Introducing nora: A Text-mining Tool for Literary Scholars

Tom Horton
Dept. of Computer Science
University of Virginia
horton@cs.virginia.edu
Talk Outline

• About the nora project
• Text mining and literary research
  – Process of text-mining and analysis
  – Text-mining outputs and their use
  – Examples of results
• Software for end-users
• Software architecture
• Issues and lessons learned
nora Project Goals

• Develop tools that
  – solve problems of interest to literary scholars
  – making use of existing digital library resources

• Text-mining (TM):
  – Develop tools and an architecture to allow non-specialists to use TM
  – provocational text-mining to support literary interpretation
About the nora project

- http://www.noraproject.org
- Funded by the Andrew W Mellon Foundation
- Multidisciplinary participants from five universities
  - Illinois at Urbana-Champaign, Alberta, Georgia, Maryland, Virginia
  - Led by John Unsworth
- Areas of activity:
  - Technical: text-mining, SW architecture, visualization, user studies, interface design
  - Literary: Emily Dickinson; sentimentalism; Virginia Woolfe; vocabulary of literary criticism...
nora Participants

• Univ. of Illinois at Urbana-Champaign
  – Director: John Unsworth
  – Data mining: Bei Yu, Loretta Auvril
  – Software: Xin Xiang, Amit Kumar

• Univ. of Maryland
  – Literary: Matt Kirschenbaum, Martha Nell Smith, Tanya Clement, …
  – Software and usability: Catherine Plaisant, James Rose
nora Participants

• Univ. of Georgia
  – Software and literary: Steve Ramsay, Sara Steger

• Univ. of Alberta
  – Interface design: Stan Rueker and team

• Univ. of Virginia
  – Literary: Kristen Taylor and others
  – Data mining and software: Tom Horton
Part 1:

• Text-mining for literary research
  – Example: Sentimentalism
  – Other examples:
    • Eroticism in Emily Dickinson
    • Vocabulary in papers on literary criticism
Some Project Assumptions

• Users: literary scholars
  – Interested in exploration, provocation
  – Not in: decision making, quantification
    (perhaps including corpus characteristics)

• Data sources: existing digital libraries
  – Not under user’s direct control
  – Can’t modify them
  – nora applications to be eventually deployed with at a DL’s site
Example: nora’s Sentimentalism Study

• Apply nora ideas to a set of 19th century novels in the Early American Fiction digital library

• Help scholars better understand sentimentalism in a core set of highly sentimental novels

• Identify seemingly sentimental parts of other documents
  - help prove the usefulness of TM in literary criticism
What is Sentimentalism?

- Term “sentimental novel” first applied to 18th century texts
  - Feeling is valued over reason
  - Author attempts to induce a specific response from the reader
    - Often for a cause: anti-slavery, female education, temperance, etc.
  - Conventional plot devices, characters, repetitions
  - Explicit authorial interventions
Why It’s an Interesting Problem

- Some novels were hugely popular in the US
- Many novels written by women
- Social issues: e.g. slavery
- Solidification of novel form, and predecessor to Victorian period
- Often used as a derogatory term
  - both then and now
  - but increased recent interest
Text-Mining for Such Problems

• Data-mining on documents
  - So far: Data (“features”) are vocabulary-based
  - Our first analyses do not use POS, parsing, etc.

• Possible goals:
  - Classification: From a small set of “known” results, make predictions about “unknown” results
    • Explanation?
  - Clustering: Group or organize unknown results based on non-obvious similarities
Our Process using TM

1. Choose a training-set of novels
2. Scholars assign a numeric score indicating degree of sentimentality for each chapter
3. Run a particular text-mining algorithm
   - Using the set of chapters with their scores to create a classification model
4. Evaluate text-mining outputs
   - from a TM perspective
   - from a literary perspective by applying traditional scholarship using TM results as a starting point
Text-Mining Outputs

1. Measures of whether a model can be built that successfully classifies the training-set
   - For the set of chapters, how often does the TM classification result match the scholar’s assignment?

2. A numeric score indicating the degree that a chapter seems sentimental (or not)
   - What’s most sentimental? Least? What’s the pattern?

3. Predictors: vocabulary ordered to show which words contribute most or least to assigning each chapter
   - Possibly a form of explanation for the scholar
Keep In Mind:

• Our use of TM is for:
  – Provocation, exploration

• We don’t assume or propose a particular “ground truth”
  – Scholars are free to assign their own scores for what is and isn’t sentimental
    • Our software tools will allow iteration and exploration
  – Prediction results are to serve as starting point for close-reading and analysis
    • Show me “more like these”
  – Predictors may or may not lead to satisfying explanation
Sentimental Experiment Plan

• Experiment 1:
  - Goal: To evaluate the use of text-mining on a small set of "core" sentimental novels.
  - Scholars assign a score or label for each chapter in five novels
  - Run text-mining and see what we learn about the methods and the novels
Experiments To Come

• Experiment 2:
  - More sentimental novels in the TM test-set (i.e. not scored initially)
  - Evaluate prediction:
    • Use the TM model from initial training-set of novels
    • Predict which chapters from test-set are most and least sentimental, and explore those novels
    • Examine if predictors from training-set generalize to new documents

• Experiment 3:
  - Apply to works perceived to not be sentimental
Scoring Test-Set Chapters

• Scores assigned by graduate students from the English department
  – Two scorers per chapter
  – Results averaged. Disagreements reconciled.

• Scored initially on scale of 1 to 10
  – Converted to High/Medium/Low
  – Eventually TM run as a two-class problem: High vs. Medium/Low
Reminder: Text-Mining Outputs

1. Measures of whether a model can be built that successfully classifies the training-set
   - For the set of chapters, how often does the TM classification result match the scholar’s assignment?

2. A numeric score indicating the degree that a chapter seems sentimental (or not)
   - What’s most sentimental? Least? What’s the pattern?

3. Predictors: vocabulary ordered to show which words contribute most or least to assigning each chapter
   - Possibly a form of explanation for the scholar
Results: Classification Accuracy

• From a TM perspective
  – Classification model not as successful as we’d like when re-classifying test-set chapters
  – A concern?
    • We’re exploring why and looking for better methods
      – Proper nouns, part of speech, SVM vs. Naïve Bayes
    • But we still believe this is a useful starting point for literary analysis
Results: Classification of Chapters

• Using the Naïve Bayes DM method
  – Each chapter gets a score
  – Positive means not highly-sentimental
  – Negative means highly-sentimental
  – How far from zero can be interpreted as a relative degree as calculated by the TM algorithm
    • Recall our scholars’ scores were used as yes/no
Change During a Novel

• Stowe’s two novels show more by-chapter variation than Rowson’s works
  – *UTC* has fluctuation between highly-sentimental episodes with scenes of minstrelsy or humor
  – *The Minister’s Wooing* shares this flow (though about marriage)
• Reminder: negative means more sentimental
Vocabulary Predictors

• Recall the NB method for TM ranks words by how strongly they indicate sentimental or not-sentimental

• Highly-sentimental words include proper names
  – Makes sense: particular characters appear in highly sentimental chapters
  – Won’t lead to models that generalize well for new novels
  – A solution: use part-of-speech tagging to ignore proper-nouns for TM
Predictors: What do They Tell Us?

• The list of words and how strongly they indicate sentimentality (or not)
  - Can they “explain” results in a way that interests or informs a literary scholar?

• The verdict:
  - Maybe! (Not clear yet.)
  - Close vocabulary study needed.
  - SVM vs. Naïve Bayes TM methods
  - Part of speech tagging, stop words
Restricting TM to Certain POS

- 8453 word-types when restricted by POS:
  - Nouns, adjectives, adverbs (no proper nouns)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Feature</th>
<th>Rank</th>
<th>Feature</th>
<th>Rank</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>senator</td>
<td>18</td>
<td>pitying</td>
<td>35</td>
<td>clairvoyant</td>
</tr>
<tr>
<td>2</td>
<td>measured</td>
<td>19</td>
<td>vow</td>
<td>36</td>
<td>intently</td>
</tr>
<tr>
<td>3</td>
<td>paternal</td>
<td>20</td>
<td>toilette</td>
<td>37</td>
<td>suspense</td>
</tr>
<tr>
<td>4</td>
<td>auctioneer</td>
<td>21</td>
<td>prayer-meeting</td>
<td>38</td>
<td>kneeling</td>
</tr>
<tr>
<td>5</td>
<td>payment</td>
<td>22</td>
<td>Spirit</td>
<td>39</td>
<td>writer</td>
</tr>
<tr>
<td>6</td>
<td>incidents</td>
<td>23</td>
<td>impulsive</td>
<td>40</td>
<td>beloved</td>
</tr>
<tr>
<td>7</td>
<td>shrink</td>
<td>24</td>
<td>pearls</td>
<td>41</td>
<td>necessities</td>
</tr>
<tr>
<td>8</td>
<td>storms</td>
<td>25</td>
<td>painfully</td>
<td>42</td>
<td>alabaster</td>
</tr>
<tr>
<td>9</td>
<td>weaker</td>
<td>26</td>
<td>hesitating</td>
<td>43</td>
<td>renew</td>
</tr>
<tr>
<td>10</td>
<td>spared</td>
<td>27</td>
<td>agency</td>
<td>44</td>
<td>straits</td>
</tr>
<tr>
<td>11</td>
<td>reverie</td>
<td>28</td>
<td>violating</td>
<td>45</td>
<td>overpowerd</td>
</tr>
<tr>
<td>12</td>
<td>aged</td>
<td>29</td>
<td>wilderness</td>
<td>46</td>
<td>build</td>
</tr>
<tr>
<td>13</td>
<td>anguish</td>
<td>30</td>
<td>mon</td>
<td>47</td>
<td>radiance</td>
</tr>
<tr>
<td>14</td>
<td>nest</td>
<td>31</td>
<td>intensity</td>
<td>48</td>
<td>giveth</td>
</tr>
<tr>
<td>15</td>
<td>infamy</td>
<td>32</td>
<td>enfant</td>
<td>49</td>
<td>drooped</td>
</tr>
<tr>
<td>16</td>
<td>retained</td>
<td>33</td>
<td>exchanged</td>
<td>50</td>
<td>pain</td>
</tr>
<tr>
<td>17</td>
<td>neatness</td>
<td>34</td>
<td>hence</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part 2: SW Apps and Architecture

• The noravis application developed at the Univ. of Maryland (for Dickinson study)
  - Support scholars with minimal knowledge of text-mining
  - Allow them to label or score documents, then run text-mining classification
    • And repeat this process iteratively
  - See classification of un-labeled documents
  - See significant vocabulary features
  - Read documents
## Will Literary Scholars Read This?

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Ratio</th>
<th>ID_Label</th>
<th>Pred.</th>
<th>Wrong?</th>
</tr>
</thead>
<tbody>
<tr>
<td>-17394.99615</td>
<td>-289.385218</td>
<td>Jeaf490_2</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-9135.329147</td>
<td>-172.7576762</td>
<td>eaf325_7</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-8380.789049</td>
<td>20.29566524</td>
<td>eaf325_8</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-8958.350474</td>
<td>-267.7518636</td>
<td>eaf325_22</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-23085.09238</td>
<td>-347.0904289</td>
<td>Jeaf490_18</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-46034.85412</td>
<td>1067.896704</td>
<td>Seaf709_20</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-11387.55962</td>
<td>-239.1338307</td>
<td>Jeaf490_23</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-15109.89116</td>
<td>-301.4195653</td>
<td>Jeaf490_28</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-14046.51397</td>
<td>75.99192534</td>
<td>Jeaf490_31</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-6568.900176</td>
<td>-175.9008935</td>
<td>eaf325_11</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-32490.81877</td>
<td>595.6814592</td>
<td>Seaf709_1</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-8441.71613</td>
<td>1.754756634</td>
<td>eaf325_19</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-7857.847789</td>
<td>-209.6624077</td>
<td>eaf325_26</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-21677.41616</td>
<td>223.639316</td>
<td>Seaf709_22</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-6630.692657</td>
<td>83.62923777</td>
<td>Jeaf490_22</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-23466.78489</td>
<td>-204.1043602</td>
<td>Seaf709_27</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>-23858.20899</td>
<td>258.1451311</td>
<td>Jeaf490_13</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>-11320.8584</td>
<td>-33.62762601</td>
<td>eaf325_4</td>
<td>M</td>
<td>H</td>
</tr>
</tbody>
</table>
Noravis User Interface
Software Architecture

• D2K and T2K
  – Data mining tools and environment from the NCSA (at UIUC)
  – http://alg.ncsa.uiuc.edu
  – A data-mining “engine” plus…
  – Modules
  – Itineraries
  – Web-services component
**D2K and Its Many Components**

- **D2K Infrastructure**
  
  *D2K API, data flow environment, distributed computing framework and runtime system*

- **D2K Modules**
  
  *Computational units written in Java that follow the D2K API*

- **D2K Itineraries**
  
  *Modules that are connected to form an application*

- **D2K Toolkit**
  
  *User interface for specification of itineraries and execution that provides the rapid application development environment*

- **D2K-Driven Applications**
  
  *Applications that use D2K modules, but do not need to run in the D2K Toolkit*
Part 3: Issues
Literary Docs and TM

• A TM Assumption: large amount of data overcomes “noise”, lack of precision

• TM is often about: news, emails
  – Lots of short documents

• Literary documents
  – Novels: big but few
    • Process by chapter, page,…
  – Often scholars want to focus on a small subset
Logical Units within Documents

• “Chunking”
• Need frequency counts by chunk
• What’s available for each document?
  – Processing and user-choice
• Document collection issues
  – Different mark-up between documents
  – Logical equivalence: treat *sections* in Doc1 like *chapters* in the other docs
Document Processing

• Excluding parts of documents
  – Just XML <DIV1> elements with <BODY>
  – Ignore <FIGURE>

• Documents that faithfully reproduce a old publication
  – “Missing” or “duplicate” chapters
  – Spellings

• Varying levels of markup
Final Remarks

• TM results are interesting to literary scholars
• “Doing TM well” for our documents is an on-going exploration
  – Need to collaborate with other communities
• Easy-to-use interfaces matter for our users
• Integration with word-lists, KWIC, etc. matter