Word Storms: Multiples of Word Clouds for Visual Comparison of Documents

Quim Castellá, Charles Sutton (WWW-2014)

Zoltán Szabó

Gatsby Unit, Tea Talk

December 18, 2014
Motivation

- Vast number of documents on the web.
- Need for quick scanning.
- Word clouds (Google: 963,000 hits; LDA - 172,000 hits):
  - One of the most popular generators: Wordle.
  - Font size = frequency of the word.
Key Problem

- Word clouds are difficult to *compare* visually.
- Word storm:

  made of word clouds,
  word cloud = subset of documents,
  allows efficient contrasting, comparison of documents.

**Goal**: visualize an entire corpus.
Cloud Examples

One cloud :=
- one document: comparing individual docs,
- one track of a conference: \( \sim \) areas,
- papers from a given period: \( \sim \) time evolution,
- one scientific field (+its subfield): \( \sim \) hierarchical categories.
Guiding Principles

1. Each cloud should represent its own document.
2. Clouds should be easy to compare/contrast.
   ⇒ Co-occurring words: similar
   - font size, color,
   - position, orientation.
Creating a Single Cloud: Notations

- Word cloud = set of words: \( \mathcal{W} = \{w_1, \ldots, w_M\} \).
- Each word \( w \in \mathcal{W} \) has a position: \( p_w = (x_w, y_w) \), font size: \( s_w \), color: \( c_w \).
- Importance of a word (=:its weight): \( \text{tf.} \)
  - \( \mathcal{W} = \) words with the top \( M \) weights.
Creating a Single Cloud

- Font size \( \propto \) word weight.
- Color, orientation: random.
- Position: spiral algorithm (next slide).
Creating a Single Cloud: Spiral Algorithm

- Given: word cloud with \( i - 1 \) words.
- New word \( w \) to the desired/random location:
  - If
    - no intersection with previous words, and
    - \( \in \) frame, then goto next word.
  - Else: \( w \) is moved outward until a valid position.
Algorithm 1 Spiral Algorithm

Require: Words $W$, optionally positions $p = \{p_w\}_{w \in W}$
Ensure: Final positions $p = \{p_w\}_{w \in W}$

1: for all words $w \in \{w_1, \ldots, w_M\}$ do
2:  if initial position $p_w$ unsupplied, sample from Gaussian
3:  count $\leftarrow 0$
4:  while $p_w$ not valid $\land$ count $< \text{Max Iteration}$ do
5:    Move $p_w$ one step along a spiral path
6:    count $\leftarrow$ count + 1
7:  end while
8:  if $p_w$ not valid then
9:    Restart with a larger frame
10: end if
11: end for
Creating a Storm

- $i^{th}$ document: $u_i = (u_{iw})$: count of word $w$ in the $i^{th}$ doc.
- $i^{th}$ word cloud: $v_i = (W_i, \{p_{iw}\}, \{c_{iw}\}, \{s_{iw}\})$.
- Alg-1:
  - Color: $\alpha$-channel = idf = $\log\left(\frac{|\text{docs}|}{|\text{docs containing } w|}\right)$.
    $\Rightarrow$ transparent: the word appears in many docs.
  - Locations:
    - Initialization: spiral method.
    - Iterate: desired locations := $\hat{\text{clouds}}[\text{previous locations}]$. 
Algorithm 2: Iterative Layout Algorithm

Require: Storm $v_i = (W_i, \{c_{iw}\}, \{s_{iw}\})$ without positions
Ensure: Word storm $\{v_1, \ldots, v_N\}$ with positions

1: for $i \in \{1, \ldots, N\}$ do
2:     $p_i \leftarrow \text{SPIRALALGORITHM}(W_i)$
3: end for
4: while Not Converged $\land$ count $< \text{Max Iteration}$ do
5:     for $i \in \{1, \ldots, N\}$ do
6:         $p_{iw}' \leftarrow \frac{1}{|V_w|} \sum_{v_j \in V_w} p_{jw}, \quad \forall w \in W_i$
7:     end for
8:     $p_i \leftarrow \text{SPIRALALGORITHM}(W_i, p_i')$
9:     count $= count + 1$
10: end while

Problem: tends to move words far away from center.
Coordinated Layout: Alg-2 – Objective

- Set of documents: $u_{1:N} = \{u_1, \ldots, u_N\}$. Storm: $v_{1:N} = \{v_1, \ldots, v_N\}$.
- Objective (how well the storm fits the corpus):

$$f_{u_{1:N}}(v_{1:N}) = \sum_{i,j=1}^{N} [d_U(u_i, u_j) - d_V(v_i, v_j)]^2 + \sum_{i=1}^{N} c(u_i, v_i).$$

  - First term: MDS. $d_U$: Euclidean distance. $\kappa \geq 0$

    $$d_V(v_i, v_j) = \sum_{w \in W_i \cup W_j} (s_{iw} - s_{jw})^2 + \kappa \sum_{w \in W_i \cap W_j} \|p_{iw} - p_{jw}\|^2.$$

  - Second term:

    $$c(u_i, v_i) = \sum_{w \in W_i} (u_{iw} - s_{iw})^2.$$
Two more penalties ($\lambda > 0$, $\mu > 0$):

$$r(v_{1:N}) = \lambda \sum_{i=1}^{N} \sum_{w, w' \in W_i} O_{i:w,w'}^2 + \mu \sum_{i=1}^{N} \sum_{w \in W_i} \|p_{iw}\|_2^2.$$  

- $O_{i:w,w'}$: minimum distance required to separate overlapping words ($w, w'$).
- Final objective: $f_{u_{1:N}}(v_{1:N}) + r(v_{1:N}) \rightarrow \min_{v_{1:N}}$.
- Optimization:
  - homotopy scheme in $\lambda$,
  - fixed subtask: gradient descent.
Iterative algorithm: fast, but not compact.
Gradient method: compact storm, but slow.
In practise: combination gives decent results.
User study: users are better in
- outlier document detection,
- the discovery of the two most similar documents.

ICML-2012:
- visualization of sessions,
  http://icml.cc/2012/whatson-all/.

Research grant abstract visualization (EPSRC):
- $1 - 5^{th} =$ material sciences, $6^{th} =$ maths.
- independent vs. coordinated layout.
EPSRC programmes: independent clouds

(a) Electronic Materials

(b) Metals and Alloys

(c) Photonic Materials

(d) Structural Ceramics and Inorganics

(e) Structural Polymers and Composites

(f) Mathematical Sciences
EPSRC programmes: coordinated storm

(a) Electronic Materials
(b) Metals and Alloys
(c) Photonic Materials
(d) Structural Ceramics and Inorganics
(e) Structural Polymers and Composites
(f) Mathematical Sciences
(a)-(e) similar: 'material', 'applications', 'properties'.

Contrast, absence of words:
- 'coating' only in (b) and (d),
- no 'material' in (f).

Informative words (transparency): 'electron' (a), 'metal' (b), 'light' (c), 'crack' (d), 'composite' (e), 'problems' (f).
Independent word clouds are difficult to compare.

Word storm:
- Similar clouds represent similar documents.
- Emphasizes the most informative words.
- Useful in comparing/contrasting documents.

Source code: [http://groups.inf.ed.ac.uk/cup/wordstorm/wordstorm.html](http://groups.inf.ed.ac.uk/cup/wordstorm/wordstorm.html)