Foto: Stana, ***Križevec***

Islands

Matjaž Zaveršnik,
Vladimir Batagelj

University of Ljubljana

XXIV International Sunbelt Social Network Conference

May 12–16, 2004, Portorož, Slovenia

Kazalo

Networks

A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{L}$ is the set of lines (links, ties). Undirected lines $\mathcal{E}$ are called *edges*, and directed lines $\mathcal{A}$ are called *arcs*. $n = \text{card}(\mathcal{V})$, $m = \text{card}(\mathcal{L})$
- $\mathcal{P}$ *vertex value functions* of properties: $p : \mathcal{V} \rightarrow A$
- $\mathcal{W}$ *line value functions* of properties: $w : \mathcal{L} \rightarrow B$

Cuts

- The *vertex-cut* of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \to \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, determined by

$$\mathcal{V}' = \{v \in \mathcal{V} : p(v) \geq t\}$$

  and $\mathcal{L}(\mathcal{V}')$ is the set of lines from $\mathcal{L}$ that have both endpoints in $\mathcal{V}'$.

- The *line-cut* of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \to \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, determined by

$$\mathcal{L}' = \{e \in \mathcal{L} : w(e) \geq t\}$$

  and $\mathcal{V}(\mathcal{L}')$ is the set of all endpoints of the lines from $\mathcal{L}'$.

- The line-cut at level t is vertex-cut at the same level for

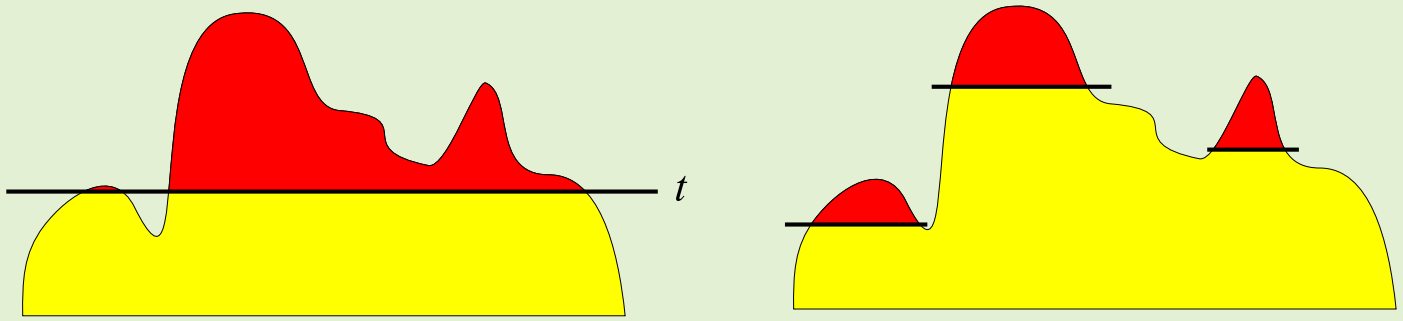
$$p(v) = \max_{u \in N(v)} w(v, u)$$

  where we preserve only lines with $w(e) \geq t$.

Simple analysis using cuts

- After making a cut at selected level t we look at the components of the $\mathcal{N}(t)$. Their number and sizes depend on t. Usually there are many small and some large components. Often we consider only components of size at least k and not exceeding K. The components of size smaller than k are discarded as noninteresting, and the components of size larger than K are cut again at some higher level.
- The values of thresholds t, k and K are determined by inspecting the distribution of vertex/line values and the distribution of component sizes and considering additional knowledge on the nature of network or goals of analysis.

Cuts and islands



Vertex islands

- Nonempty subset of vertices $\mathcal{C} \subseteq \mathcal{V}$ is *vertex island*, if
  - the corresponding induced subgraph $\mathcal{G}|\mathcal{C} = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$ is connected, and
  - the values of the vertices in the neighborhood of $\mathcal{C}$ are less than or equal to values of vertices from $\mathcal{C}$.

$$\max_{u \in N(\mathcal{C})} p(u) \leq \min_{v \in \mathcal{C}} p(v)$$

- Vertex island $\mathcal{C} \subseteq \mathcal{V}$ is *regular vertex island*, if stronger condition holds:

$$\max_{u \in N(\mathcal{C})} p(u) < \min_{v \in \mathcal{C}} p(v)$$

Some properties of vertex islands

- The sets of vertices of connected components of vertex-cut at selected level t are regular vertex islands.
- The set $\mathcal{H}_p(\mathcal{N})$ of all regular vertex islands of network $\mathcal{N}$ is complete hierarchy:
  - two islands are disjoint or one of them is subset of another one
  - each vertex belongs to at least one island
- Vertex islands are independent on the values of vertices, only the order according to their values is important. This means that we can transform the values using any monotone increasing function, and the islands remain the same.
- Two connected vertices cannot belong to two disjoint regular vertex islands.

Algorithm for determining maximal regular vertex islands of limited size

- We sink the network into the water, then we lower the water level step by step.
- Each time a new vertex v appears from the water, we check with which of the already visible islands is connected.
- We join these islands and the vertex v obtaining a new (larger) island.
- These islands are *subislands* of the new island.
- Vertex v is *port* of the new island (the vertex with the smallest value).

algorithm ...

$islands := \emptyset$
sort $\mathcal{V}$ in decreasing order according to p
for each $v \in \mathcal{V}$ (in the obtained order) **do begin**
    $island :=$ **new** $Island()$
    $island.port := v$
    $island.subislands := \{i \in islands : i \cap N(v) \neq \emptyset\}$
    $islands := islands \cup \{island\} \setminus island.subislands$
    for each $i \in island.subislands$ **do** $i.regular := p(i.port) > p(v)$
end
for each $i \in islands$ **do** $i.regular :=$ **true**

... algorithm

$L := \emptyset$
while $islands \neq \emptyset$ **do begin**
    select $island \in islands$
    $islands := islands \setminus \{island\}$
    if $|island| < min$ **then delete** $island$
    else if $|island| > max \vee \neg island.regular$ **then begin**
        $islands := islands \cup island.subislands$
        delete $island$
    end
    else $L := L \cup \{island\}$
end

Simple vertex islands

- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is *local vertex summit*, if it is regular vertex island and all of its vertices have the same value.
- Vertex island with only one local vertex summit is called *simple vertex island*.
- The types of vertex islands:
  - FLAT – all vertices have the same value
  - SINGLE – island has only one local vertex summit
  - MULTI – island has more than one local vertex summits
- Only the islands of type FLAT or SINGLE are simple islands.

Determining the type of vertex island

if $|island.subislands| = 0$ **then** $island.type :=$ FLAT
else if $|island.subislands| = 1$ **then begin**
    select $i \in island.subislands$
    if $i.type \neq$ FLAT **then** $island.type := i.type$
    else if $p(i.port) = p(v)$ **then** $island.type :=$ FLAT
    else $island.type :=$ SINGLE
end
else begin
    for each $i \in island.subislands$ **do begin**
        $ok := i.type =$ FLAT $\wedge\ p(i.port) = p(v)$
        if $\neg ok$ **then break**
    end
    if ok **then** $island.type :=$ FLAT
    else $island.type :=$ MULTI
end

Edge islands

- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is *edge island*, if it is a singleton (degenerated island) or the corresponding induced subgraph is connected and there exists a spanning tree $\mathcal{T}$, such that the values of edges with exactly one endpoint in $\mathcal{C}$ are less than or equal to the values of edges of the tree $\mathcal{T}$.

$$\max_{\substack{(u\,;\,v)\in\mathcal{L}:\\ u\in\mathcal{C}\wedge v\notin\mathcal{C}}} w((u\,;v)) \leq \min_{e\in\mathcal{L}(\mathcal{T})} w(e)$$

- Edge island $\mathcal{C} \subseteq \mathcal{V}$ is *regular edge island*, if stronger condition holds:

$$\max_{\substack{(u\,;\,v)\in\mathcal{L}:\\ u\in\mathcal{C}\wedge v\notin\mathcal{C}}} w((u\,;v)) < \min_{e\in\mathcal{L}(\mathcal{T})} w(e)$$

Some properties of edge islands

- The sets of vertices of connected components of line-cut at selected level t are regular edge islands.
- The set $\mathcal{H}_w(\mathcal{N})$ of all nondegenerated regular edge islands of network $\mathcal{N}$ is hierarchy (not necessarily complete):
  - two islands are disjoint or one of them is subset of another one
- Edge islands are independent on the values of edges, only the order according to their values is important. This means that we can transform the values using any monotone increasing function, and the islands remain the same.
- Two connected vertices may belong to two disjoint regular edge islands.

Algorithm for determining maximal regular edge islands of limited size

- We sink the network into the water, then we lower the water level step by step.
- Each time a new edge e appears from the water, we check with which of the already visible islands is connected (there are exactly two such islands).
- We join these two islands obtaining a new (larger) island.
- These islands are *subislands* of the new island.
- Edge e is *port* of the new island (not necessarily the edge with the smallest value).

algorithm ...

$islands := \{\{v\} : v \in \mathcal{V}\}$
for each $i \in islands$ **do** $i.port :=$ **null**
sort $\mathcal{L}$ in decreasing order according to w
for each $e(u\,;v) \in \mathcal{L}$ (in the obtained order) **do begin**
    $i1 := island \in islands : u \in island$
    $i2 := island \in islands : v \in island$
    if $i1 \neq i2$ **then begin**
        $island :=$ **new** $Island()$
        $island.port := e$
        $island.subisland1 := i1$
        $island.subisland2 := i2$
        $islands := islands \cup \{island\} \setminus \{i1, i2\}$
        $i1.regular := i1.port =$ **null** $\vee\, w(i1.port) > w(e)$
        $i2.regular := i2.port =$ **null** $\vee\, w(i2.port) > w(e)$
    end
end
for each $i \in islands$ **do** $i.regular :=$ **true**

... algorithm

$L := \emptyset$
while $islands \neq \emptyset$ **do begin**
    select $island \in subislands$
    $subislands := subislands \setminus \{island\}$
    if $|island| < min$ **then delete** $island$
    else if $|island| > max \vee \neg island.regular$ **then begin**
        $islands := islands \cup \{island.subisland1, island.subisland2\}$
        delete $island$
    end
    else $L := L \cup \{island\}$
end

Simple edge islands

- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is *local edge summit*, if it is regular edge island and there exists a spanning tree of the corresponding induced network, in which all edges have the same value as the edge with the largest value.
- Edge island with only one local edge summit is called *simple edge island*.
- The types of edge islands:
  - FLAT – there exists a spanning tree, in which all edges have the same value as the edge with the largest value.
  - SINGLE – island has only one local edge summit.
  - MULTI – island has more than one local edge summits.
- Only the islands of type FLAT or SINGLE are simple islands.

Determining the type of edge islands

$$p1 := i1.type = \text{FLAT} \wedge (i1.port = \textbf{null} \vee w(i1.port) = w(e))$$
$$p2 := i2.type = \text{FLAT} \wedge (i2.port = \textbf{null} \vee w(i2.port) = w(e))$$

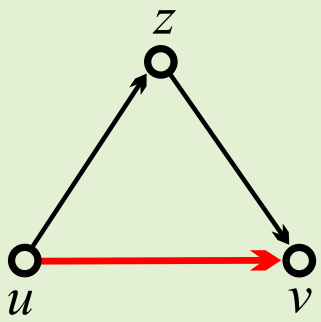
if $p1 \wedge p2$ **then** $island.type := \text{FLAT}$
else if $p1 \vee p2$ **then** $island.type := \text{SINGLE}$
else $island.type := \text{MULTI}$

Example: The Edinburgh Associative Thesaurus

- The Edinburgh Associative Thesaurus is a set of word association norms showing the counts of word association as collected from subjects.
- The data were collected by asking several people to say a word which first comes to their minds upon receiving the stimulus word.
- The network contains 23219 vertices (words) and 325624 arcs (stimulus→response), including 564 loops. Almost 70% of arcs have value 1.
- The subjects were mostly undergraduates from a wide variety of British universities. The age range of the subjects was from 17 to 22 with a mode of 19. The sex distribution was 64 per cent male and 36 per cent female. The data were collected between June 1968 and May 1971.

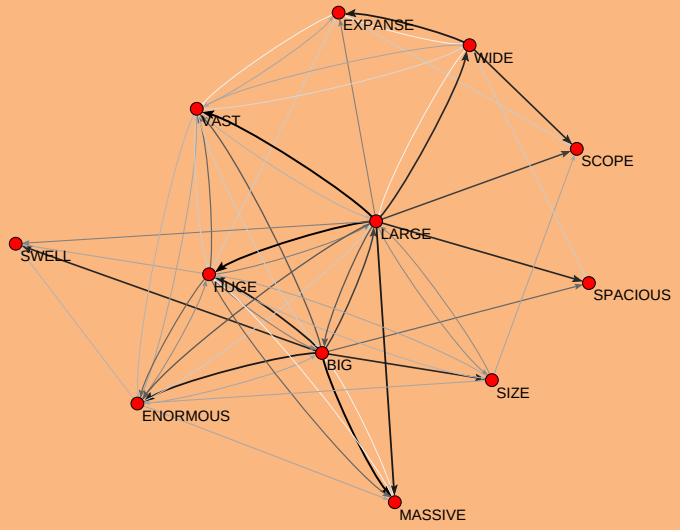
Example: The Edinburgh Associative Thesaurus

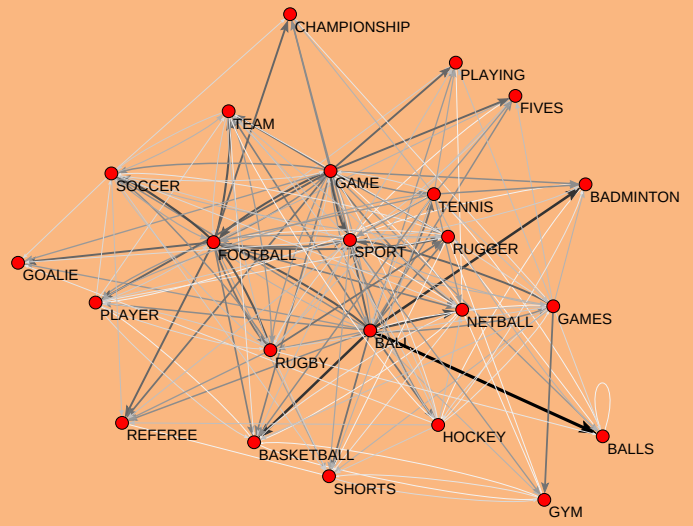
- We would like to identify the most important themes – groups of words with the strongest ties.
- For each arc we determined its weight by counting, to how many transitive triangles it belongs (we are also interested in indirect ties).

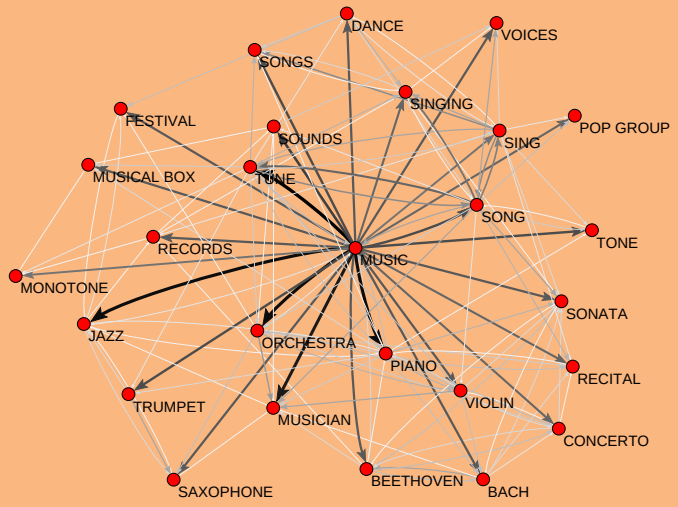


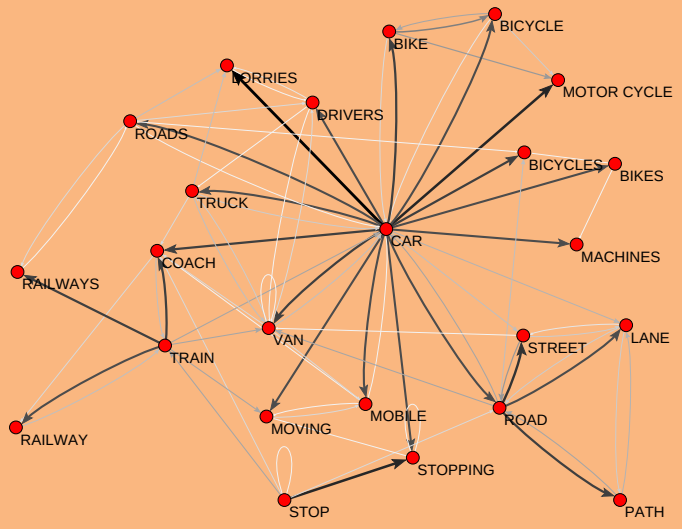
- There are 53 edge islands of size at least 5 and at most 30. They contain 664 vertices (all together).

Selected themes in EAT

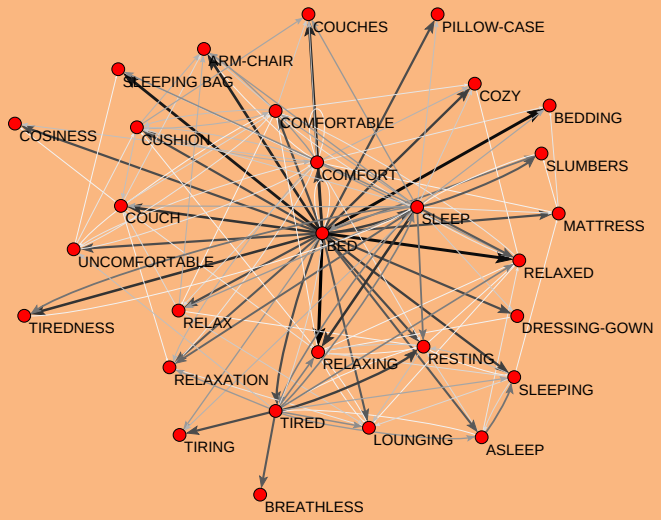


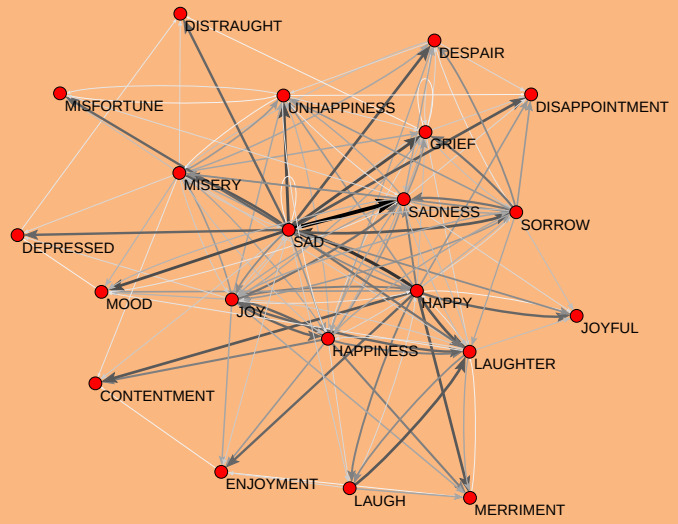


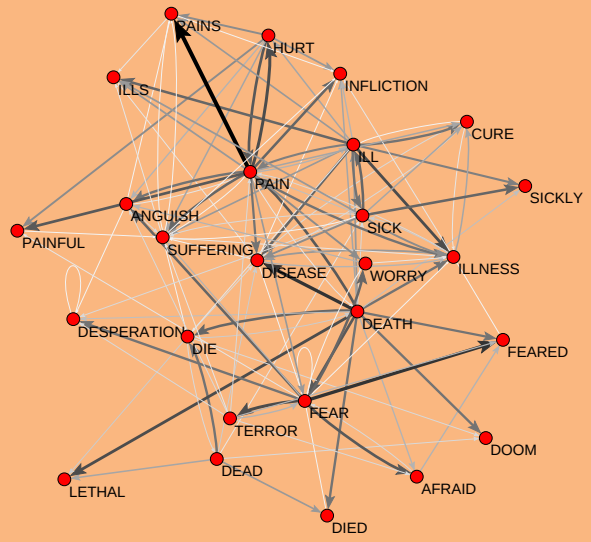


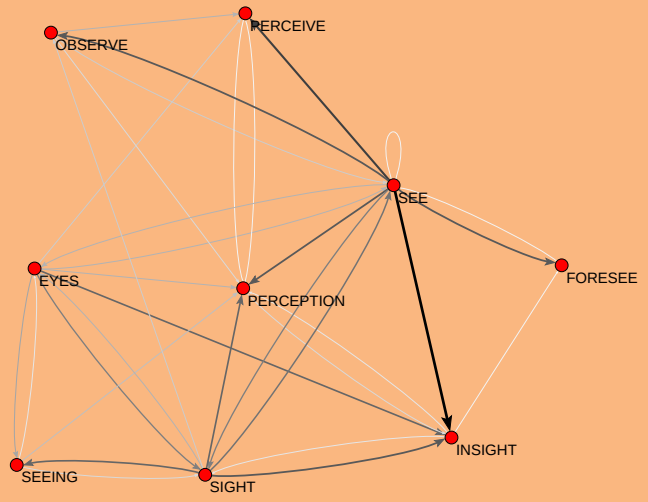


Selected themes in EAT

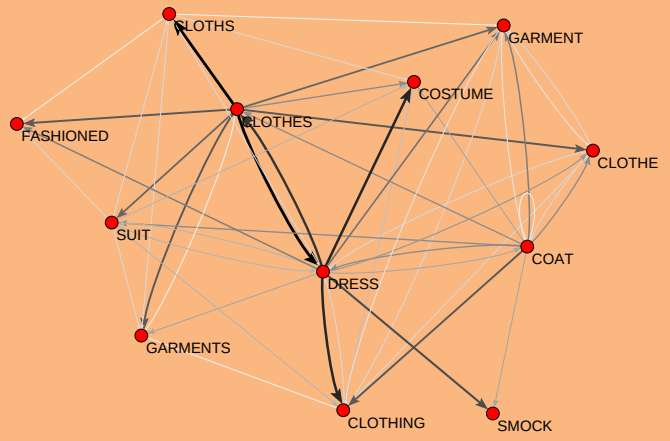


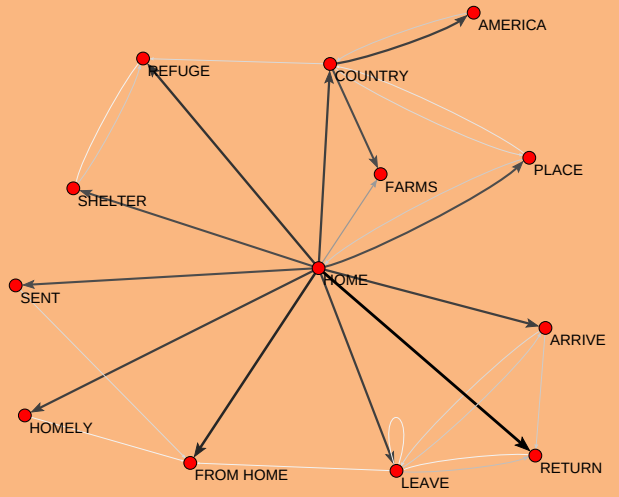


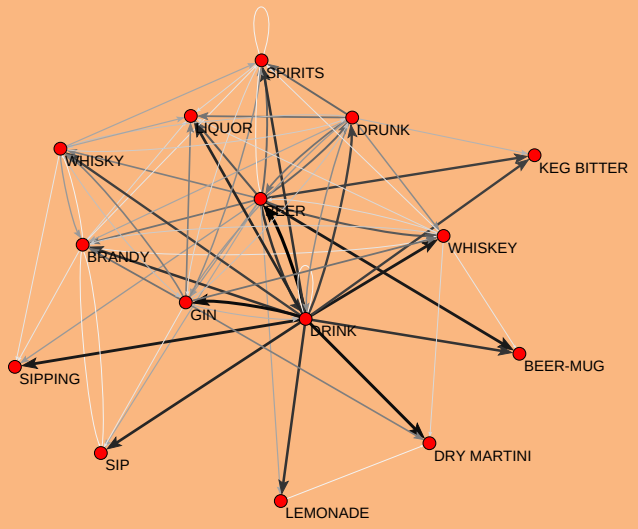


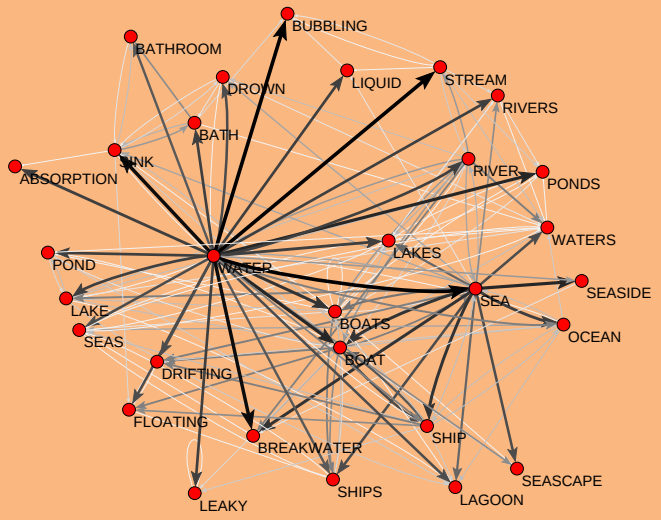


Selected themes in EAT

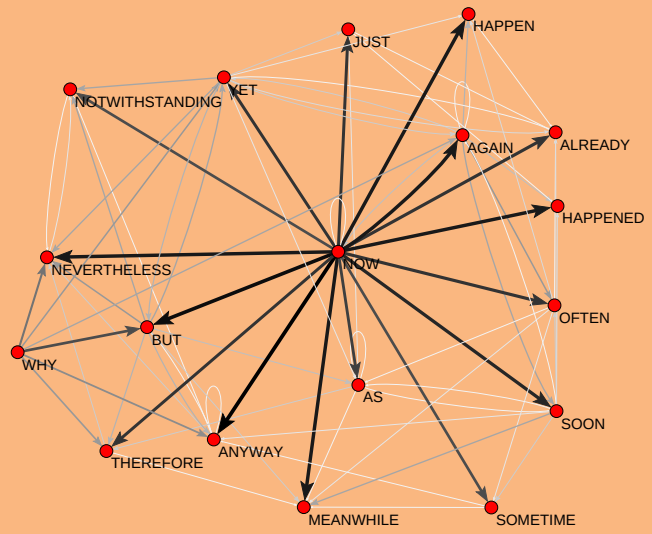


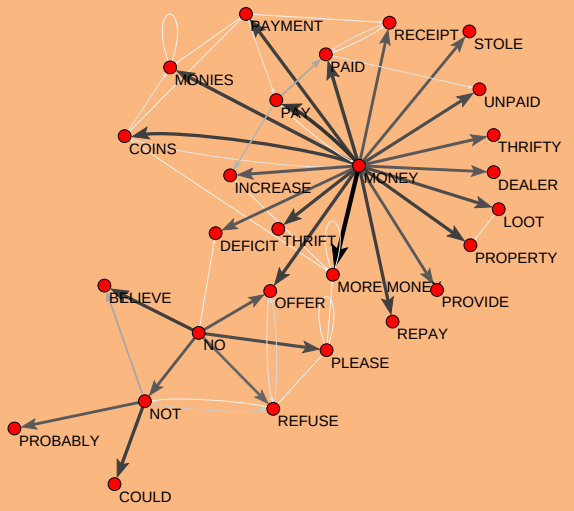


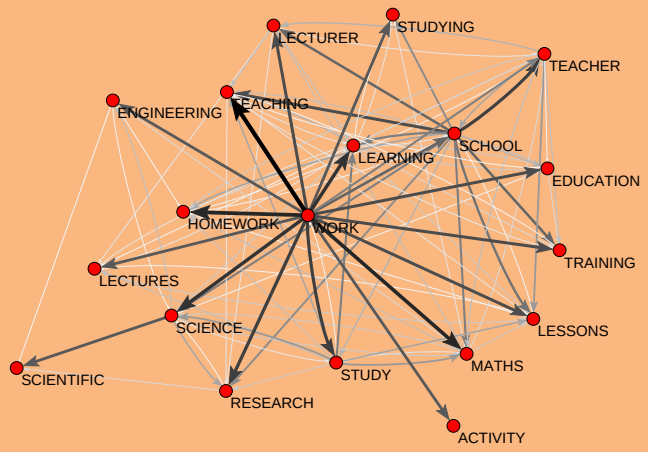


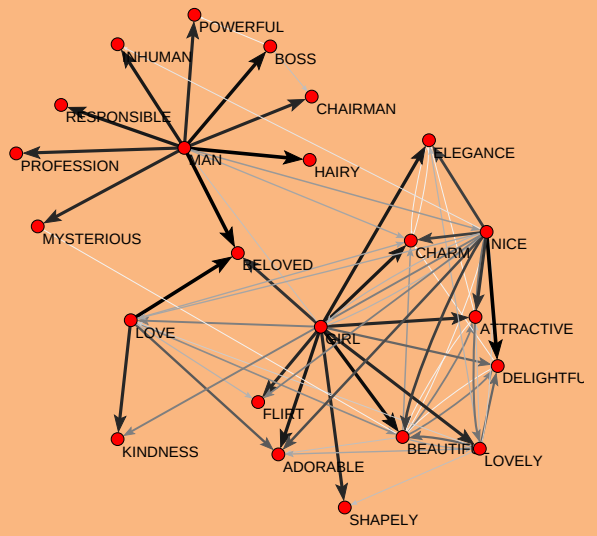


Selected themes in EAT









Conclusions

- We proposed an approach to the analysis of networks that can be used also for very large networks with millions of vertices and edges.
- Very large/small numbers that result as weights in large networks are not easy to use. One possibility to overcome this problem is to use the logarithms of the obtained weights – logarithmic transformation is monotone and therefore preserve the ordering of weights (the importance of vertices and edges). The transformed values are also more convenient for visualization with line thickness of edges.
- Invitation: Saturday, May 15, 8:30–8:55, Hall 5,
  Nataša Kejžar, Simona Korenjak-Černe, Vladimir Batagelj,
  Analysis of US Patents Network