9. Text & Documents
Visualizing and Searching Documents

Lecture „Informationsvisualisierung“
Prof. Dr. Andreas Butz, WS 2012/13
Concept and slides: Thorsten Büring,
3rd, revised edition
Outline

• Characteristics of text data
• Detecting patterns
  – SeeSoft
  – Arc diagrams
  – Visualizing Plagiarism
• Keyword search
  – TextArc
  – Enhanced scrollbar
  – TileBars
• Cluster Maps
  – Visualization for the document space
  – WEBSOM
  – ThemeScapes
• Cluster map vs keyword search
Text & Documents

- The main vehicle for humans to store information
- Huge existing resources: libraries, WWW
- What to visualize?
- Text is a nominal data type, but with many additional and interesting properties
- Text structure
- Meta data
  - Author
  - Dates
  - Descriptions
- Relations between documents (e.g. citation, similarity)
- Relevance of documents to a query
- Text statistics (e.g., frequency of different words)
- Content / Semantics
PaperLens
SeeSoft

- Eick et al. 1993
- Software visualization tool to display code line statistics (e.g., age, programmer, number of executions in recent test, etc.)
- Encoding
  - Each column represents a file
  - Height of column: length of the document
  - Files exceeding the height of the screen are continued over to the next columns
  - Each row represents a line of code
  - Width of row: length of line
  - Color: age of the line (red: newest; blue: oldest)
- Scales up to 50,000 lines on a single screen
- Example: 20 files with 9,365 lines of code
- Reading windows controlled by virtual magnifying boxes
SeeSoft

Seechanges:
Using SeeSoft To Visualize Program Change History

By Joe Steffen and Steve Eick

9-14-92
Arc Diagrams

- Wattenberg 2002
- Visualizes repetition in string data
- Application domains: text, DNA sequences, music
- Approach: to avoid clutter, only visualize an essential subset of all possible pairs of matching substrings
- Display string on a single line
- Connect the consecutive intervals by a semi-circular arc
  - Thickness of the arc: length of the matching substring
  - Height of the arc: proportional to the distance of substrings
Arc Diagrams

- Apply translucency to not obscure matches
- Still: for strings with a high frequency of small repeated substrings the visualization may cause clutter
- Provide users with the ability to filter by minimum substring length to consider
Arc Diagrams

- Comparison to a dotplot diagram
- Recap Matrix diagram
  - Correlation matrix
  - String of n symbols $a_1, a_2, \ldots, a_n$ is represented by an $n \times n$ matrix
  - Pixel at coordinate $(i, j)$ is black if $a_i = a_j$
  - Can handle very large datasets
  - Shows both small and large-scale structures

- Heavy clutter caused by small substrings with high frequency: $n$ repetitions of a substring lead to $n^2$ visual marks
- Arc Diagrams mark only similar substrings, which are subsequent
Arc Diagrams

• Applied to music, Minuet in G Major, Bach
• Shows classic pattern of a minuet: two main parts, each consisting of a long passage played twice
• Parts are loosely related: bundle of thin arcs connecting the two main parts
• Overlap of the two main arcs shows that the end of the first passage is the same as the beginning of the second passage
Visualizing Plagiarism

• Ribler & Abrams 2000
• Problem: programming assignment in a class with large number of students
• High probability of plagiarism
• Need to compare every document (code file) with every other document
• Visualization must support two steps
  – Highlight suspicious documents
  – Allow for detailed examination of the similar passages - high level of similarity between documents may not be due to cheating (e.g., headers)
Visualizing Plagiarism

- Categorical Patterngram
- Visualize frequencies of sequences of characters present in more than one document
- Remove all non-printable characters in the document collection
- Define length of character sequence to analyse (in the example: 4)
- Histogram-like approach
  - X-axis: start character of sequence
  - Y-axis: number of documents containing the sequence
  - Doc at Y = 1: base document to compare against all other documents
Visualizing Plagiarism

• Composite Categorical Patterngram
• Visualizes which particular documents are similar
• Y-axis: each value corresponds to an individual document
Visualizing Plagiarism

- Case study
- Students were asked to extend a sample program of about 30 lines of code
- Average completed program was about 150 lines
- Submission via email
- Graphic shows categorical patterngram for a single submission
  - Sequence length = 10
  - Lines not text due to high density
  - Rather confusing color coding
- Color coding (not very reasonable)
  - Green: frequency \( \geq 10 \)
  - Red: frequency \(< 10\)
  - Blue: base document
- Plagiarism or not?
Visualizing Plagiarism

• What to look out for?
  – Sequences that occur frequently are not of interest - all points with $y \geq 10$ are plotted as $y = 10$
  – Suspicious: accumulation of points with low frequencies

• Analysis
• Majority of points are plotted at $Y = 1$
• Hence most 10-char sequences are unique to the base document
• Number of points plotted at $Y = 2$, but evenly distributed
Visualizing Plagiarism

• Composite Categorical Patterngram for the submission
• Solid line represents the base document (submission number 23)
• Large number of points plotted in the range of $x = [0; 500]$: email message header
• Other frequent sequences due to the sample program
• Pattern typical for independent work
Visualizing Plagiarism

- Example of patterngrams indicating extensive plagiarism
Visualizing Plagiarism

- Patterngram of more subtle plagiarism
Visualizing Plagiarism

• What may a student do to mask plagiarized code
  – Change variable names
  – Minimize masking effect by replacing all alphanumeric strings in all documents into single characters

• Two documents with the same code but different variable names will produce similar patterngrams
A more recent example [guttenplag.wikia.com]

1218 Plagiatsfragmente aus 135 Quellen
auf 371 von 393 Seiten (94.4%)
in 10421 plagiierten Zeilen (63.8%)

Stand: 03.04.2011 11:55 Uhr

- Seiten, auf denen Plagiate gefunden wurden
- Seiten mit Plagiaten aus mehreren Quellen
- Seiten, auf denen bisher keine Plagiate gefunden wurden
- Das Inhaltsverzeichnis (Seiten 1-14) und die Anhänge (ab Seite 408) wurden nicht bei der Berechnung des Prozentsatzwertes mit einbezogen
TextArc

- http://www.textarc.org/ - demo
- Represents the entire text as 1 pixel lines in an outer circle
- Text is revealed via mouse-over
- Words are repeated in inner circle at a readable size
- Position of the words depend on where the word appears in the document
- Words that appear throughout the novel will be drawn to the center
- Frequent words stand out
- Example visualizes the novel “Alice in Wonderland”
- Various visualization features
Search Terms on a Scrollbar

- Byrd 1999
- Searching of keywords in a single document
- Color coding to map each occurrence of a keyword in the document as a small colored icon in the scrollbar
- Provides an overview of the entire document, not only of the portion currently visible
- Users can directly jump to keyword occurrences by moving the slider thumb
TileBars

• Hearst 1995

• Problem with document ranking of common search engines?

• Ranking approach is opaque:
  – What role did the query terms play in the ranking process
  – What is the relationship between the query terms in the document

• TileBars attempts to let the users make informed decisions about which documents and passages to view
TileBars

- Users provide sets of query terms
  - OR within a set
  - AND between sets
- Documents are partitioned into adjacent, non-overlapping multi-paragraph segments
- Each document of the result set is represented by a rectangle - width indicates relative length of the document
- Stacked squares correspond to text segments
- Each row of the stack corresponds to a set of query terms
- Darkness of the square indicates the frequency of terms from the corresponding term set - (Why is this a reasonable color mapping?)
- Title + initial words appear next to each document
- Users can click on segments to retrieve the corresponding text
**TileBars**

- **Analysis hints**
  - Overall darkness indicates that all term sets are discussed in detail throughout the document
  - When terms are discussed simultaneously the tiles blend together causing an easy to spot block
  - Scattered term set occurrence show large areas of white space
  - Helps to distinguish between passing remarks and prominent topic terms

- **Users may also set distribution constraints to refine the query**
  - Minimum number of hits per term set
  - Minimum distribution (percentage of tiles containing at least one hit)
  - Minimum adjacent overlap span
Cluster Maps

• Downscaling of n-dimensional document space to 2D
• Map of a document collection
• Similar documents are placed close to each other
• Dissimilar documents are placed farther apart from each other
• Provide thematic overview for exploration (same concept as product arrangements in a store)

• How to - Vector space model and map construction
  – Create inverted index of document collection
  – Exclude stop words and the most frequent words (“and” may not be a good discriminator of content)
  – Matrix of indexing words versus documents gives you document vectors
  – A document vector reflects the frequency of index words occurring in the document
Cluster Maps

• How to - Vector space model and map construction (continued)
  – Compute similarity between pairs of documents (e.g. dot product of vectors)
  – Layout documents in 1D/2D/3D

• Common approaches
  – Spring model of graph layout
  – Multi-dimensional scaling
  – Clustering (e.g. hierarchical)
  – Self-organizing maps (SOM aka Kohonen map)

Document vectors

<table>
<thead>
<tr>
<th></th>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Doc 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Artificial”</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>“Creativity”</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>“Java”</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Similarity Matrix

<table>
<thead>
<tr>
<th></th>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Doc 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doc 1</td>
<td>1</td>
<td>0.66</td>
<td>0</td>
</tr>
<tr>
<td>Doc 2</td>
<td>0.66</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Doc 3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
SOM

- Unsupervised learning algorithm
- SOM map is formed from a regular grid of neurons (nodes)
- Each node has
  - An x y coordinate in the grid
  - A weight vector of the same dimensionality as the input vectors

- Input vectors
  - Used to train the map
  - Represent collection of objects

- In case of visualizing text, input vectors are usually equal to document vectors
SOM - Algorithm

1. Start with assigning small random weights to the nodes of the grid
2. Choose a vector at random from the set of input vectors and present it to the grid
3. For each node: calculate the Euclidean distance between each node's weight vector and the current input vector - the closest node is called the Best Matching Unit (BMU)
4. Calculate the radius of the BMU (radius diminishes with each time-step)
5. For each node within the radius of the BMU: adjust the weights to make them more similar to the input vector - the closer a node is to the BMU, the more its weights get altered
6. Repeat step 2 for N iterations

When training is completed each document is assigned to its BMU
Skupin
2004
Cluster Maps

- Lin 1992
- Personal collection of 660 research documents
- 2500 learning iterations
- Labeled word show most frequent title words
- Size maps to frequencies of occurrence of the words
- Neighboring relationships of areas indicate frequencies of the co-occurrence of words
Cluster Maps

• Research interest changing over time

(a) Distribution of the first 100 documents in the personal collection

(b) Distribution of the latest 100 documents in the personal collection
WEBSOM

- http://websom.hut.fi/websom/
- SOM of Finnish news bulletins for exploring and retrieving documents
- Labels show the topics of areas in the SOM
- Coloring encodes density - light areas contain more documents
- Navigation via zooming and panning
- Documents can be retrieved on the lowest level of the visualization
ThemeScapes

- Wise et al. 1995
- Map document density to third dimension
- News article visualized as an abstract 3D landscape
- Mountains represent frequent themes in the document corpus (height proportional to number of documents relating to the theme)
- Spatial characteristics of the map should map to interconnections of themes

Cluster Map vs Keyword Search

• Cluster Map pros
  – Facilitates non-targeted exploration and browsing by spatially organizing documents
  – Provides overview of document set: major themes, sizes of clusters, relationships between themes
  – Scales up

• Cluster Map cons
  – How to label groups?
  – What does the space mean? How to label space?
  – Where to locate documents with multiple themes: both mountains, between mountains, …?
  – Relationships within documents?
  – Algorithm (SOM) is time-consuming
Cluster Map vs Keyword Search

• Keyword search pros
  – Reduces the browsing space according to user’s interests

• Keyword search cons
  – What keywords do I use?
  – What about other related documents that don’t use these keywords?
  – No initial overview
  – Mega-hit, zero-hit problem
TagClouds

- Show the frequency of words in a text
- Frequency is mapped to size and/or color
- Often found as navigation aid on web pages
- example below generated by www.wordle.net