervised feature extraction through NMF

4. Catalyst clustering engine

5. Shortcuts we have taken

Agenda
Working in eDiscovery
Time is an elusive commodity
Documents are plentiful and messy
(and mostly electronic and multi-language unlike this photo)
Pressure is stressful

http://www.americanairworks.com/images/dial_a_pressure.gif
Gamesmanship is always in play
Deadlines are real and costly
• Time is not a lawyer's friend
• Documents are numerous
• Pressure is high
• Gamesmanship
• Real deadlines
Search Analytics: Result Side Clustering
Getting Information Without Drowning
How to find the key concepts?

http://www.geronto.at/Links/hauptteil_links.html
documents + vectorizer → "doc vector" or DV
• Vectorizer counts term frequency over phrases (n=2 to 6)

• Boosts can be applied for proper nouns, dictionary validation, entity validation, and customer dictionary validation
This is the vector summary of the document.
The basic theory that lays behind this implementation...

\[ w(s, d) = [\text{TF}(s, d)]^{\alpha} \times [\text{IDF}(s, D)]^{\beta} \]

where \( w(s, d) \) is the tf-idf weight of a term \( s \) in a document \( d \).

Note that the document is represented by an array of tf-weighted terms, normalized between 0..1. The D is the corpus of all terms -- all terms in the result set. Alpha and beta are configurable constants to skew in favor of TF or IDF. Salt to taste.
RSC in Product

<table>
<thead>
<tr>
<th>Key Concepts Tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>This page allows you to tune how the Key Concepts are returned.</td>
</tr>
</tbody>
</table>

Would you like to favor terms that appear **more** frequently? Scores less than 1 will decrease the importance of frequently occurring terms. Scores greater than 1 will increase the importance of frequently occurring terms.

Would you like to favor terms that appear **less** frequently? Scores less than 1 will decrease the importance of infrequent terms. Scores greater than 1 will increase the importance of infrequent terms.

- **I'd like to leave this alone. Take the default (1.0)**
- **Decrease the importance (0.5)**
- **Completely ignore infrequent terms (0)**
Feature Extraction using Non-Negative Matrix Factorization
This may be review for many of you...

"(They'll) want to know what's hot in machine learning (nonnegative matrix factorization is so 1998! Latent Dirichlet Allocation is all the rage now)" - Michael C. Mozer
The collection of documents
We try to approximate this $n+1$
All documents → Reduction & Sampling → N-Grams

- DV DV DV
- DV DV

- netscape
- client products
- aol agreement
- licensed products
- section
- aol affiliates
- standard client
- client product
<table>
<thead>
<tr>
<th>N-Grams</th>
<th>netscape</th>
<th>client products</th>
<th>aol</th>
<th>agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>33</td>
<td>44</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td>DV</td>
<td>33</td>
<td>34</td>
<td>44</td>
<td>54</td>
</tr>
<tr>
<td>DV</td>
<td>44</td>
<td>110</td>
<td>38</td>
<td>23</td>
</tr>
<tr>
<td>DV</td>
<td>42</td>
<td>38</td>
<td>61</td>
<td>51</td>
</tr>
<tr>
<td>DV</td>
<td>37</td>
<td>23</td>
<td>51</td>
<td>91</td>
</tr>
<tr>
<td>DV</td>
<td>18</td>
<td>45</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>
Math is the new black

(photo courtesy of pioneering work with Wolfram|Alpha)
Factorization converges on a set of features.

Matrix (N-Gram freq)

<table>
<thead>
<tr>
<th></th>
<th>netscape</th>
<th>client products</th>
<th>aol</th>
<th>agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>33</td>
<td>44</td>
<td>42</td>
<td>37</td>
</tr>
<tr>
<td>DV</td>
<td>33</td>
<td>34</td>
<td>44</td>
<td>54</td>
</tr>
<tr>
<td>DV</td>
<td>44</td>
<td>110</td>
<td>38</td>
<td>23</td>
</tr>
<tr>
<td>DV</td>
<td>42</td>
<td>38</td>
<td>61</td>
<td>51</td>
</tr>
<tr>
<td>DV</td>
<td>37</td>
<td>23</td>
<td>51</td>
<td>91</td>
</tr>
<tr>
<td>DV</td>
<td>18</td>
<td>45</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Features

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DV</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>DV</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>DV</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>DV</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>DV</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
We call these features ‘Centroids’

<table>
<thead>
<tr>
<th>Features</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>aol</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>agreement</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>licensed</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>section</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>aol affiliates</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
What is a centroid?

Each centroid will be used to ‘attract’ documents into a cluster

$$C_i = \text{aol browser search james clark}$$

$$\text{navigator netscape} \quad \text{netscape}$$

$$\text{navigator time warner}$$

A list of weighted N-Grams

Each centroid will be used to ‘attract’ documents into a cluster
Actual Centroids
Catalyst Clustering Engine aka Putting Centroids to Work
• The “engine” reference is simply a metaphor

• Catalyst cannot provide any mechanical work for your automobile, motorcycle, lawn mower, or scooter

Hyperbole Alert
• Search derived
• Highly structured query
• Custom rank profiles
• Overlap and tiebreaker resolution
• Optional hand tuning and refinement
• Any external tool can define a centroid

What Is the Clustering Engine?
Centroids attract documents
Documents and centroids

Note that the centroids are not really part of the collection
n+1
(no centroid, therefore no affinity)
One of these things is not like the other: $n+1$ cluster
• Not all documents will match a centroid

• Clustering engine allows scoring threshold to determine membership

• Put all the noise into one cluster

• Multiple NMF processes can be run against different filtered sets

N+1 Cluster
5 Shortcuts
An Exploratory Analysis of Phrases in Text Retrieval, Jeremy Pickens and W. Bruce Croft

A Formal Derivation of Heaps’ Law, D.C. van Leijenhorst, Th.P. van der Weide

Algorithms for Non-negative Matrix Factorization, Daniel D. Lee, H. Sebastian Seung

Andrey A Puretskiy
Dr Michael W Berry
Validation was quantitative
Acceptance was qualitative
Easy to quantify different,
hard to quantify better
• Vectorizer (long documents, short documents, OCR)
• Factorizer loop count (diminishing returns, diff-cost may be local minima)
• Feature limit (many clusters may be too confusing)
• N+1 (don't aim for noisy completeness)
• Query expression can determine corpus (custodian=jct)
numbers@catalystsecure.com