Language Phenomena and Graphs

Lecture at ESSLLI 2022

DOI: 10.5281/zenodo.6667766

Outline

1. Introduction
2. Graphs and Language
3. Graphs and Computation
4. Centrality Measures
5. Case Studies
6. Conclusion

Section 1

Introduction

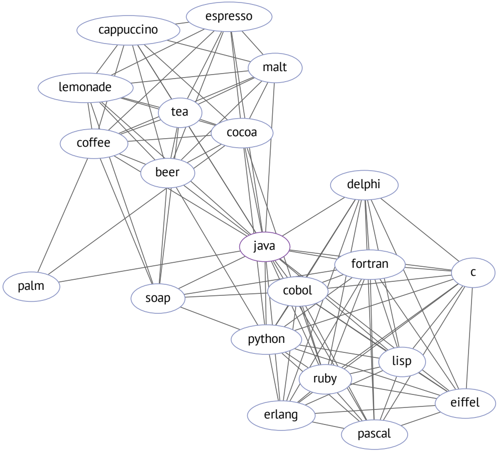
Introduction

- Natural Language Processing (NLP) focuses on the *analysis* and *synthesis* of natural language
- Linguistic phenomena instantiate in linguistic data, showing interconnections and relationships
- In this course we will learn how *graphs, computation, and language* are tightly connected
- We will start with classic graph-based NLP techniques and finish with modern approaches

Source: Adamovich (2015)

Motivation

Look at this *distributional thesaurus*!



Source: Ustalov et al. (2019)

- This graph represents words and their connections
- Can we learn word meanings from its structure?
- Can we infer linguistic knowledge computationally?
- **Yes.**

Section 2

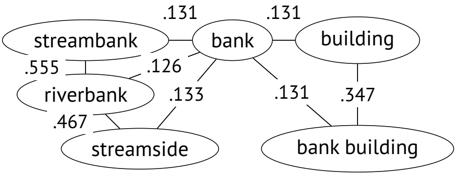
Graphs and Language

Graph Theory Essentials I

Definition

A graph is a tuple $G = (V, E)$, where V is a set of objects called *nodes* and $E \subseteq V^2$ is a set of unordered pairs called *edges*.

- Graphs can be *weighted*, i.e., there is $w : (u, v) \to \mathbb{R}$, $\forall (u, v) \in E$
- A neighborhood $G_u = (V_u, E_u)$ is a subgraph induced from G containing the nodes *incident* to $u \in V$ without u
- A node degree $\deg(u) = |V_u|$ is the number of neighbors of the node $u \in V$; maximal node degree is $\Delta(G) = \max\limits_{u \in V} \deg(u)$

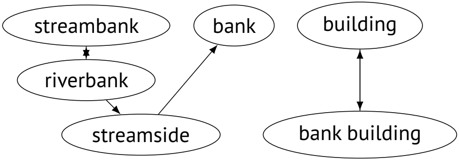


$$\deg(\text{bank}) = 5$$
$$\Delta(G) = 5$$

Graph Theory Essentials II

Graphs can be *directed*, so the edges are ordered pairs and called *arcs*.

- There are *indegrees* $\deg^-(u)$ and *outdegrees* $\deg^+(u)$, i.e., $\deg^-(u) = |(v, u) \in E|, u \in V$
- *Successors* $\mathrm{succ}(u) \subset V$ are the nodes reachable from $u \in V$



$$\deg^-(\text{bank}) = 0$$
$$\deg^+(\text{bank}) = 1$$
$$\mathrm{succ}(\text{bank}) = \emptyset$$
$$\mathrm{succ}(\text{streamside}) = \{\text{bank}\}$$

Graph Theory Essentials III

- The maximal number of edges in an *undirected* graph is

$$\frac{|V|(|V|-1)}{2}$$

- The maximal number of arcs in a *directed* graph is

$$|V|(|V|-1)$$

- The sum of degrees equals twice the number of edges (handshaking lemma):

$$\sum_{u \in V} \deg(u) = 2|E|$$

- Degree distribution is the fraction of nodes in the graph with degree $k \in \mathbb{Z}^{0+}$:

$$P(k) = \frac{|u \in V : \deg(u) = k|}{|V|}$$

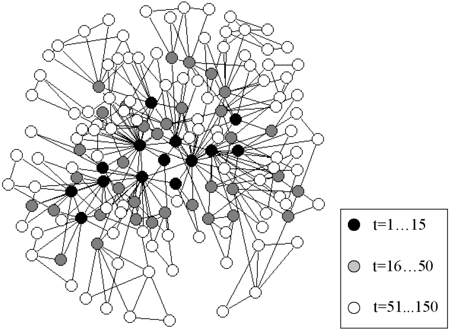
Can We Trust Language Graphs?

Graphs representing linguistic phenomena follow similar distributions and exhibit similar properties (Biemann, 2012):

- *co-occurrence networks* tend to follow the Dorogovtsev-Mendes distribution (2001),
- *semantic networks* tend to follow the scale-free properties (Steyvers et al., 2005), etc.

Yes We Can

These properties do not depend on a language w.r.t. the parameters (Kapustin et al., 2007).

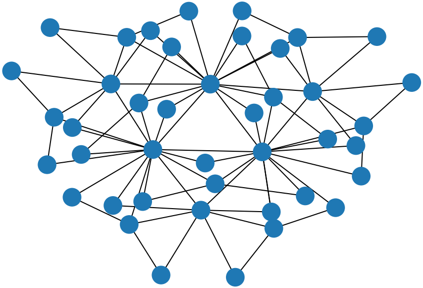


Source: Steyvers et al. (2005)

Co-Occurrence Graphs

- A pair of words are said to *co-occur* if they both appear together
- *Co-occurrence networks* tend to follow the Dorogovtsev-Mendes distribution (2001):

$$P(k) \cong \frac{1}{2} k^{-\frac{3}{2}}$$

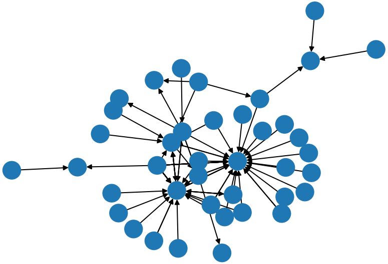


Semantic Networks

- Semantic relations are synonymy, antonymy, hypernymy/hyponymy, holonymy/meronymy, etc.
- A semantic network (or a knowledge graph) is a graph that represents semantic relations between concepts
- *Semantic networks* tend to follow the scale-free properties (Steyvers et al., 2005):

$$P(k) \propto k^{-\gamma}$$

for some $\gamma \in \mathbb{R}$

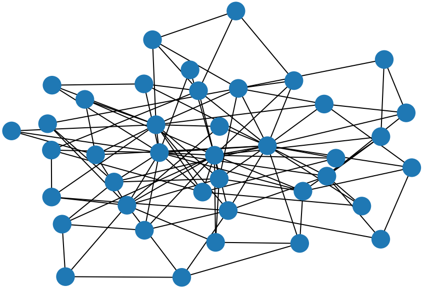


World Wide Web

- World Wide Web follows the scale-free degree distribution with the *preferential attachment* mechanism (Barabási et al., 1999):

$$P(k) \propto k^{-3}$$

- "The rich get richer"
- Citation networks and social networks also follow this distribution



Graphs and Language: Wrap-Up

- We have all the necessary definitions and now we can reason about graphs and their elements
- Graphs are different w.r.t. the represented information, internal structure, and size
- Real-world graphs tend to follow well-known distributions, and this is good!

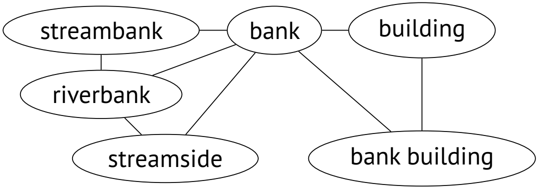
Source: rawpixel (2017)

Section 3

Graphs and Computation

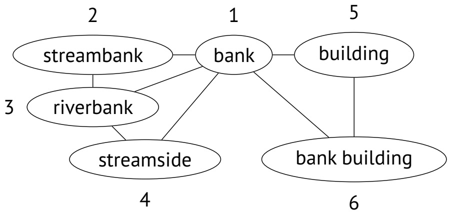
Graphs and Computation

- Graphs need to be represented both mathematically and in computer memory
- Formal representations: edge and adjacency lists, adjacency and incidence matrices, etc.
- Computer representations: non-matrix, dense and sparse matrices



Edge List

Edge List is the simplest way to define a graph by listing its edges.

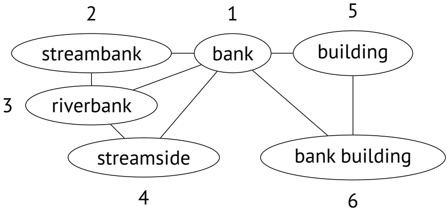


bank	streambank
bank	riverbank
streamside	bank
bank	building
bank	bank building
streambank	riverbank
riverbank	streamside
building	bank building

- Nodes with zero degree cannot be represented

Adjacency List

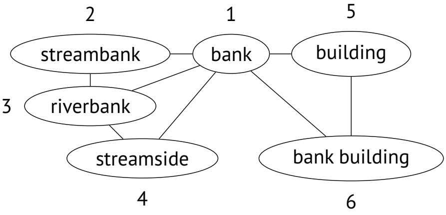
Adjacency List is the generalization of an *edge list* in which each node lists its incident nodes.



bank	streambank, riverbank, streamside, building, bank building
streambank	riverbank
riverbank	streamside
streamside	
building	bank building
bank building	

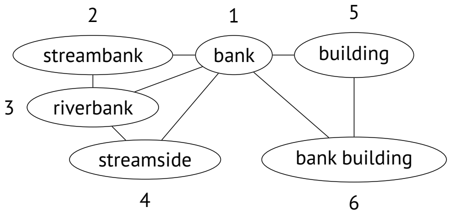
Adjacency Matrix

Adjacency Matrix $A \in \mathbb{R}^{|V|\times|V|}$ is a square matrix that indicates whether pairs of nodes are adjacent or not.



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Incidence Matrix $B \in \mathbb{R}^{|V|\times|E|}$ is a Boolean matrix that indicates whether the nodes are incident in edges.

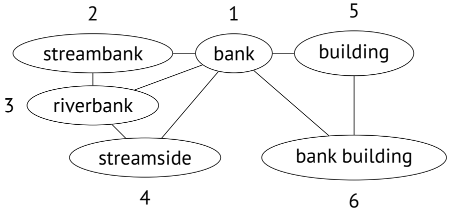


$$B = \begin{array}{c} \begin{array}{cccccccc} e_{12} & e_{13} & e_{14} & e_{15} & e_{16} & e_{23} & e_{34} & e_{56} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Degree Matrix

A node *degree* is the number of nodes incident to this node, e.g., $\deg(\text{riverbank}) = 3$; the maximal degree Δ in this graph is 5

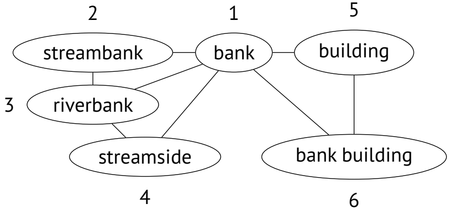
Degree Matrix $D \in \mathbb{Z}^{0+^{|V|\times|V|}}$ is a diagonal matrix that indicates the corresponding node degrees.



$$D = \begin{pmatrix} 5 & & & & & \\ & 2 & & & & \\ & & 3 & & & \\ & & & 2 & & \\ & & & & 2 & \\ & & & & & 2 \end{pmatrix}$$

Laplacian Matrix

Laplacian Matrix $L = D - A = B^\top B$.

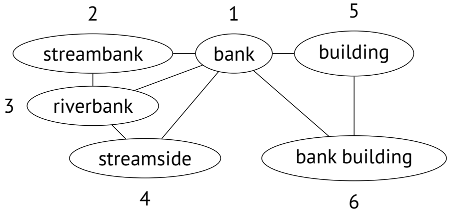


$$L = \begin{pmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ -1 & 0 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

- L is positive-semidefinite, i.e., $\vec{x}^\top L \vec{x} \geq 0, \forall \vec{x} \in \mathbb{R}^{|V|} \setminus \{0\}$, and symmetric, enabling the spectral graph theory (von Luxburg, 2007)
- For *digraphs*, we have to choose between indegree and outdegree matrix

Normalized Laplacian Matrix

Normalized Laplacian Matrix $L^{\text{norm}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.



$$D^{-\frac{1}{2}} = \begin{pmatrix} .45 & & & & & \\ & .71 & & & & \\ & & .58 & & & \\ & & & .71 & & \\ & & & & .71 & \\ & & & & & .71 \end{pmatrix}$$

$$L^{\text{norm}} = \begin{pmatrix} 1 & -.32 & -.26 & -.32 & -.32 & -.32 \\ -.32 & 1 & -.41 & 0 & 0 & 0 \\ -.26 & -.41 & 1 & -.41 & 0 & 0 \\ -.32 & 0 & -.41 & 1 & 0 & 0 \\ -.32 & 0 & 0 & 0 & 1 & -.50 \\ -.32 & 0 & 0 & 0 & -.50 & 1 \end{pmatrix}$$

- All eigenvalues of the normalized Laplacian are real and non-negative

Graphs in Computer Memory

Every representation differs in terms of intended purpose, the computational complexity of operations is different:

- Matrix-free representations
- Dense matrix representations
- Sparse matrix representations

Source: Amos (2011)

Dict-of-Dict-of-Dict

Dictionary for the *source* node contains a dictionary for the *target* node that contains a dictionary for edge *data*.

$$\{\text{bank building} : \{\text{building} : \{\text{weight} : 1\}, \text{bank} : \{\text{weight} : 1\}\}, \dots\}$$

Used by NetworkX (Hagberg et al., 2008).

Set and Set

One set for *nodes* and another set of *edges*.

Nodes

bank
streambank
riverbank
streamside
building
bank building

Edges

bank	streambank
bank	riverbank
streamside	bank
bank	building
bank	bank building
streambank	riverbank
riverbank	streamside
building	bank building

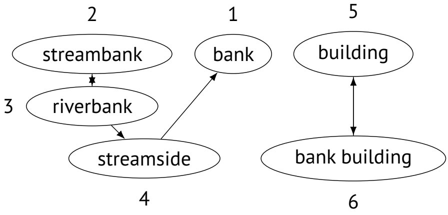
Used by JGraphT (Michail et al., 2020).

Dense Matrix Representations

In general, matrices are stored in computer memory as contiguous arrays of numbers:

- Row-Major Order Matrix
- Column-Major Order Matrix
- Block Matrix

As an example, we will use the adjacency matrix of a *directed graph* so it is non-symmetric.



Row-Major Order Matrix

In the **row-major order matrix**, the row arrays contain column arrays.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]

Column-Major Order Matrix

In the **column-major order matrix**, the column arrays contain row arrays.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

[0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]

Block Matrix

In a **block matrix**, the matrix is split into several blocks, and each block is stored as a contiguous array in memory.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Sparse Matrix Representations

In language graphs, most graph matrices are *sparse* and contain many zeroes.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Only 6 elements out of these 36 are non-zeroes!

There are representations that take sparseness into account:

- Coordinate Sparse Matrix (COO) that is convenient for modification
- Compressed Sparse Rows/Columns (CSR/CSC) that are convenient for matrix operations

Coordinate Sparse Matrix

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & \boxed{1} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned} \texttt{data} &= [\boxed{1}, 1, \boxed{1}, 1, 1, 1] \\ \texttt{row} &= [\boxed{1}, 2, \boxed{2}, 3, 4, 5] \\ \texttt{col} &= [\boxed{2}, 1, \boxed{3}, 0, 5, 4] \end{aligned}$$

Each element of A is positioned by $(\texttt{row}, \texttt{col})$ and contains the corresponding element of `data`.

Compressed Sparse Rows

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned} \texttt{data} &= [1, 1, 1, 1, 1, 1] \\ \texttt{colind} &= [2, 1, 3, 0, 5, 4] \\ \texttt{rowind} &= [0, 0, 1, 3, 4, 5, 6] \end{aligned}$$

In CSR, for the i-th row:

- column indices are stored in `colind[rowind[i]:rowind[i + 1]]`
- elements are stored in `data[rowind[i]:rowind[i + 1]]`

Compressed Sparse Columns

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\texttt{data} = [1, 1, \boxed{1}, 1, 1, 1]$$
$$\texttt{rowind} = [3, 2, \boxed{1}, 2, 5, 4]$$
$$\texttt{colind} = [0, 1, \boxed{2}, 3, 4, 5, 6]$$

In CSC, for the i-th column:

- row indices are stored in `rowind[colind[i]:colind[i + 1]]`
- elements are stored in `data[colind[i]:colind[i + 1]]`

Graph Search Algorithms

Often one needs to *traverse* the graph, for which there are two approaches:

- **Breadth-First Search** (BFS) that explores neighbors at the present depth level before moving to the next level
- **Depth-First Search** (DFS) that moves to the deepest level before exploring all the neighbors

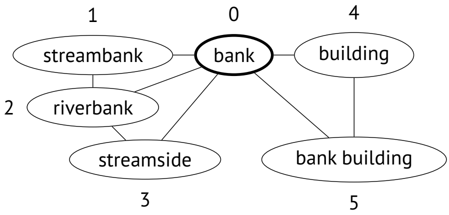
Both algorithms are data intensive; parallel BFS enables the **Graph500** benchmark of high-performance computing systems: `https://graph500.org/`.

Source: Merrill (2014)

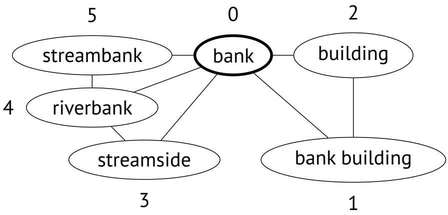
Graph Search Algorithms: Example

Suppose we start traversing from the node “bank”.

Breadth-First Search

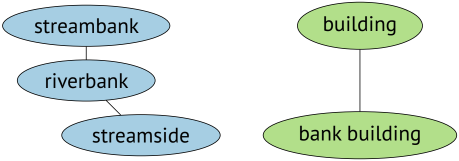


Depth-First Search

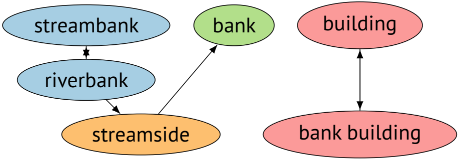


Connected Components

In *undirected* graphs, a connected component is a subset of nodes that are connected via paths.

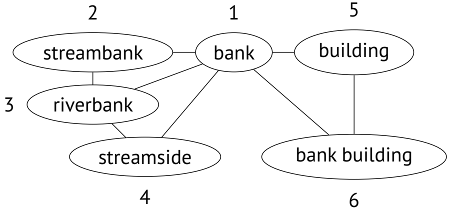


In *directed* graphs, a strongly-connected component is a subset of nodes that are reachable from each other.



Shortest Paths

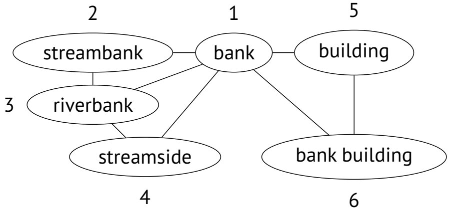
A path in a graph is a sequence of edges from node $u \in V$ to $v \in V$, e.g., $(1 \to 6 \to 5)$.



- The **shortest path** is the path with the smallest number of steps, e.g., $(1 \to 5)$
- Well-known approaches are Dijkstra's algorithm (1959), Johnson's algorithm (1977), see more in Cormen et al. (2022, Chapters 22–23)

Random Walks

A **random walk** is a succession of random steps on a mathematical space (on a graph in our case).



$$(4)$$
$$(3 \to 2)$$
$$(5 \to 1 \to 4)$$
$$(6 \to 5 \to 6 \to 5)$$
$$(1 \to 2 \to 3 \to 1 \to 5)$$

Stochastic Matrices

- Recall that the adjacency matrix A represents edge weights in a graph G
- A column-normalized matrix M is called a *stochastic matrix* that shows transition probabilities between nodes of G:

$$M_{ij} = \frac{A_{ij}}{\sum_{u_k \in V} A_{kj}}$$

- For each node $u \in V$, we can obtain the probability of random walking to other nodes

$$M = \begin{pmatrix} 0 & 0 & .33 & .5 & .5 & .5 \\ .2 & 0 & .33 & 0 & 0 & 0 \\ .2 & 1 & 0 & .5 & 0 & 0 \\ .2 & 0 & .33 & 0 & 0 & 0 \\ .2 & 0 & 0 & 0 & 0 & .5 \\ .2 & 0 & 0 & 0 & .5 & 0 \end{pmatrix}$$

$$\vec{x} = (1, 0, 0, 0, 0, 0)^\top$$

$$M\vec{x} = (0, .2, .2, .2, .2, .2)^\top$$

$$MM\vec{x} = (.37, .07, .3, .07, .1, .1)^\top$$

Keep in mind this idea, we will come back to it soon!

Stochastic Matrices: Ergodicity

- If one can travel from any node to any other node with a non-zero probability, M is *irreducible*
- If one can not return to the chosen node after some transition with certainty, M is *aperiodic*
- If the stochastic matrix M is *ergodic*, i.e., irreducible and aperiodic, random walks converge to a *stationary distribution*

$$M = \begin{pmatrix} 0 & 0 & .33 & .5 & .5 & .5 \\ .2 & 0 & .33 & 0 & 0 & 0 \\ .2 & 1 & 0 & .5 & 0 & 0 \\ .2 & 0 & .33 & 0 & 0 & 0 \\ .2 & 0 & 0 & 0 & 0 & .5 \\ .2 & 0 & 0 & 0 & .5 & 0 \end{pmatrix}$$

Graphs and Computation: Wrap-Up

- Graphs have to be represented in computer memory; the best representation takes into account graph structure and usage pattern
- Examining paths and walks reveals important information about the graph structure
- There is a connection between probability theory and graph theory, allowing the use of stochastic matrices to reason about graphs

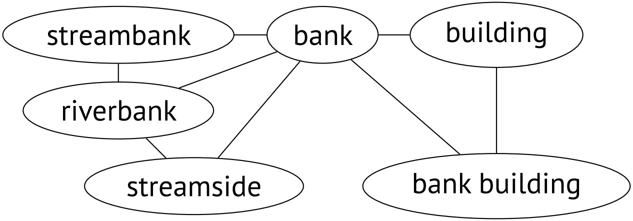
Source: rawpixel (2018)

Section 4

Centrality Measures

What is Centrality?

Which node is the most important in the graph $G = (V, E)$?



- Node **centrality** $C(u) \in \mathbb{R}$ quantifies the importance of a node $u \in V$
- There is also a similar concept of edge centrality $C(e) \in \mathbb{R}$, which is defined for an edge $e \in E$

Centrality in NLP

We will review several centrality measures popular in NLP applications (Mihalcea et al., 2011; Boudin, 2013):

- degree centrality
- closeness centrality (Bavelas, 1950)
- betweenness centrality (Freeman, 1977)
- eigenvector centrality (Bonacich, 1987)

There is a multitude of variations, we will cover some of them, too.

Source: Tama66 (2018)

Degree Centrality

- **Degree centrality** $C_D(u)$ is a simple centrality measure that is defined as the number of nodes incident to the node $u \in V$:

$$C_D(u) = \deg(u)$$

- There are variations, such as **normalized degree centrality** $C'_D(u)$, that normalize the degree by the number of remaining nodes $|V| - 1$:

$$C'_D(u) = \frac{\deg(u)}{|V| - 1}$$

V	$C_D(u)$	$C'_D(u)$
bank	5	1
streambank	2	.4
riverbank	3	.6
streamside	2	.4
building	2	.4
bank building	2	.4

Closeness Centrality

- Let distance $d(u, v) \in \mathbb{Z}^{0+}$ be the length of the shortest path from $u \in V$ to $v \in V$
- Bavelas (1950) formulated the **closeness centrality** $C_C(u)$ as a reciprocal of the sum of shortest path lengths:

$$C_C(u) = \frac{1}{\sum_{v \in V} d(v, u)}$$

- Comparison between different graphs is possible by normalizing $C_C(u)$ by the number of nodes $|V|$: $C'_C(u) = |V| \cdot C_C(u)$

V	$C_C(u)$	$C'_C(u)$
bank	1	6
streambank	.63	3.75
riverbank	.63	3.75
streamside	.71	4.29
building	.63	3.75
bank building	.63	3.75

Betweenness Centrality

- If a large number of shortest paths between nodes $s, t \in V$ pass through the node $u \in V$, this node u is important
- Let $\sigma_{st}(u)$ be the number of shortest paths from s to t via u such that $s \neq v \neq t$
- Let σ_{st} be the total number of shortest paths from s to t
- Freeman (1977) formulated **betweenness centrality** as the sum of ratios:

$$C_B(u) = \sum_{s \neq u \neq t \in V} \frac{\sigma_{st}(u)}{\sigma_{st}}$$

V	$C_B(u)$
bank	.65
streambank	0
riverbank	0.05
streamside	0
building	0
bank building	0

Edge Betweenness Centrality

- It is possible to naturally expand this centrality measure to edges as well
- Let $\sigma_{st|e}(u)$ be the number of shortest paths from $s \in V$ to $t \in V$ via edge $e \in E$ that is incident to $u \in V$
- Brandes (2008) proposed **Edge Betweenness Centrality** that quantifies the number of shortest paths passing through the edges E:

$$C_B(e) = \sum_{s,t \in V} \frac{\sigma_{st|e}(u)}{\sigma_{st}}$$

E	$C_B(e)$
{bank, streambank}	.23
{bank, riverbank}	.20
{streamside, bank}	.23
{bank, building}	.27
{bank, bank building}	.27
{streambank, riverbank}	.10
{riverbank, streamside}	.10
{building, bank building}	.07

Eigenvector Centrality

- Recall that the eigenvector $\vec{x}$ is $A\vec{x} = \lambda\vec{x}$ and λ is the eigenvalue that defines the length of the transformation
- Bonacich (1987) proposed **eigenvector centrality** $C_E(u)$ in which the centrality of node $u_i \in V$ is the i-th element of the largest eigenvector of A:

$$C_E(u) = \frac{1}{\lambda} \sum_{v \in V_u} C_E(v)$$

- We can obtain the largest eigenvector with the *power method*: $\vec{x}_{i+1} = \frac{A\vec{x}_i}{\|A\vec{x}_i\|}$ (Perron–Frobenius theorem)

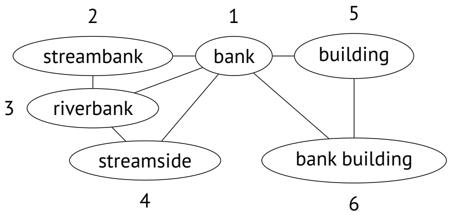
V	$C_E(u)$
bank	.60
streambank	.35
riverbank	.44
streamside	.35
building	.31
bank building	.31

Eigenvector Centrality: Algorithm

Input: graph $G = (V, E)$, adjacency matrix A
Output: eigenvector centralities $C_E(u), \forall u \in V$

1: $\vec{x} \leftarrow \texttt{random}(\mathbb{R}^{|V|})$
2: **while** $\vec{x}$ changes **do** ▷ Estimate $\vec{x}$ using the power method
3: $\quad \vec{x} \leftarrow \dfrac{A\vec{x}}{\|A\vec{x}\|}$
4: $C_E(u_i) \leftarrow \vec{x}_i$ **for all** $u_i \in V$
5: **return** C_E

Eigenvector Centrality: Example

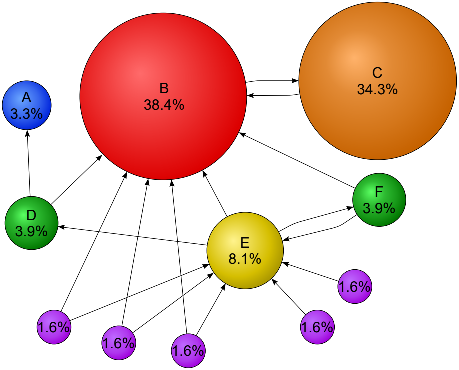


V	$C_E(u)$
bank	.60
streambank	.35
riverbank	.44
streamside	.35
building	.31
bank building	.31

This is an example using the graph from Ustalov et al. (2019, Figure 2)

Random Walks and Centrality

- Recall that for random walks to converge to a stationary distribution, our graph G should be either undirected and connected or directed and strongly-connected
- What if we can work around this strong requirement?
- Let us make G (strongly-)connected by adding the missing edges/arcs!



Source: 345Kai et al. (2007)

PageRank

PageRank (1998) is a probabilistic graph centrality measure that simulates how a user travels across the Web (*the billion-dollar algorithm*).

- The user visits a page and then either follows to a linked page or teleports to a random page with probability $1 - d$ (called the *damping factor*, $d = 0.85$)
- "Dangling" nodes with zero outdegree are artificially connected to all other nodes in the graph
- PageRank is very well-studied, one might enjoy reading a more detailed analysis by Gallardo (2007)

$$\Pr(u) = d \sum_{v \in \mathrm{In}(u)} \frac{\Pr(v)}{|\operatorname{Out}(u)|} + \frac{1-d}{|V|}$$

$$P^\top = \left(d \cdot P + \cdot \frac{1-d}{|V|} \cdot \mathbf{1}\right)^\top$$

PageRank: Algorithm

Input: graph $G = (V, E)$, adjacency matrix A, damping factor $0 \leq d \leq 1$
Output: PageRank $C_P(u), \forall u \in V$

$A_{ij} \leftarrow \frac{1}{|V|}$ **for** $1 \leq i \leq |V|, 1 \leq j \leq |V|$ **if node** i **is dangling**

$P_{ij} \leftarrow \frac{A_{ij}}{\sum_{1 \leq k \leq |V|} A_{kj}}$ **for all** $1 \leq i \leq |V|, 1 \leq j \leq |V|$ $\triangleright$ Normalize

$P \leftarrow d \cdot P + \frac{1-d}{|V|} \cdot \mathbf{1}$ $\triangleright$ Apply damping factor

$\vec{x} \leftarrow \mathtt{random}(\mathbb{R}^{|V|})$ $\triangleright$ Same as in eigenvector centrality

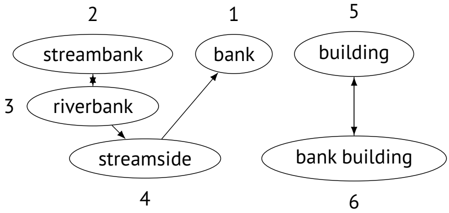
while $\vec{x}$ changes **do** $\triangleright$ Estimate $\vec{x}$ using the power method

$\quad \vec{x} \leftarrow \dfrac{P^\top \vec{x}}{\|P^\top \vec{x}\|}$

$C_P(u_i) \leftarrow \vec{x}_i$ **for all** $u_i \in V$

return C_P

PageRank: Example



V	$C_P(u)$
bank	.12
streambank	.09
riverbank	.12
streamside	.09
building	.28
bank building	.28

This is an example using the graph from Ustalov et al. (2019, Figure 2)

Centrality Measures: Wrap-Up

- Centrality measures allow one to determine the most important node in the graph
- There is no silver bullet (Boudin, 2013): pick the method that matches your graph structure well
- PageRank is not the only algorithm in its kind, see HITS by Kleinberg (1999)

Source: Free-Photos (2016)

Section 5

Case Studies

Case Studies

We will discuss three classic applications of graph-based methods for NLP:

- Keyword Extraction (Mihalcea et al., 2004a)
- Text Summarization (Mihalcea et al., 2004a)
- Word Sense Disambiguation (Mihalcea et al., 2004b)

Implementations: pytextrank and biased_textrank.

Keyword Extraction

Mihalcea et al. (2004a) proposed an unsupervised approach for *keyword extraction* using graphs.

1. Build a word graph:

   nodes are words

   edges are co-occurrences
2. Run PageRank
3. Extract phrases

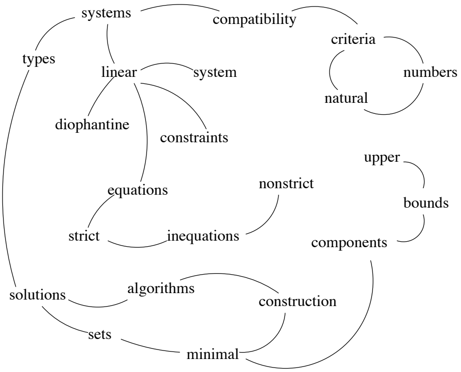
Variations: DegExt uses a directed graph (Litvak et al., 2013), PositionRank uses biased PageRank (Florescu et al., 2017), etc.

Keyword Extraction: Example

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.

Keywords assigned by TextRank:
linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Keywords assigned by human annotators:
linear constraints; linear diophantine equations; minimal generating sets; non–strict inequations; set of natural numbers; strict inequations; upper bounds



Source: Mihalcea et al. (2004a)

Text Summarization

Mihalcea et al. (2004a) also proposed an unsupervised approach for *extractive summarization*.

1. Build a sentence graph:

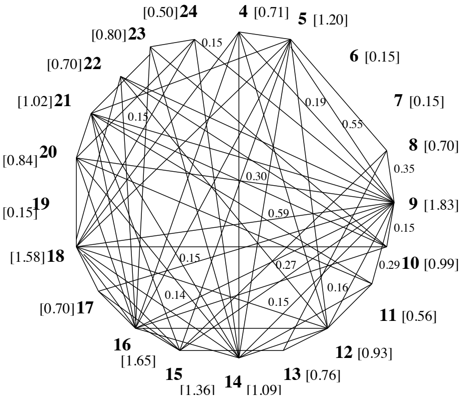
   nodes are sentences

   edges are drawn between “similar” sentences
2. Run PageRank
3. Extract sentences

Variations: sentence clustering (Azadani et al., 2018), biased TextRank (Kazemi et al., 2020), etc.

Text Summarization: Example

3: BC–HurricaineGilbert, 09–11 339
4: BC–Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.



Source: Mihalcea et al. (2004a)

Word Sense Disambiguation

Mihalcea et al. (2004b) proposed an unsupervised approach for word sense disambiguation (WSD) using graphs.

1. Build a text-synset graph:

   nodes are synsets for open class words

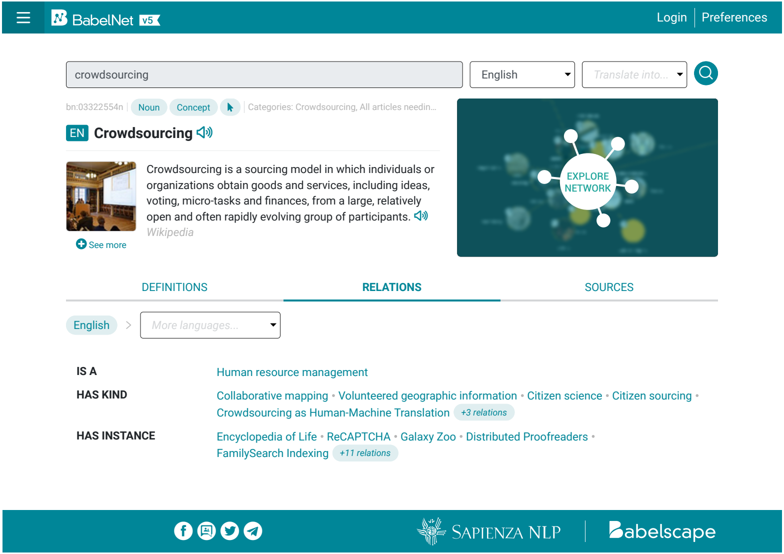
   edges are semantic relations from WordNet (Fellbaum, 1998)
2. Run PageRank
3. Assign word meanings

Variations: densest subgraph heuristic (Moro et al., 2014), personalized PageRank (Agirre et al., 2014) and syntagmatic relations (Scozzafava et al., 2020), etc.

Datasets

- **Stanford Network Analysis Project**, https://snap.stanford.edu/data/
- **Leipzig Corpora Collection** (Goldhahn et al., 2012)
- **Wikipedia** and **Wiktionary** (Zesch et al., 2008; Krizhanovsky et al., 2013)
- **WordNet** (Fellbaum, 1998) and **BabelNet** (Navigli et al., 2012)
- **DBpedia** (Auer et al., 2007)

Datasets: BabelNet



Source: https://babelnet.org/synset?id=bn:03322554n&lang=EN

Datasets: DBpedia

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: Saint Petersburg

An Entity of Type : city, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

Saint Petersburg (Russian: Санкт-Петербург, tr. Sankt-Peterburg, IPA: [ˈsankt pʲɪtʲɪrˈburk] ()), formerly known as Petrograd (Петроград) (1914–1924), then Leningrad (Ленинград) (1924–1991), is a city situated on the Neva River, at the head of the Gulf of Finland on the Baltic Sea. It is Russia's second-largest city after Moscow. With over 5.3 million inhabitants as of 2018, it is the fourth-most populous city in Europe, as well as being the northernmost megalopolis. As an important Russian port on the Baltic Sea, it is governed as a federal city.

Property	Value
dbo:PopulatedPlace/areaTotal	▪ 1439.0
dbo:PopulatedPlace/populationDensity	▪ 3699.31
dbo:abstract	▪ Saint Petersburg (Russian: Санкт-Петербург, tr. Sankt-Peterburg, IPA: [ˈsankt pʲɪtʲɪrˈburk] ()), formerly known as Petrograd (Петроград) (1914–1924), then Leningrad (Ленинград) (1924–1991), is a city situated on the Neva River, at the head of the Gulf of Finland on the Baltic Sea. It is Russia's second-largest city after Moscow. With over 5.3 million inhabitants as of 2018, it is the fourth-most populous city in Europe, as well as being the northernmost megalopolis. As an important Russian port on the Baltic Sea, it is governed as a federal city. The city was founded by Tsar Peter the Great on 27 May [O.S. 16 May] 1703, on the site of a captured Swedish fortress. It served as a capital of the Russian Tsardom and the subsequent Russian Empire from 1713 to 1918 (being replaced by Moscow for a short period of time between 1728 and 1730). After the October Revolution, the Bolsheviks moved their government to Moscow. In modern times, Saint Petersburg is considered the Northern Capital and serves as a home to some federal government bodies such as the Constitutional Court of Russia and the Heraldic Council of the President of the Russian Federation. It is also a seat for the National Library of Russia and a planned location for the Supreme Court of the Russian Federation. The Historic Centre of Saint Petersburg and Related Groups of Monuments constitute a UNESCO World Heritage Site, so it's also referred to as Russia's cultural capital. Saint Petersburg is home to the Hermitage, one of the largest art museums in the world, and the Lakhta Center, the tallest skyscraper in Europe. Many foreign consulates, international corporations, banks and businesses have offices in Saint Petersburg. (en)
dbo:areaTotal	▪ 1439000000.000000 (xsd:double)
dbo:country	▪ dbr:Russia

Source: https://dbpedia.org/page/Saint_Petersburg

Section 6

Conclusion

Conclusion

- Graphs are an extremely powerful representation of the data
- Even the "simple" possibility of selecting the most important nodes reveals great insights
- We have defined a mathematical framework for reasoning about graphs that we will use in the next lectures
- Choose centrality algorithms carefully according to your data assumptions (Boudin, 2013)

Source: Dumlao (2017)

Resources I

Events:

- **TextGraphs**, the Workshop on Graph-Based Algorithms for NLP, http://www.textgraphs.org/

Books:

- Graph Algorithms (Cormen et al., 2022, Chapters 20–25)
- Graph-Based NLP & IR (Mihalcea et al., 2011)
- Structure Discovery in Natural Language (Biemann, 2012)
- The Nature of Complex Networks (Dorogovtsev et al., 2022)

Resources II

Network Analysis Software:

- **Python:** NetworkX (Hagberg et al., 2008), igraph (Csárdi et al., 2006), graph-tool, Snap.py
- **R:** igraph, RBGL
- **Java:** JGraphT (Michail et al., 2020), GraphX (Gonzalez et al., 2014)
- **C/C++:** igraph, Boost Graph Library, SNAP (Leskovec et al., 2016)

Questions?

Contacts

Dr. **Dmitry Ustalov**

https://github.com/dustalov

mailto:dmitry.ustalov@gmail.com

0000-0002-9979-2188

Revision: 5d35748

References I

Agirre E., López de Lacalle O., and Soroa A. (2014). Random Walks for Knowledge-Based Word Sense Disambiguation. *Computational Linguistics*, vol. 40, no. 1, pp. 57–84. DOI: 10.1162/COLI_a_00164.

Auer S. et al. (2007). DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007. Proceedings*. Vol. 4825. Lecture Notes in Computer Science. Berlin and Heidelberg, Germany: Springer Berlin Heidelberg, pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52.

Azadani M. N., Ghadiri N., and Davoodijam E. (2018). Graph-based biomedical text summarization: An itemset mining and sentence clustering approach. *Journal of Biomedical Informatics*, vol. 84, pp. 42–58. DOI: 10.1016/j.jbi.2018.06.005.

Barabási A.-L. and Albert R. (1999). Emergence of Scaling in Random Networks. *Science*, vol. 286, no. 5439, pp. 509–512. DOI: 10.1126/science.286.5439.509.

Bavelas A. (1950). Communication Patterns in Task-Oriented Groups. *The Journal of the Acoustical Society of America*, vol. 22, no. 6, pp. 725–730. DOI: 10.1121/1.1906679.

Biemann C. (2012). Structure Discovery in Natural Language. Theory and Applications of Natural Language Processing. Springer Berlin Heidelberg. ISBN: 978-3-642-25922-7. DOI: 10.1007/978-3-642-25923-4.

Bonacich P. (1987). Power and Centrality: A Family of Measures. *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182. DOI: 10.1086/228631.

Boudin F. (2013). A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. IJCNLP 2013. Nagoya, Japan: Asian Federation of Natural Language Processing, pp. 834–838. URL: https://aclanthology.org/I13-1102.

Brandes U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, vol. 30, no. 2, pp. 136–145. DOI: 10.1016/j.socnet.2007.11.001.

Brin S. and Page L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, vol. 30, no. 1. Proceedings of the Seventh International World Wide Web Conference, pp. 107–117. DOI: 10.1016/S0169-7552(98)00110-X.

Cormen T. H. et al. (2022). Introduction to Algorithms. Fourth Edition. MIT Press. ISBN: 978-0262046305.

Csárdi G. and Nepusz T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, vol. 1695, pp. 1–9. URL: http://www.interjournal.org/manuscript_abstract.php?361100992.

Dijkstra E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271. DOI: 10.1007/BF01386390.

Dorogovtsev S. N. and Mendes J. F. F. (2001). Language as an evolving word web. *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 268, no. 1485, pp. 2603–2606. DOI: 10.1098/rspb.2001.1824.

References II

Dorogovtsev S. N. and Mendes J. F. F. (2022). The Nature of Complex Networks. Oxford, UK: Oxford University Press. ISBN: 9780199695119.

Fellbaum C. (1998). WordNet: An Electronic Database. Massachusetts, MA, USA: MIT Press. ISBN: 978-0-262-06197-1. DOI: 10.7551/mitpress/7287.001.0001.

Florescu C. and Caragea C. (2017). PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL 2017. Vancouver, BC, Canada: Association for Computational Linguistics, pp. 1105–1115. DOI: 10.18653/v1/P17-1102.

Freeman L. C. (1977). A Set of Measures of Centrality Based on Betweenness. *Sociometry*, vol. 40, no. 1, pp. 35–41. DOI: 10.2307/3033543.

Gallardo P. F. (2007). Google's secret and Linear Algebra. *EMS Newsletter*, vol. 63, pp. 10–15. URL: https://www.ems-ph.org/journals/newsletter/pdf/2007-03-63.pdf.

Goldhahn D., Eckart T., and Quasthoff U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. *Proceedings of the Eight International Conference on Language Resources and Evaluation*. LREC 2012. Istanbul, Turkey: European Language Resources Association (ELRA), pp. 759–765. URL: http://www.lrec-conf.org/proceedings/lrec2012/summaries/327.html.

Gonzalez J. E. et al. (2014). GraphX: Graph Processing in a Distributed Dataflow Framework. *11th USENIX Symposium on Operating Systems Design and Implementation*. OSDI '14. Broomfield, CO, USA: USENIX Association, pp. 599–613. URL: https://www.usenix.org/conference/osdi14/technical-sessions/presentation/gonzalez.

Hagberg A. A., Schult D. A., and Swart P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the 7th Python in Science Conference*. SciPy2008. Pasadena, CA, USA, pp. 11–15. URL: https://conference.scipy.org/proceedings/SciPy2008/paper_2.

Johnson D. B. (1977). Efficient Algorithms for Shortest Paths in Sparse Networks. *Journal of the ACM*, vol. 24, no. 1, pp. 1–13. DOI: 10.1145/321992.321993.

Kapustin V. and Jamsen A. (2007). Vertex Degree Distribution for the Graph of Word Co-Occurrences in Russian. *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*. TextGraphs-2. Rochester, NY, USA: Association for Computational Linguistics, pp. 89–92. URL: https://aclanthology.org/W07-0213.

Kazemi A., Pérez-Rosas V., and Mihalcea R. (2020). Biased TextRank: Unsupervised Graph-Based Content Extraction. *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 1642–1652. DOI: 10.18653/v1/2020.coling-main.144.

Kleinberg J. M. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, vol. 46, no. 5, pp. 604–632. DOI: 10.1145/324133.324140.

References III

Krizhanovsky A. A. and Smirnov A. V. (2013). An approach to automated construction of a general-purpose lexical ontology based on Wiktionary. *Journal of Computer and Systems Sciences International*, vol. 52, no. 2, pp. 215–225. DOI: 10.1134/S1064230713020068.

Leskovec J. and Sosič R. (2016). SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 1, 1:1–1:20. DOI: 10.1145/2898361.

Litvak M., Last M., and Kandel A. (2013). DegExt: a language-independent keyphrase extractor. *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 3, pp. 377–387. DOI: 10.1007/s12652-012-0109-z.

von Luxburg U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, vol. 17, no. 4, pp. 395–416. DOI: 10.1007/s11222-007-9033-z.

Michail D. et al. (2020). JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Transactions on Mathematical Software*, vol. 46, no. 2, 16:1–16:29. DOI: 10.1145/3381449.

Mihalcea R. and Radev D. (2011). Graph-Based Natural Language Processing and Information Retrieval. Cambridge University Press. ISBN: 978-0-521-89613-9. DOI: 10.1017/CBO9780511976247.

Mihalcea R. and Tarau P. (2004a). TextRank: Bringing Order into Text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2004. Barcelona, Spain: Association for Computational Linguistics, pp. 404–411. URL: https://aclanthology.org/W04-3252.

Mihalcea R., Tarau P., and Figa E. (2004b). PageRank on Semantic Networks, with Application to Word Sense Disambiguation. *Proceedings of the 20th International Conference on Computational Linguistics*. COLING 2004. Geneva, Switzerland: COLING, pp. 1126–1132. DOI: 10.3115/1220355.1220517.

Moro A., Raganato A., and Navigli R. (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244. DOI: 10.1162/tacl_a_00179. the content is released under a CC BY-NC-ND license, but used with the permission of The MIT Press.

Navigli R. and Ponzetto S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, vol. 193, pp. 217–250. DOI: 10.1016/j.artint.2012.07.001.

Scozzafava F. et al. (2020). Personalized PageRank with Syntagmatic Information for Multilingual Word Sense Disambiguation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. ACL 2020. Online: Association for Computational Linguistics, pp. 37–46. DOI: 10.18653/v1/2020.acl-demos.6.

Steyvers M. and Tenenbaum J. B. (2005). The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. *Cognitive Science*, vol. 29, no. 1, pp. 41–78. DOI: 10.1207/s15516709cog2901_3.

References IV

Ustalov D. et al. (2019). Watset: Local-Global Graph Clustering with Applications in Sense and Frame Induction. *Computational Linguistics*, vol. 45, no. 3, pp. 423–479. DOI: 10.1162/COLI_a_00354.

Zesch T., Müller C., and Gurevych I. (2008). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. *Proceedings of the 6th International Conference on Language Resources and Evaluation*. LREC 2008. Marrakech, Morocco: European Language Resources Association (ELRA), pp. 1646–1652. URL: http://www.lrec-conf.org/proceedings/lrec2008/summaries/420.html.

Supplementary Media I

345Kai and Stannered (November 25, 2007). Numeric examples of PageRank values in a small graph with a damping factor of 0.85. Wikimedia Commons. URL: https://commons.wikimedia.org/wiki/File:PageRanks-Example.svg. Licensed under Public Domain.

Adamovich O. (September 3, 2015). Girls Whispering Best Friends. Pixabay. URL: https://pixabay.com/images/id-914823/. Licensed under Pixabay License.

Amos E. (December 19, 2011). The Vectrex video game console, shown with controller. Wikimedia Commons. URL: https://commons.wikimedia.org/wiki/File:Vectrex-Console-Set.jpg. Licensed under CC BY-SA 3.0, used with author's permission.

Dumlao N. (November 21, 2017). two person pouring coffee with piled cups photo. Unsplash. URL: https://unsplash.com/photos/eksqjXTLpak. Licensed under Unsplash License.

Free-Photos (August 9, 2016). Person Mountain Top Achieve. Pixabay. URL: https://pixabay.com/images/id-1245959/. Licensed under Pixabay License.

Merrill B. (July 24, 2014). Pedestrians People Busy. Pixabay. URL: https://pixabay.com/images/id-400811/. Licensed under Pixabay License.

rawpixel (April 18, 2017). Calm Freedom Location. Pixabay. URL: https://pixabay.com/images/id-2218409/. Licensed under Pixabay License.

rawpixel (June 23, 2018). Agreement Business Businessman. Pixabay. URL: https://pixabay.com/images/id-3489902/. Licensed under Pixabay License.

Tama66 (October 1, 2018). Saint Petersburg Historic Center. Pixabay. URL: https://pixabay.com/images/id-3714288/. Licensed under Pixabay License.