Towards a format for describing networks

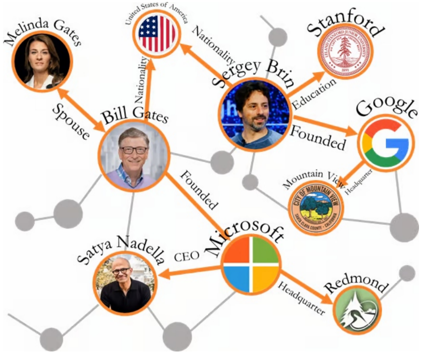
Vladimir Batagelj, Tomaž Pisanski, Iztok Savnik. Ana Slavec, Nino Bašić

IMFM Ljubljana, UP IAM Koper, UL FMF Ljubljana

1358+1359. Sredin seminar

on Zoom, March 12 and 19, 2025

Outline



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
Current version of slides (March 19, 2025 at 07 : 22): slides PDF
`https://github.com/bavla/netsJSON`

Unit identification

How many books?

name
(in context)
classes

Computer-assisted text analysis

An often used way to obtain networks is the *computer-assisted text analysis* (CaTA). *Terms* considered in TA are collected in a *dictionary* (it can be fixed in advance, or built dynamically). The main two problems with terms are

- *equivalence* – different words representing the same term, and
- *ambiguity* – same word representing different terms.

Because of these the *coding* – transformation of raw text data into formal *description* – is done often manually or semiautomaticly.

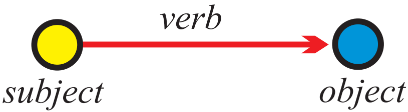
We assume that unit identification (entity recognition) assigns a unique identifier (ID) to each unit. For some types of units, such IDs are standardized: ISO 3166-1 alpha-2 two-letter country codes, ISO 9362 Bank Identifier Codes (BIC), ORCID Open Researcher and Contributor ID, ISSN International Standard Serial Number, DOI Digital Object Identifier, URI Uniform Resource Identifier, etc.
In data displays, often IDs are replaced by corresponding (short) labels/names.

...approaches to CaTA

As *units* of TA we usually consider clauses, statements, paragraphs, news, messages, ...

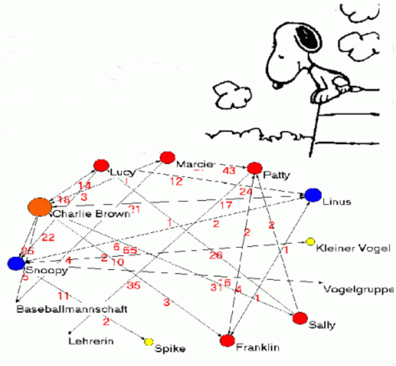
In thematic TA the units are coded as rectangular matrix *Text units* $\times$ *Concepts* which can be considered as a two-mode network.

In semantic TA the units (often clauses) are encoded according to the S-V-O (*Subject-Verb-Object*) model or its improvements.



This coding can be directly considered as network with *Subjects* $\cup$ *Objects* as nodes and links (arcs) labeled with *Verbs*.

Networks



Alexandra Schuler/ Marion Laging-Glaser:
Analyse von Snoopy Comics

A *network* is based on two sets – set of *nodes* (vertices), that represent the selected *units*, and set of *links* (lines), that represent *ties* between units. They determine a *graph*. A link can be *directed* – an *arc*, or *undirected* – an *edge*.

Additional data about nodes or links can be known – their *properties* (attributes). For example: name/label, type, value, . . .

Network = Graph + Data

The data can be measured or computed.

Networks / Formally

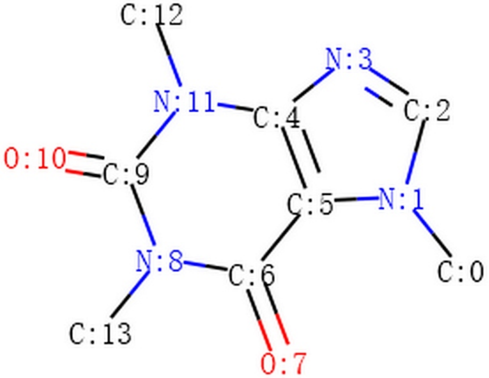
A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of:

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{A}$ is the set of arcs, $\mathcal{E}$ is the set of edges, and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of links.
  $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ *node value functions* / properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ *link value functions* / weights: $w : \mathcal{L} \to B$

Additional information/data about values:

- How can we compute with values – algebraic structures semigroup, monoid, group, semiring, etc.
- Properties of values – in a molecular graph an atom is assigned to each node; properties of relevant atoms are such additional data.

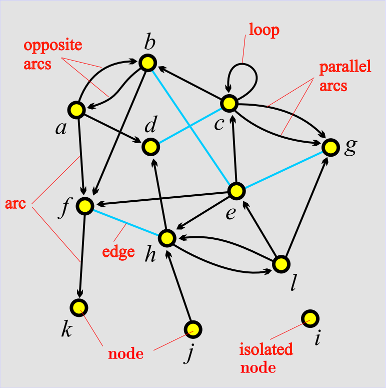
Molecular graph



Atomic Symbol	Valency
C	4
H	1
O	2
N	3

WWW, Carbon, Hydrogen, Oxygen, Nitrogen

Graph



unit, actor – node, vertex
tie, line – link, edge, arc

arc = directed link, (a, d)
a is the *initial* node,
d is the *terminal* node.

edge = undirected link, $(c : d)$
c and d are *end* nodes.

Types of networks

Besides ordinary (directed, undirected, mixed) networks some extended types of networks are also used:

- *2-mode networks*, bipartite (valued) graphs – networks between two disjoint sets of nodes.
- *multi-relational networks*.
- *linked networks* and *collections of networks*.
- *temporal networks*, dynamic graphs – networks changing over time.
- specialized networks: representation of genealogies as *p-graphs*; *Petri's nets*, . . .

The network (input) file formats should provide means to express all these types of networks. All interesting data should be recorded (respecting privacy).

Two-mode networks

In a *two-mode* network $\mathcal{N} = ((\mathcal{U}, \mathcal{V}), \mathcal{L}, \mathcal{P}, \mathcal{W})$ the set of nodes consists of two disjoint sets of nodes $\mathcal{U}$ and $\mathcal{V}$, and all the links from $\mathcal{L}$ have one endnode in $\mathcal{U}$ and the other in $\mathcal{V}$. Often also a *weight* $w : \mathcal{L} \to \mathbb{R} \in \mathcal{W}$ is given; if not, we assume $w(u, v) = 1$ for all $(u, v) \in \mathcal{L}$.
A two-mode network can also be described by a rectangular matrix $\mathbf{A} = [a_{uv}]_{\mathcal{U}\times\mathcal{V}}$.

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

Examples: (persons, societies, years of membership), (buyers/consumers, goods, quantity), (parlamentarians, problems, positive vote), (persons, journals, reading).
Authors and works.

Deep South

Classical example of two-mode network are the Southern women (Davis 1941).

Davis.paj. Freeman's overview.

NAMES OF PARTICIPANTS OF GROUP I	CODE NUMBERS AND DATES OF SOCIAL EVENTS REPORTED IN *Old City Herald*													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson	×	×	×	×	×	×		×	×					
2. Miss Laura Mandeville	×	×	×		×	×	×	×						
3. Miss Theresa Anderson		×	×	×	×	×	×	×	×					
4. Miss Brenda Rogers	×		×	×	×	×	×	×						
5. Miss Charlotte McDowd			×	×	×		×							
6. Miss Frances Anderson			×		×	×		×						
7. Miss Eleanor Nye					×	×	×	×						
8. Miss Pearl Oglethorpe						×		×	×					
9. Miss Ruth DeSand					×		×	×	×					
10. Miss Verne Sanderson							×	×	×			×		
11. Miss Myra Liddell								×	×	×		×		
12. Miss Katherine Rogers								×	×	×		×	×	×
13. Mrs. Sylvia Avondale							×	×	×	×		×	×	×
14. Mrs. Nora Fayette						×	×		×	×	×	×	×	×
15. Mrs. Helen Lloyd							×	×		×	×	×		
16. Mrs. Dorothy Murchison								×	×					
17. Mrs. Olivia Carleton									×		×			
18. Mrs. Flora Price									×		×			

Multi-relational networks

A *multi-relational network* is denoted by

$$\mathcal{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$$

and contains different relations $\mathcal{L}_i$ (sets of links) over the same set of nodes. Also the weights from $\mathcal{W}$ are defined on different relations or their union.

Examples of such networks are: Transportation system in a city (stations, lines); WordNet (words, semantic relations: synonymy, antonymy, hyponymy, meronymy,...), KEDS networks (states, relations between states: Visit, Ask information, Warn, Expel person, ...), ...

Linked networks and collections of networks

In a *linked* or *multimodal* network

$$\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_j), (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$$

the set of nodes $\mathcal{V}$ is partitioned into subsets (*modes*) $\mathcal{V}_i$, $\mathcal{L}_s \subseteq \mathcal{V}_p \times \mathcal{V}_q$, and properties and weights are usually partial functions.

A set of networks $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_k\}$ in which each network shares a set of nodes with some other network is called a *collection* of networks.

Bibliographical information is usually represented as a collection of bibliographical networks $\{WA, Cite, WK, WC, WI, \ldots\}$.

Multilevel networks

A *multilevel network* organizes nodes into hierarchical levels, where each level represents a different scale or granularity of interaction or connectivity. It is a special case of a linked network.

Key characteristics of multilevel networks are: hierarchical levels, inter-level links, scale-specific dynamics, and modularity.

Each level may have its unique dynamics, rules, or behaviors. For example, in a biological network, one level might represent protein-protein interactions, while another level represents cellular interactions. They often exhibit modularity, where nodes within a level are more densely linked to each other than to nodes in other levels. This modularity can help in understanding the functional or structural organization of the network.

Example: a multilevel network in the context of a university: Level 1: Individual students and faculty members. Level 2: Departments or academic units. Level 3: The entire university.

Temporal network

In a *temporal network* the presence/activity of node/link can change through time.

Temporal network

$$\mathcal{N}_T = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}, T)$$

is obtained if the *time* T is attached to an ordinary network. T is a set of *time points* $t \in T$.

In temporal network nodes $v \in \mathcal{V}$ and links $l \in \mathcal{L}$ are not necessarily present or active in all time points. If a link $l(u, v)$ is active in time point t then also its endnodes u and v should be active in time t.

We will denote the network consisting of links and nodes active in time $t \in T$ by $\mathcal{N}(t)$ and call it a *time slice* in time point t.

`Pajek` supports two types of descriptions of temporal networks based on *presence* and on *events*.

Temporal networks – events

Event	Explanation
TI t	initial events – following events happen when time point t starts
TE t	end events – following events happen when time point t is finished
AV v n s	add vertex v with label n and properties s
HV v	hide node v
SV v	show node v
DV v	delete node v
AA u v s	add arc (u, v) with properties s
HA u v	hide arc (u, v)
SA u v	show arc (u, v)
DA u v	delete arc (u, v)
AE u v s	add edge $(u : v)$ with properties s
HE u v	hide edge $(u : v)$
SE u v	show edge $(u : v)$
DE u v	delete edge $(u : v)$
CV v s	change property of node v to s
CA u v s	change property of arc (u, v) to s
CE u v s	change property of edge $(u : v)$ to s
CT u v	change (un)directedness of link (u, v)
CD u v	change direction of arc (u, v)
PE u v s	replace pair of arcs (u, v) and (v, u) by single edge $(u : v)$ with properties s
AP u v s	add pair of arcs (u, v) and (v, u) with properties s
DP u v	delete pair of arcs (u, v) and (v, u)
EP u v s	replace edge $(u : v)$ by pair of arcs (u, v) and (v, u) with properties s

s can be empty.
In case of parallel links : k denotes the k-th link – `HE:3 14 37` hides the third edge linking nodes 14 and 37.

```
*Vertices 3
*Events
TI 1
AV 2 "b"
TE 3
HV 2
TI 4
AV 3 "e"
TI 5
AV 1 "a"
TI 6
AE 1 3 1
TI 7
SV 2
AE 1 2 1
TE 7
DE 1 2
DV 2
TE 8
DE 1 3
TE 10
HV 1
TI 12
SV 1
TE 14
DV 1
```

Time.tim. Friends.tim.

Multi-relational temporal network – KEDS/WEIS


```
% Recoded by WEISmonths, Sun Nov 28 21:57:00 2004
% from http://www.ku.edu/~keds/data.dir/balk.html
*vertices 325
1 "AFG" [1-*]
2 "AFR" [1-*]
3 "ALB" [1-*]
4 "ALBMED" [1-*]
5 "ALG" [1-*]
 ...
318 "YUGGOV" [1-*]
319 "YUGMAC" [1-*]
320 "YUGMED" [1-*]
321 "YUGMTN" [1-*]
322 "YUGSER" [1-*]
323 "ZAI" [1-*]
324 "ZAM" [1-*]
325 "ZIM" [1-*]
*arcs :0 "*** ABANDONED"
*arcs :10 "YIELD"
*arcs :11 "SURRENDER"
*arcs :12 "RETREAT"
 ...
*arcs :223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
*arcs
224: 314 153 1 [4]          890402  YUG     KSV     224  (RIOT)  RIOT-TORN
212: 314 83 1 [4]           890404  YUG     ETHALB  212  (ARREST PERSON) ALB ETHNIC JAILED
224: 3 83 1 [4]             890407  ALB     ETHALB  224  (RIOT)  RIOTS
123: 83 153 1 [4]           890408  ETHALB  KSV     123  (INVESTIGATE)   PROBING
 ...
42: 105 63 1 [175]          030731  GER     CYP     042  (ENDORSE)       GAVE SUPPORT
212: 295 35 1 [175]         030731  UNWCT   BOSSER  212  (ARREST PERSON) SENTENCED TO PRIS
43: 306 87 1 [175]          030731  VAT     EUR     043  (RALLY) RALLIED
13: 295 35 1 [175]          030731  UNWCT   BOSSER  013  (RETRACT)       CLEARED
121: 295 22 1 [175]         030731  UNWCT   BAL     121  (CRITICIZE)     CHARGES
122: 246 295 1 [175]        030731  SER     UNWCT   122  (DENIGRATE)     TESTIFIED
121: 35 295 1 [175]         030731  BOSSER  UNWCT   121  (CRITICIZE)     ACCUSED
```

Kansas Event Data System *KEDS*

Temporal quantities

We introduce a notion of a *temporal quantity*

$$a(t) = \begin{cases} a'(t) & t \in T_a \\ \maltese & t \in \mathcal{T} \setminus T_a \end{cases}$$

where T_a is the *activity time set* of a and $a'(t)$ is the value of a in an instant $t \in T_a$, and $\maltese$ denotes the value *undefined*.

We assume that the values of temporal quantities belong to a set A which is a *semiring* $(A, +, \cdot, 0, 1)$ for binary operations $+ : A \times A \to A$ and $\cdot : A \times A \to A$.
Let $A_\maltese(\mathcal{T})$ denote the set of all temporal quantities over $A_\maltese$ in time $\mathcal{T}$. To extend the operations to networks and their matrices we first define the *sum* (parallel links) $a + b$ as

$$(a+b)(t) = a(t) + b(t) \quad \text{and} \quad T_{a+b} = T_a \cup T_b.$$

The *product* (sequential links) $a \cdot b$ is defined as

$$(a \cdot b)(t) = a(t) \cdot b(t) \quad \text{and} \quad T_{a \cdot b} = T_a \cap T_b.$$

Temporal quantities

Let us define TQs $\mathbf{0}$ and $\mathbf{1}$ with requirements $\mathbf{0}(t) = \maltese$ and $\mathbf{1}(t) = 1$ for all $t \in \mathcal{T}$. Again, the structure $(A_\maltese(\mathcal{T}), +, \cdot, \mathbf{0}, \mathbf{1})$ is a semiring.

To produce a software support for computation with TQs we limit it to TQs that can be described as a sequence of disjoint time intervals with a constant value

$$a = [(s_i, f_i, v_i)]_{i \in 1..k}$$

where s_i is the starting time and f_i the finishing time of the i-th time interval $[s_i, f_i)$, $s_i < f_i$ and $f_i \leq s_{i+1}$, and v_i is the value of a on this interval (over combinatorial semiring). Outside the intervals the value of TQ a is undedined, $\maltese$.

For example

$$a = [(1,5,2), (6,8,1), (11,12,3), (14,16,2), (17,18,5), (19,20,1)]$$

In a temporal network some node properties and/or link weights are assigning TQs.

Nets and NetsJSON

For dealing with networks with properties with structured values (for example, temporal quantities) we are developing a Python package Nets [**nets**].

For describing temporal networks we initially, extending Pajek format, defined and used the Ianus format.

In 2015 we started to develop a new format based on JSON – we named it netJSON. On February 26, 2019 the format was renamed to NetsJSON because of the collision with http://netjson.org/rfc.html.

NetsJSON has two versions: a *basic* and a *general* version. Current implementation of the Nets / TQ library supports only the basic version. Nets.

Besides for a *description* of networks with structured values, NetsJSON should *envelope* (most of) existing network description formats [**bodlaj**] (archiving, conversion) and provide input data for D3.js *visualization*s.

Description of networks using a spreadsheet

How to describe a network $\mathcal{N}$? In principle the answer is simple – we list its components $\mathcal{V}$, $\mathcal{L}$, $\mathcal{P}$, and $\mathcal{W}$.

The simplest way is to describe a network $\mathcal{N}$ by providing $(\mathcal{V}, \mathcal{P})$ and $(\mathcal{L}, \mathcal{W})$ in a form of two tables.

As an example, let us describe a part of network determined by the following works:
Generalized blockmodeling, Clustering with relational constraint, Partitioning signed social networks, The Strength of Weak Ties

There are nodes of different types (modes): persons, papers, books, series, journals, publishers; and different relations among them: author_of, editor_of, contained_in, cites, published_by.

Both tables are often maintained in Excel. They can be exported as text in CSV (Comma Separated Values) format.

Factorization and description of large networks

To save space and improve the computing efficiency we often replace values of categorical variables with integers. In R this encoding is called a *factorization*.

We enumerate all possible values of a given categorical variable (coding table) and afterwards replace each its value by the corresponding index in the coding table.

This approach is used in most programs dealing with large networks. Unfortunately the coding table is often a kind of meta-data.

Knowledge graph

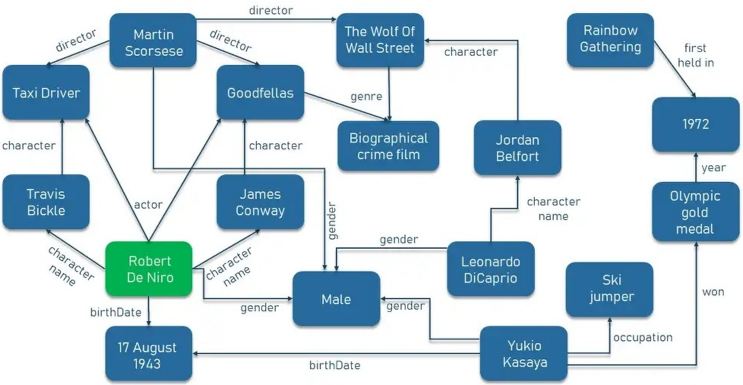
A knowledge graph is a structured representation of knowledge that captures entities, relationships, and attributes in a graph-based format.

A knowledge graph is a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities [1].

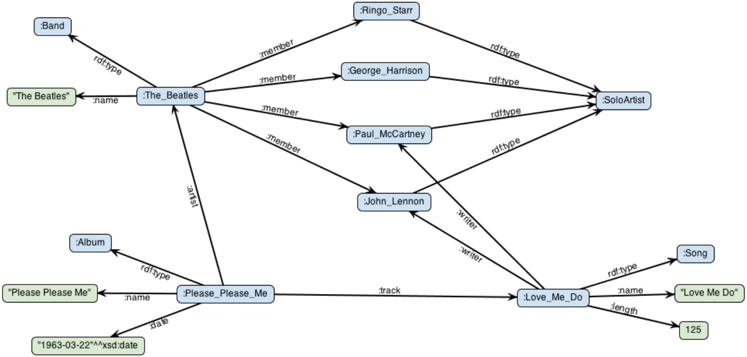
A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge [2].

Knowledge graphs are widely used in applications like semantic search, recommendation systems, natural language processing, and artificial intelligence. They are often built using standards like RDF (Resource Description Framework) and queried using languages like SPARQL.

Knowledge graph



Knowledge graph



Knowledge graphs formally

Formally, a *knowledge graph* can be defined as a directed, labeled multigraph:

$$G = (E, R, A, \mathcal{L}, \mathcal{F})$$

Where:

- E is a set of *entities* (nodes), representing real-world objects, concepts, or instances.
- R is a set of *relationships* (arcs), representing directed links between entities.
- A is a set of *attributes*, representing properties or characteristics of entities or relationships.
- $\mathcal{L}$ is a set of *labels*, which provide semantic meaning to entities, relationships, and attributes.
- $\mathcal{F}$ is a set of *facts*, where each fact is a triple of the form (e_1, r, e_2) or (e, a, v), with: $e, e_1, e_2 \in E$, $r \in R$, $a \in A$, and v (value, which can be a literal or another entity).

. . . Knowledge graphs formally

Key characteristics

- **Graph Structure**: Entities are nodes, and relationships are edges connecting them.
- **Semantic Richness**: Labels and attributes provide meaning to the entities and relationships.
- **Interconnectedness**: Entities are linked through relationships, enabling traversal and inference.
- **Extensibility**: New entities, relationships, and attributes can be added dynamically.

. . . Knowledge graphs

Simple example

In a knowledge graph about people and organizations:

- Entities: "Albert Einstein," "Princeton University."
- Relationships: "worked_at" (Albert Einstein → Princeton University).
- Attributes: "birth_date" (Albert Einstein → "1879-03-14").

Knowledge graphs

$E \leftrightarrow V$

$R \leftrightarrow L$ – partition of R with respect to assigned labels = multirelational network

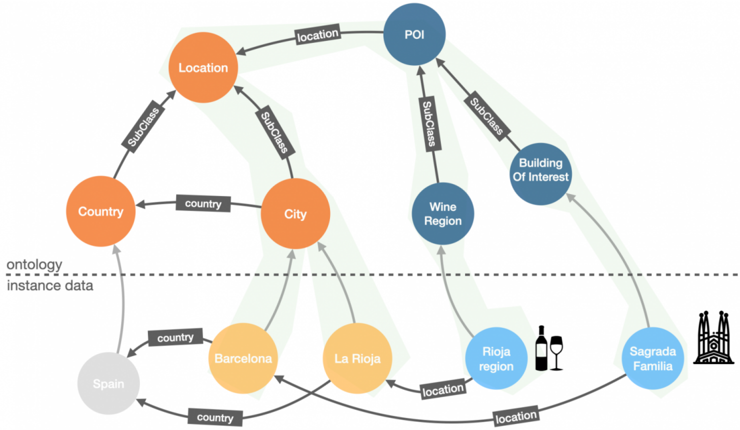
$(E, a, X) \leftrightarrow a : E \to X$

. . . Knowledge graphs

History
Introduction to RDF
RDF 1.1 Resource Description Framework; OWL 2 Web Ontology Language; URI; URN; Turtle; n-ary Relations
RDF 1.2 draft;

. . . Knowledge graphs

Example



Recommendations for sharing network data and materials

In 2023, the International Network for Social Network Analysis (INSNA) requested that Zachary Neal form a working group to develop **recommendations for sharing network data and materials**. They were published in *Network Science* in 2024 [3] accompanied with the *Endorsement page* [4].

It would be useful to have a common "archiving/intermediate" format that can describe (almost) all networks. It is easy to write converters from this format to a selected format or corresponding network reading procedures.

There are many tools and programs for network analysis UCINET, Pajek, Gephi, NetMiner, Cytoscape, NodeXL, E-Net, Tulip, PUCK, GraphViz, SocNetV, Kumu, Polinode, etc.

Software support for network analysis

Programmers can use network analysis packages/libraries in different programming languages

- **Python:** NetworkX, igraph, Snap.py, graph-tool, NetworKit, PyGraphistry, Nets, cdlib, node2vec, DGL, PyG, Tulip, PyVis,
- **R:** igraph, statnet, sna, qgraph, RSiena, tnet, multiplex, NetSim, influenceR, tidygraph, intergraph, netUtils, ggraph, networkD3, visNetwork, DiagrammeR, graphlayouts, ndtv,
- **Julia:** LightGraphs, Graphs, MetaGraphs, SimpleWeightedGraphs, Erdos, MultilayerGraphs, GraphDataFrameBridge, GraphIO, NetworkDynamics, TemporalGPs, EcologicalNetwork, CommunityDetection, GraphPlot, NetworkLayout,
- **C++:** Boost Graph Library, igraph, SNAP, NetworKit, NetworkX, Graph-tool, GraphBLAS, Lemon Graph Library, GraphHopper, Gelly, Tulip, OGDF,
- etc.

. . . Software support for network analysis

They are supporting different network description formats: CSV, UCINET DL, Pajek NET, Gephi GEXF, GDF, GML, GraphML, GraphViz DOT, Tulip TPL, Netdraw VNA, Spreadsheet, etc. [5].
In addition, network data appears in several application areas such as chemistry and genealogy. There are many formats for describing molecular graphs: Molfile, SDF, CML, PDB, XYZ, CIF, FASTA, CDX, CDXML, JCAMP-DX, SMILES, InChI, and others. The most widely used format for genealogical data exchange, GEDCOM is a plain text file format that stores information about individuals, families, events, and sources. It has several derivatives. It is considered an exchange format between various genealogy programs, which are often based on their own format. Some of the most well-known are Ancestry Tree Files, Family Tree Maker, Legacy Family Tree, RootsMagic, OpenGen Alliance, Open Archives Format, FamilySearch JSON, Gramps XML, TEI, PROGEN, Webtrees, PAF.
Tomaž Pisanski et al. – Vega

Network representations

There are three commonly used file representations of graphs and networks.

- **Link list (with weights)** This is the most commonly used and expressively most flexible representation.
- **Matrix representation** It is often found in older sources. It is suitable for describing smaller, denser simple networks. We lose the distinction between directed and undirected and multiple links. For larger networks, which are usually sparse, it requires a lot of space – most matrix entries have the value 0.
- **Neighbor sets** This representation is very economical but only useful for networks without link properties.

Repositories of graph and network data

- ICON – Colorado Index of Complex Networks
- UCINET datasets
- Pajek networks
- UCI Network Data Repository
- CASOS – **C**omputational **A**nalysis of **S**ocial and **O**rganizational **S**ystems
- SNAP – **S**tanford **N**etwork **A**nalysis **P**latform
- KONECT – **Ko**blenz **Ne**twork **C**ollec**t**ion
- Netzschleuder – network catalogue, repository and centrifuge
- Schochastics network data
- Network Repository
- Siena data sets

. . . Repositories of graph and network data

- The House of Graphs
- Encyclopedia of Graphs
- Datasets of Highly Symmetric Objects
- Awesome Network Analysis datasets
- Kinsources
- RCSB Protein Data Bank

Description of networks using a spreadsheet

How to describe a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$? In principle the answer is simple – we list its components $\mathcal{V}$, $\mathcal{L}$, $\mathcal{P}$, and $\mathcal{W}$. The simplest way is to describe a network $\mathcal{N}$ by providing $(\mathcal{V}, \mathcal{P})$ and $(\mathcal{L}, \mathcal{W})$ in a form of two tables.

As an example, let us describe a part of the network determined by the bibliographical data about the following works: Generalized blockmodeling, Clustering with relational constraint, Partitioning signed social networks, The Strength of Weak Ties.
There are nodes of different types (modes): persons, papers, books, series, journals, publishers; and different relations among them: author_of, editor_of, contained_in, cites, published_by. For some types of nodes additional properties are known: sex, year, volume, number, first and last page, etc.
Both tables are often maintained in Excel. They can be exported as text in CSV (Comma Separated Values) format.

bibNodes.csv

```
name;mode;country;sex;year;vol;num;fPage;lPage;x;y
"Batagelj, Vladimir";person;SI;m;;;;;;809.1;653.7
"Doreian, Patrick";person;US;m;;;;;;358.5;679.1
"Ferligoj, Anuška";person;SI;f;;;;;;619.5;680.7
"Granovetter, Mark";person;US;m;;;;;;145.6;660.5
"Moustaki, Irini";person;UK;f;;;;;;783.0;228.0
"Mrvar, Andrej";person;SI;m;;;;;;478.0;630.1
"Clustering with relational constraint";paper;;;1982;47;;413;426;684.1;3
"The Strength of Weak Ties";paper;;;1973;78;6;1360;1380;111.3;329.4
"Partitioning signed social networks";paper;;;2009;31;1;1;11;408.0;337.8
"Generalized Blockmodeling";book;;;2005;24;;1;385;533.0;445.9
"Psychometrika";journal;;;;;;;;741.8;086.1
"Social Networks";journal;;;;;;;;321.4;236.5
"The American Journal of Sociology";journal;;;;;;;;111.3;168.9
"Structural Analysis in the Social Sciences";series;;;;;;;;310.4;082.8
"Cambridge University Press";publisher;UK;;;;;;;534.3;238.2
"Springer";publisher;US;;;;;;;884.6;174.0
```

bibNodes.csv

In large networks, we split a network into some subnetworks – a collection, to avoid the empty cells.

bibLinks.csv

```
from;relation;to
"Batagelj, Vladimir";authorOf;"Generalized Blockmodeling"
"Doreian, Patrick";authorOf;"Generalized Blockmodeling"
"Ferligoj, Anuška";authorOf;"Generalized Blockmodeling"
"Batagelj, Vladimir";authorOf;"Clustering with relational constraint"
"Ferligoj, Anuška";authorOf;"Clustering with relational constraint"
"Granovetter, Mark";authorOf;"The Strength of Weak Ties"
"Granovetter, Mark";editorOf;"Structural Analysis in the Social Sciences
"Doreian, Patrick";authorOf;"Partitioning signed social networks"
"Mrvar, Andrej";authorOf;"Partitioning signed social networks"
"Moustaki, Irini";editorOf;"Psychometrika"
"Doreian, Patrick";editorOf;"Social Networks"
"Generalized Blockmodeling";containedIn;"Structural Analysis in the Soci
"Clustering with relational constraint";containedIn;"Psychometrika"
"The Strength of Weak Ties";containedIn;"The American Journal of Sociolo
"Partitioning signed social networks";containedIn;"Social Networks"
"Partitioning signed social networks";cites;"Generalized Blockmodeling"
"Generalized Blockmodeling";cites;"Clustering with relational constraint
"Structural Analysis in the Social Sciences";publishedBy;"Cambridge Univ
"Psychometrika";publishedBy;"Springer"
```

bibLinks.csv

Factorization and description of large networks

To save space and improve computing efficiency we often replace values of categorical variables with integers. In R this encoding is called a *factorization*.

We enumerate all possible values of a given categorical variable (coding table) and afterward replace each value with the corresponding index in the coding table. Since node labels/IDs can be considered a categorical variable, factorization is usually applied also on them.

This approach is used in most programs dealing with large networks. Unfortunately, the coding table is often considered as a kind of meta-data and is omitted from the description.
Be careful, in data analysis, indices start with 1, but real computer scientists start counting from 0.
Using a short program in R we transform both tables into Pajek files: a network file `bib.net` and partition files `bibMode.clu` and `bibSex.clu`. All the files related to the bibliographic example are available at GitHub/Bavla [6].

CSV2Pajek.R

```
# transforming CSV file to Pajek files
# by Vladimir Batagelj, June 2016
# setwd("C:/Users/batagelj/work/Python/graph/SVG/EUSN")
# colC <- c(rep("character",4),rep("numeric",7)); nas=c("","NA","NaN")
colC <- c(rep("character",4),rep("numeric",5)); nas=c("","NA","NaN")
nodes <- read.csv2("bibNodes.csv",encoding='UTF-8',colClasses=colC,na.strings=nas)
n <- nrow(nodes); M <- factor(nodes$mode); S <- factor(nodes$sex)
mod <- levels(M); sx <- levels(S); S <- as.numeric(S); S[is.na(S)] <- 0
links <- read.csv2("bibLinks.csv",encoding='UTF-8',colClasses="character")
F <- factor(links$from,levels=nodes$name,ordered=TRUE)
T <- factor(links$to,levels=nodes$name,ordered=TRUE)
R <- factor(links$relation); rel <- levels(R)
net <- file("bib.net","w"); cat('*vertices ',n,'\n',file=net)
clu <- file("bibMode.clu","w"); sex <- file("bibSex.clu","w")
cat('%',file=clu); cat('%',file=sex)
for(i in 1:length(mod)) cat(' ',i,mod[i],file=clu)
cat('\n*vertices ',n,'\n',file=clu)
for(i in 1:length(sx)) cat(' ',i,sx[i],file=sex)
cat('\n*vertices ',n,'\n',file=sex)
for(v in 1:n) {
  cat(v,' "',nodes$name[v],'"\n',sep='',file=net);
  cat(M[v],'\n',file=clu); cat(S[v],'\n',file=sex)
}
for(r in 1:length(rel)) cat('*arcs :',r,' "',rel[r],'"\n',sep='',file=net)
cat('*arcs\n',file=net)
for(a in 1:nrow(links))
  cat(R[a],': ',F[a],' ',T[a],' 1 l "',rel[R[a]],'"\n',sep='',file=net)
close(net); close(clu); close(sex)
```

CSV2Pajek.R

bib.net

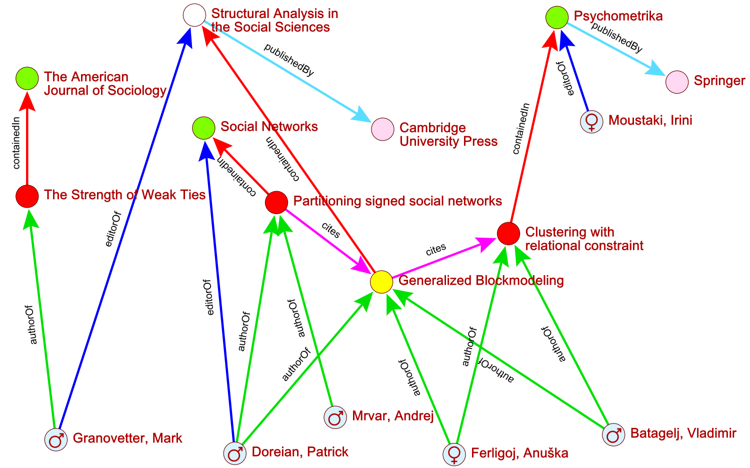
```
*vertices  16
1 "Batagelj, Vladimir"
2 "Doreian, Patrick"
3 "Ferligoj, Anuška"
4 "Granovetter, Mark"
5 "Moustaki, Irini"
6 "Mrvar, Andrej"
7 "Clustering with relational constraint"
8 "The Strength of Weak Ties"
9 "Partitioning signed social networks"
10 "Generalized Blockmodeling"
11 "Psychometrika"
12 "Social Networks"
13 "The American Journal of Sociology"
14 "Structural Analysis in the Social Sciences"
15 "Cambridge University Press"
16 "Springer"
*arcs :1 "authorOf"
*arcs :2 "cites"
*arcs :3 "containedIn"
*arcs :4 "editorOf"
*arcs :5 "publishedBy"
```

```
*arcs
1: 1 10 1 l "authorOf"
1: 2 10 1 l "authorOf"
1: 3 10 1 l "authorOf"
1: 1 7 1 l "authorOf"
1: 3 7 1 l "authorOf"
1: 4 8 1 l "authorOf"
4: 4 14 1 l "editorOf"
1: 2 9 1 l "authorOf"
1: 6 9 1 l "authorOf"
4: 5 11 1 l "editorOf"
4: 2 12 1 l "editorOf"
3: 10 14 1 l "containedIn"
3: 7 11 1 l "containedIn"
3: 8 13 1 l "containedIn"
3: 9 12 1 l "containedIn"
2: 9 10 1 l "cites"
2: 10 7 1 l "cites"
5: 14 15 1 l "publishedBy"
5: 11 16 1 l "publishedBy"
```

bib.net, bibMode.clu, bibSex.clu; bib.paj, bib.ini.

Reading Pajek files in R

Bibliographic network – picture / Pajek



Nets and NetsJSON

We were satisfied with the "traditional" network description until we became interested in networks with node/link properties that are not measured in standard scales (ratio, interval, ordinal, nominal), but have structured values (text, subset, interval, distribution, time series, temporal quantity, function, etc.). For describing temporal networks we initially, extending the Pajek format, defined and used the Ianus format [7]. We needed a format that could describe structured values. We could base our format on two options – XML and JSON. We chose JSON and in 2015 started developing the NetJSON format and the Nets Python package to handle networks with structured-valued properties [8–10].

. . . Nets and NetsJSON

On February 26, 2019, the format was renamed to NetsJSON because of the collision with http://netjson.org/rfc.html. NetsJSON has two versions: a *basic* and a *general* version. The current implementation of the Nets library supports only the basic version.

In addition to describing networks with structured values, NetsJSON is expected to offer the capabilities of (most) existing network description formats [11] (archiving, conversion) and provide input data for D3.js visualizations.

Informal description of the basic netsJSON format


```
{
"netsJSON": "basic",
"info": {
   "org":1, "nNodes":n, "nArcs":mA, "nEdges":mE,
   "simple":TF, "directed":TF, "multirel":TF, "mode":m,
   "network":fName, "title":title,
   "time": { "Tmin":tm, "Tmax":tM, "Tlabs": {labs} },
   "meta": [events], ...
   },
"nodes": [
   { "id":nodeId, "lab":label, "x":x, "y":y, ... },
   ***
   ]
"links": [
   { "type":arc/edge, "n1":nodeID1, "n2":nodeID2, "rel":r, ... }
   ***
   ]
}
```

where ... are user-defined properties and *** is a sequence of such elements.

Basic netsJSON format

An event description can contain fields:

```
{  "date": date,
   "title": short description,
   "author": name,
   "desc": long description,
   "url": URL,
   "cite": reference,
   "copy": copyright
}
```

It is intended to provide information about the "life" of the dataset – changes, releases, uses, publications, etc.

For describing temporal networks a node element and a link element have an additional required property `tq` – a temporal quantity.

For an example see GitHub/Bavla/Graph/JSON/violenceU.json describing Franzosi's violence network.

Basic netsJSON format

The general NetsJSON format is also expected to support the description of network collections.

In recent years we also analyzed bike systems (link weight is a daily number of trips distribution), bibliographies (yearly distributions of publications or citations), and multiway networks [12–14]. It turned out that it was necessary to add another main field, `data`, to the basic NetsJSON format, in which we provide additional data about the properties of values (translations of labels in selected languages, algebraic structure [15]).

Elements of a common network format

Our experience with network analysis to date is summarized in the following recommendations on the elements of a common format for describing networks.

For data integrity, it makes sense to combine data and metadata into a single file. To preserve the structure of data, it makes sense to base the format on JSON, which fits well with the data structures of modern programming languages.

We would also encourage the provision, as metadata, of information about the context of the network, additional knowledge about it, articles or notebooks on its analysis, comments of users, etc. Kaggle is a good example. An improved ICON repository or Networkrepository (we disagree with their "citation request") could be the way to go. Existing metadata standards should be taken into account (Dublin Core, FAIR, Schema). Data has a "life". When selecting data, its age is often important. Metadata should include at least the creation date and the last modification date.

. . . Elements of a common network format

The format must support all types of networks (simple, 2-mode, linked, multi-relational, multi-level, temporal). The network can contain both arcs and edges, as well as parallel links.

As mentioned earlier, using factorization produces a more concise description of the network. In cases where the node names are not too long and are readable, we sometimes want to avoid factorization. This can be achieved by using a switch that indicates whether factorization is used. We can also shorten the description length by introducing default values. If we also allow counting from 0, it makes sense to add information about the smallest index.

Long labels cause problems when printing/visualizing (parts of) networks and results. Therefore, it is useful to have abbreviated versions of labels available.

...Elements of a common network format

Most of the network datasets produced by network science have no node labels. Node labels are not needed if you study distributions, but they are very important in the interpretation of the obtained "important substructures". We would encourage providing node labels, or at least some typology info in the case of privacy issues.

Operations and transformations on networks: extensions (new yearly data), constructive description (, NetML [16]), intersection, union, ... product; derived networks .

We have not yet started working on a general format. It is supposed to enable descriptions of collections of networks. The question arises about the scope of validity of IDs - does the same ID in different networks represent the same or different units? This is important for operations such as union or intersection of networks. Which way to go - introducing contexts or using matchings?

…Elements of a common network format

Generalizations: multiway, hypernets.

Additional ideas may be found on the page "*A Python Graph API?*" [17].

Conclusions

The availability of the data used in the article enables the reproducibility and verifiability of the analyses performed. In addition, the obtained results can be verified or supplemented with other methods. When developing new methods, accessible and well-documented data are also very important - it is good to test a new method on several data sets and check whether it gives meaningful/expected results.

Network format

V. Batagelj et al.

Acknowledgments

The computational work reported in this paper was performed using programs R and Pajek [18]. The code and data are available at Github/Bavla [6].

This work is supported in part by the Slovenian Research Agency (research program P1-0294, research program CogniCom (0013103) at the University of Primorska, and research projects J5-2557, J1-2481, and J5-4596), and prepared within the framework of the COST action CA21163 (HiTEc).

References I

1. Hogan, A. *et al.* Knowledge graphs. *ACM Computing Surveys (Csur)* **54,** 1–37 (2021).
2. Ehrlinger, L. & Wöß, W. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* **48,** 2 (2016).
3. Neal, Z. P. *et al.* Recommendations for sharing network data and materials. *Network Science* **12,** 404–417 (2024).
4. Neal, Z. P. & et al. *Recommendations for sharing network data and materials – Endorsement page.* https://www.zacharyneal.com/datasharing. Accessed: 2025-03-18. 2024.
5. Gephi. *Supported Graph Formats.* https://gephi.org/users/supported-graph-formats/. Accessed: 2025-03-18. 2022.
6. GitHub/Bavla. *Bibliographic network example.* https://github.com/bavla/netsJSON/tree/master/example/bib. Accessed: 2025-03-18. 2016.

References II

7. Batagelj, V. & Praprotnik, S. An algebraic approach to temporal network analysis based on temporal quantities. *Social Network Analysis and Mining* **6**, 1–22 (2016).

8. Batagelj, V. *Network visualization based on JSON and D3.js.* https://github.com/bavla/netsJSON/blob/master/doc/netVis.pdf. Accessed: 2025-03-18. 2016.

9. GitHub/Bavla. *NetsJSON – a JSON format for network analysis.* https://github.com/bavla/netsJSON. Accessed: 2025-03-18. 2016.

10. GitHub/Bavla. *Nets – Python package for network analysis.* https://github.com/bavla/Nets. Accessed: 2025-03-18. 2016.

11. Bodlaj, J. & Cerinšek, M. in *Encyclopedia of Social Network Analysis and Mining* (eds Alhajj, R. & Rokne, J.) 1525–1540 (Springer New York, New York, NY, 2018). ISBN: 978-1-4939-7131-2. https://doi.org/10.1007/978-1-4939-7131-2_298.

12. Batagelj, V. & Ferligoj, A. *Symbolic network analysis of bike sharing data / Citi Bike.* https://github.com/bavla/Bikes/blob/master/bikes.pdf. Accessed: 2025-03-18. 2016.

References III

13. Batagelj, V. & Maltseva, D. Temporal bibliographic networks. *Journal of Informetrics* **14,** 101006 (2020).
14. Batagelj, V. Cores in multiway networks. *Social Network Analysis and Mining* **14,** 122 (2024).
15. Batagelj, V. *Semirings in network data analysis / an overview.* https://github.com/bavla/semirings/blob/master/docs/semirings.pdf. Accessed: 2025-03-18. 2021.
16. Batagelj, V. & Mrvar, A. *INSNA Sunbelt International Social Network Conference, London, July 6-10 1995. Slides.* https://github.com/bavla/netsJSON/blob/master/doc/snetml.pdf. Accessed: 2025-03-18. 1995.
17. The Python Wiki. *A Python Graph API?.* https://wiki.python.org/moin/PythonGraphApi. Accessed: 2025-03-18. 2011.
18. De Nooy, W., Mrvar, A. & Batagelj, V. *Exploratory social network analysis with Pajek.* (Cambridge university press, 2018).