

Drama Mining

Netzwerkanalyse II

Dr. Jochen Tiepmar

Institut für Informatik
Abteilung Computational Humanities
Abteilung Automatische Sprachverarbeitung
Universität Leipzig

27. Mai 2019

[Letzte Aktualisierung: 27/05/2019, 14:17]

Überblick

- 1 Graphentheorie
- 2 Visualisierung
- 3 Netzwerkanalyse
- 4 Netzwerkanwendungen
- 5 Vor- und Nachteile

- 1 Graphentheorie
- 2 Visualisierung
- 3 Netzwerkanalyse
- 4 Netzwerkanwendungen
- 5 Vor- und Nachteile

Zusammenfassung von letzter Woche

- Arten von Graphen
 - Gerichtet, Gewichtet
 - Teilgraphen, Komponenten
 - Property, Semantisch, Temporal
- Drama Mining → Konfigurationsmatrix → Graphanalyse

- 1 Graphentheorie
- 2 Visualisierung**
- 3 Netzwerkanalyse
- 4 Netzwerkanwendungen
- 5 Vor- und Nachteile

Zusammenfassung von letzter Woche

- Ästhetik von Graphen, Visualisierung vs. Interpretation
- Graphlayouts

1 Graphentheorie

2 Visualisierung

3 Netzwerkanalyse

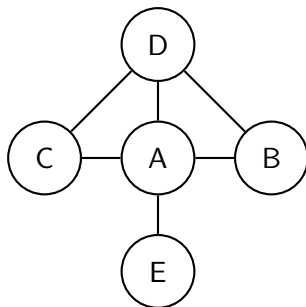
- 7 Punkte Zusammenfassung
- Prestige
- Zentralität
- Pagerank

4 Netzwerkanwendungen

5 Vor- und Nachteile

7 Punkte Zusammenfassung

- Größe: $|V|$ = Anzahl Knoten und $|E|$ = Anzahl Kanten
- Dichte = $\frac{|E|}{\text{Anzahl Knotenpaare}}$
- Anzahl Gruppen = Komponenten (Maximal verbundene Teilgraphen)
- Kompaktheit = Längster Kürzester Pfad, Diameter = Durchschnittliche kürzeste Pfadlänge
- Transitivität = $\frac{\text{Anzahl_Dreiecke}}{\text{Anzahl_verbundener_Tripel}}$



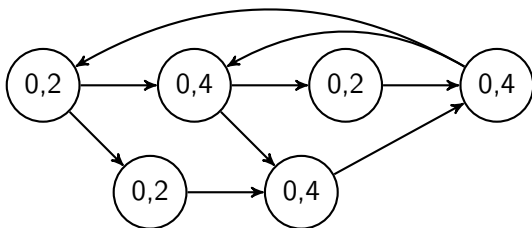
- Größe: $|V| = 5$ und $|E| = 6$
- Dichte = $\frac{|E|}{\text{Anzahl Knotenpaare}} = \frac{6}{20} = 0,3$
- Anzahl Gruppen = Komponenten = 1
- Kompaktheit = 2, Diameter = 1,45
- Transitivität = $\frac{\text{Anzahl_Dreiecke}}{\text{Anzahl_verbundener_Tripel}} = \frac{2}{4} = 0,5$

Prestige

Es sei $G = (V, E)$ ein gerichteter Graph mit

- $|V|$ = Anzahl der Knoten
- $EG(v)$ = Eingangsgrad von v

Degree Prestige / Relativer Innengrad $DP(v) = \frac{EG(v)}{|V|-1}$



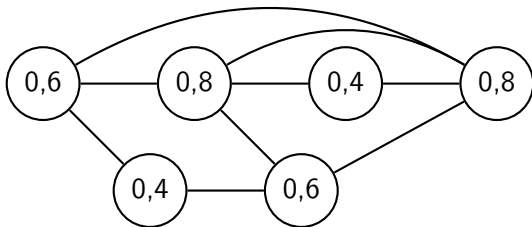
- Simpel zu berechnen
- Mit Temporalen Graphen Absteiger und Aufsteiger berechenbar
- Rang Prestige komplexer aber nur eingeschränkt anwendbar

Zentralität

Es sei $G = (V, E)$ ein ungerichteter Graph mit

- $|V|$ = Anzahl der Knoten
- $G(v)$ = Grad von v

Degree Centrality / Gradzentralität $DC(v) = \frac{G(v)}{|V|-1}$

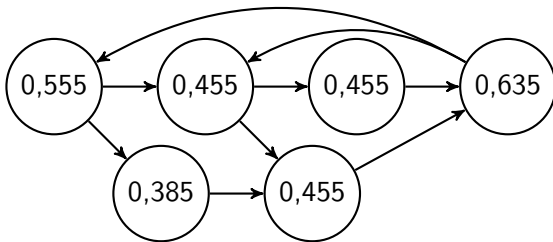


- Im Grunde Degree Prestige auf ungerichteten Graphen

Es sei $G = (V, E)$ ein (un)gerichteter Graph mit

- $P(v, u)$ = Anzahl der Kanten der kürzesten duplikatfreien Verbindung von Knoten v zu Knoten u (Pfadlänge zwischen v zu u)
 - $P(v, u)$ = unendlich wenn kein Pfad existiert (i)
 - Berechnung $\rightarrow$ Dijkstra, Bellman-Ford,...
- $P_{avg}(v)$ = Summe aller $P(v, u)$ für alle $u \in V$
- $|V|$ = Anzahl der Knoten

Closeness Centrality $CC(v) = \frac{1}{\frac{P_{avg}(v)}{|V|-1}} = \frac{|V|-1}{P_{avg}(v)}$

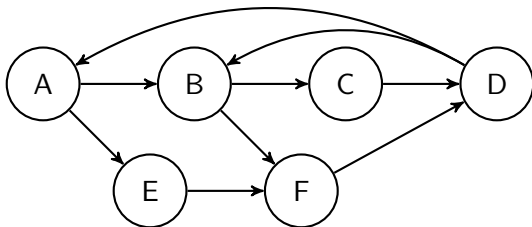


- Auf unverbundenen Graphen nicht berechenbar $\rightarrow$ (i)

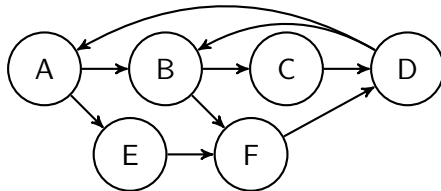
Es sei $G = (V, E)$ ein (un)gerichteter Graph mit

- P = Menge aller kürzesten Verbindungen zwischen allen Knotenpaaren aus V
 - Achtung: $(x, y) \neq (y, x)$
- P_v = Menge aller $p \in P$, die v enthalten, aber nicht mit v beginnen oder enden
- Normalisierung: Division durch $(|V| - 1) * (|V| - 2)$ (gerichtet) oder $\frac{(|V|-1)*(|V|-2)}{2}$ (ungerichtet)

Betweenness Centrality $BC(v) = \frac{P_v}{P}$

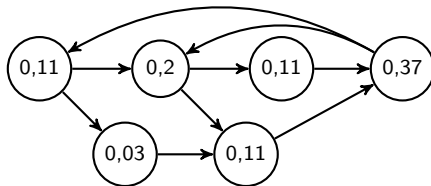


- Auf unverbundenen Graphen nicht berechenbar $\rightarrow$ (i)



$$NR = \begin{pmatrix} & A & B & C & D & E & F \\ A & - & BCDA/BFDA & CDA & DA & EFDA & FDA \\ B & AB & - & CDB & DB & EFDB & FDB \\ C & ABC & BC & - & DBC & EFDBC & FD BC \\ D & ABCD/ABFD & BCD/BFD & CD & - & EFD & FD \\ E & AE & BCDAE/BFDAE & CDAE & DAE & - & FDAE \\ F & ABF/AEF & BF & CDBF & DBF & EF & - \end{pmatrix}$$

$$BC(v) = \frac{P_v}{35}$$

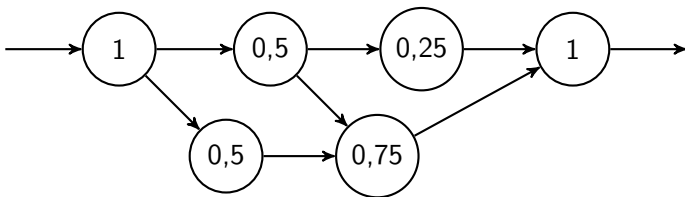


Pagerank

Es sei $G = (V, E)$ ein gerichteter Graph mit

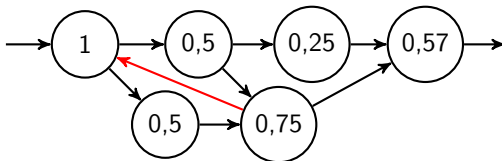
- $(a, b) = \text{Kante aus } E$
- $AG(a) = \text{Ausgangsgrad von } a$
- $V_b = \text{Vorgängermenge von } b$

Pagerank $R(b) = \sum_{a \in V_b} \frac{R(a)}{AG(a)}$

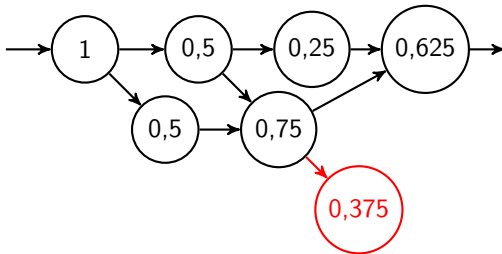


Pagerank

- Tendenziell für chaotische querverlinkte Graphen
- Angewendet auf Internet (Grundstein für Google)
- Iterativ berechnet bis Ergebnis konvergiert
- Probleme
 - Senken



- Sackgassen



Es sei $G = (V, E)$ ein gerichteter Graph mit

- $(u, v) = \text{Kante aus } E$
- $F_v = \text{Ausgangsgrad von } v$
- $B_u = \text{Vorgänger von } u$
- $c = \text{Normalisierungsfaktor (empirischer Richtwert 0.95)}$

Pagerank
$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

Anwendung des Random-Surfer-Modell

- $d = \text{Dämpfungsfaktor, ähnlich } c$
- $(1 - d) = \text{Wahrscheinlichkeit, dass jemand zufällig/ohne Link auf einer Seite landet}$

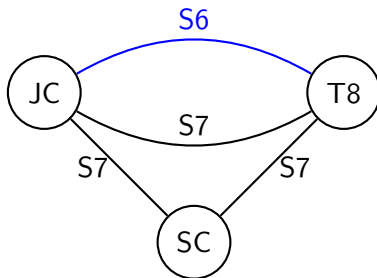
Pagerank
$$R(u) = (1 - d) + d \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- 1 Graphentheorie
- 2 Visualisierung
- 3 Netzwerkanalyse
- 4 **Netzwerkanwendungen**
 - Social Network Analysis
 - Zitatanalyse
 - Modellierung von Argumentation
 - Semantic Web / Linked Open Data
- 5 Vor- und Nachteile

Social Network Analysis

Soziales Netzwerk / Figurennetzwerk als Graph

- Knoten = Figuren / Akteure
- Kanten = Gerichtete (!) Beziehung
- Inverse Relationen (Sohn_von, Mutter_von) oder bidirektional (Freund_von)
- Ungerichtete Graphen eher unpassend (reduzieren auf bidirektionale Relation)
- Sonst Graphen beliebig nutzbar (Gewicht, Temporal, Property, Semantisch,...)
- Graphanalysen direkt auf Soziale Netzwerke anwendbar



Zitatanalyse

:1
:1.1
:1.1.1 O Tannenbaum, O Tannenbaum,
:1.1.2 Wie treu sind deine Blätter.
:1.1.3 Du grünst nicht nur zur Sommerzeit,
:1.1.4 Nein auch im Winter wenn es schneit.
:1.1.5 O Tannenbaum, O Tannenbaum,
:1.1.6 Wie grün sind deine Blätter!
:1.2
:1.2.1 O Tannenbaum, O Tannenbaum,
:1.2.2 Du kannst mir sehr gefallen!
:1.2.3 Wie oft hat schon zur Winterszeit
:1.2.4 Ein Baum von dir mich hoch erfreut!
:1.2.5 O Tannenbaum, O Tannenbaum,
:1.2.6 Du kannst mir sehr gefallen!
:1.3
:1.3.1 O Tannenbaum, O Tannenbaum,
:1.3.2 Dein Kleid will mich was lehren:
:1.3.3 Die Hoffnung und Beständigkeit
:1.3.4 Gibt Mut und Kraft zu jeder Zeit!
:1.3.5 O Tannenbaum, O Tannenbaum,
:1.3.6 Dein Kleid will mich was lehren.

:1.1.1 → 1.1.5, 1.2.1, 1.2.5, 1.3.1, 1.3.5
:1.2.2 → 1.2.6
:1.3.2 → 1.3.6
:1.1.2 → 1.1.6(?)

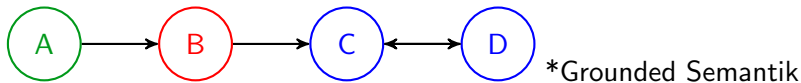
- Wer zitiert wen?
- AKA Textreue
- Berechnung eines (gerichteten) Zitatgraphen
 - Knoten = Referenzen für Textstellen (bspw. CTS URNs)
 - Kanten = Ähnlichkeit $>$ Threshold $\times$
 - (Richtung anhand des Publikationsdatums)
- Alternativ Nutzung von Quellenverweisen

- Pagerank als Zitatanalyse
- Serendipitätsprinzip
- Finden von disziplinaren Netzwerken
- Identifizieren von Forschungscommunities
- Finden von gehäuften Selbstreferenzen

Modellierung von Argumentation

Dung's Argumentation Framework

- Knoten = Argumente / widersprüchlichen Fakten
- gerichtete Kanten = *Attack*-Relation
- 3 Knotenlabel:
 - **gerechtfertigt (IN)**: kein attackierender Knoten ist IN
 - **ungerechtfertigt (OUT)**: mindestens 1 attackierender Knoten ist IN
 - **unentscheidbar (UNDEC)**: kein attackierender Knoten ist IN, aber mindestens 1 nicht OUT



- 1 Konfiguration = 1 Labeling (zulässig & vollständig)
- SNA: Modellierung von Feindschaftsrelationen

Dung's Argumentation Framework

- 3 Semantiken

- Grounded

DO

Alle nicht oder von **OUT** attackierten Knoten $\rightarrow$ **IN**

Alle von **IN** attackierten Knoten $\rightarrow$ **OUT**

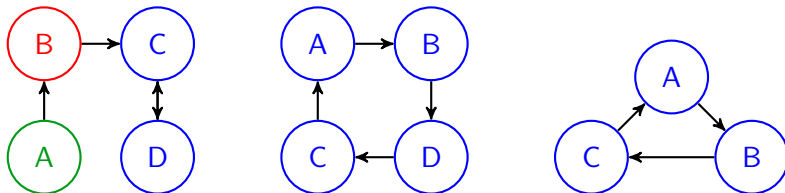
WHILE (es gibt unattackierte Knoten)

Jeder Knoten ohne Label $\rightarrow$ **UNDEC**

- Preferred

- Stable

- Grounded **IN** und **OUT** werden in jeder Semantik analog gewertet



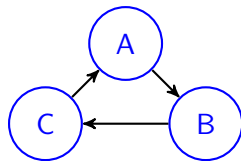
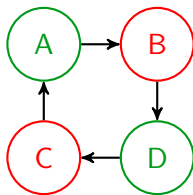
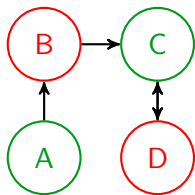
Dung's Argumentation Framework

- 3 Semantiken

- Grounded
- Preferred

- Labeling L_1 , sodass kein zulässiges Labeling L_2 existiert mit $L_1 \in L_2$
- Mehrdeutig
- Im Grunde What-If Szenario
- Rekursive Berechnung aufwendig (Siehe Literatur) aber grob:
Erstelle All-IN Labeling
Repariere bis zulässig

- Stable



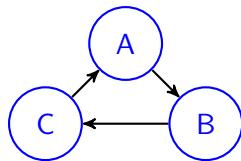
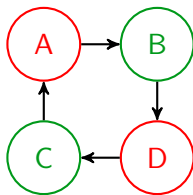
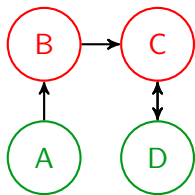
Dung's Argumentation Framework

- 3 Semantiken

- Grounded
- Preferred

- Labeling L_1 , sodass kein zulässiges Labeling L_2 existiert mit $L_1 \in L_2$
- Mehrdeutig
- Im Grunde What-If Szenario
- Rekursive Berechnung aufwendig (Siehe Literatur) aber grob:
Erstelle All-IN Labeling
Repariere bis zulässig

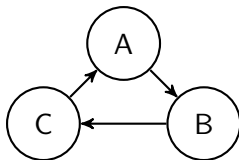
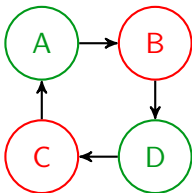
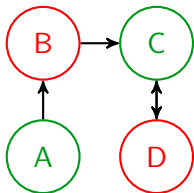
- Stable



Dung's Argumentation Framework

- 3 Semantiken

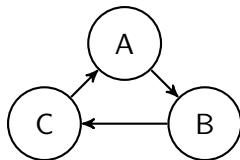
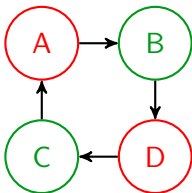
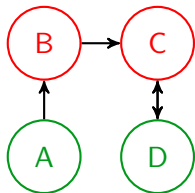
- Grounded
- Preferred
- Stable
 - Jedes Preferred Labeling ohne UNDEC
 - Mehrdeutig
 - Nicht immer möglich



Dung's Argumentation Framework

- 3 Semantiken

- Grounded
- Preferred
- Stable
 - Jedes Preferred Labeling ohne UNDEC
 - Mehrdeutig
 - Nicht immer möglich



Dung's Argumentation Framework

- Lehre

- Prof. Dr. Gerhard Brewka, Dr. Ringo Baumann

- Literatur

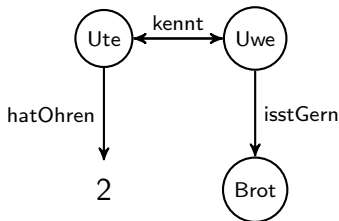
- Baroni, P., and Giacomin, M.(2009), Semantics of Abstract Argument Systems in Argumentation in Artificial Intelligence eds. I. Rahwan and G. Sinari, Berlin:Springer, pp.25-44
 - Modgil, S., and Caminada, M. (2009), Proof Theories and Algorithms for Abstract Argumentation Frameworks in Argumentation in Artificial Intelligence eds. I.Rahwan and G. Sinari, Berlin:Springer, pp. 105-129
 - Dung, P. (1995), On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence, 77(2) 321-357.
 - Oder als deutsche (aber Bachelor-) Arbeit:
<http://nbn-resolving.de/urn:nbn:de:bsz:15-qucosa2-172000>

Semantic Web / Linked Open Data

Ein gerichteter Graph ist ein semantischer Graph wenn

- S = eine endliche Menge von Entitäten (Subjekten)
- O = eine endliche Menge von Entitäten und Literalen (Objekten)
- R = eine Menge von Relationen
- E = eine Menge von Tripeln aus je einem Element aus S , O und R

Beispiel $S = \{Uwe, Ute\}$, $O = \{Eis, Brot\}$, $R = \{isstGern, hatOhren, kennt\}$,
 $E = \{(Ute, hatOhren, "2")\}, (Uwe, isstGern, Brot), (Ute, kennt, Uwe), (Uwe, kennt, Ute)\}$



Ein gerichteter Graph ist ein semantischer Graph wenn

- S = eine endliche Menge von Entitäten (Subjekten)
- O = eine endliche Menge von Entitäten und Literalen (Objekten)
- R = eine Menge von Relationen
- E = eine Menge von Tripeln aus je einem Element aus S , O und R

Beispiel $S = \{Uwe, Ute\}$, $O = \{Eis, Brot\}$, $R = \{isstGern, hatOhren, kennt\}$,
 $E = \{(Ute, hatOhren, "2"), (Uwe, isstGern, Brot), (Ute, kennt, Uwe), (Uwe, kennt, Ute)\}$

Subjekt	Prädikat	Objekt
< Uwe >	< hatOhren >	2
< Uwe >	< kennt >	< Ute >
< Ute >	< kennt >	< Uwe >
< Ute >	< isstGern >	< Brot >

Ontologie

- "Wissenssammlung"
- 2 Bestandteile
 - Schema / Regelwerk für Relationen / Taxonomien / Konzeptwissen
 - isstGern, hatOhren sind Relationen
 - hatOhren erwartet Ganzzahl als Objekt
 - RDF und RDFS in der Regel obligatorisch
 - Instanzdaten / Fakten
 - Beschrieben bspw. als RDF/XML oder Turtle

Beispiel $S = \{Uwe, Ute\}$, $O = \{Eis, Brot\}$, $R = \{isstGern, hatOhren, kennt\}$,
 $E = \{(Ute, hatOhren, "2"), (Uwe, isstGern, Brot), (Ute, kennt, Uwe), (Uwe, kennt, Ute)\}$

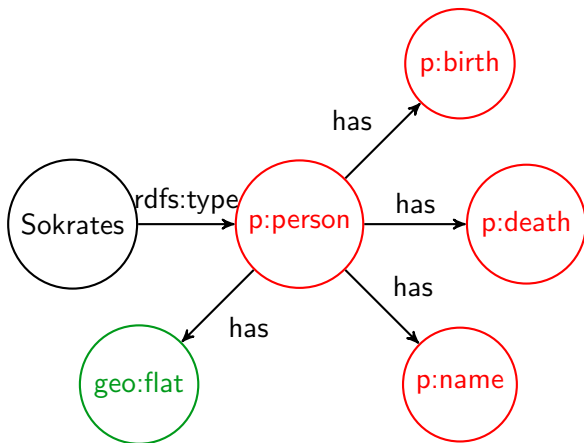
Subjekt	Prädikat	Objekt
< Uwe >	< hatOhren >	2
< Uwe >	< kennt >	< Ute >
< Ute >	< kennt >	< Uwe >
< Ute >	< isstGern >	< Brot >

Triplifizierung mit Turtle

Datenabfrage mit SPARQL

Transitives Schließen

- Größte Stärke von Semantik Web und Grundlage von Linked Open Data
- Transitiver Schluß auf **externes Schema person**:
 - Sokrates ist vom Typ p:person
 - p:person hatRelation death
 - → Sokrates hatRelation death



- 1 Graphentheorie
- 2 Visualisierung
- 3 Netzwerkanalyse
- 4 Netzwerkanwendungen
- 5 Vor- und Nachteile

Vor- und Nachteile Graphbasierter Datenmodellierung

- Vorteile

- Intuitive aber mathematisch fundierte Datenmodellierung
- Transitives Schließen
- Visualisierung als Analyse aussagekräftig
- Vielseitig konfigurierbar (temporal, gewichtet, ...)
- Vielseitig anwendbar

- Nachteile

- Overuse, insb. bezogen auf Semantic Web
- Die Intuitivität führt oft dazu, dass Probleme in Graphen gedrückt werden, obwohl andere Datenstrukturen besser wären
- Visualisierung anfällig für Fehlinterpretation
- Algorithmische Schwierigkeiten
 - Kreise schwer identifizierbar und generell problematisch
 - Keine inhärent geordnete Elementmenge,...
 - ...deshalb nur sequentielle Suche möglich

