Network Analysis of Tweets

Building and graphing networks of users and tweets

Outline

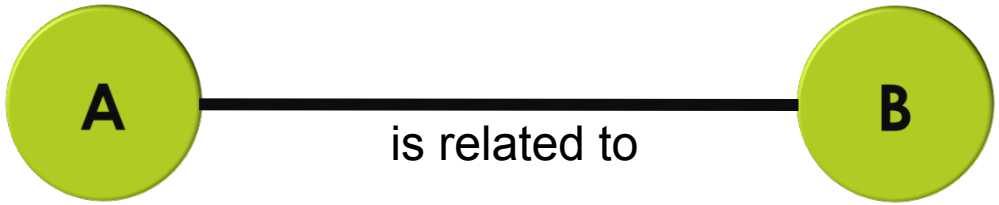
- Introduction to Graphs
  - Not "charting"
- Constructing Network Structure
  - build_graph.py
    - Parsing tweets
    - Using networkx
- Presentation of Graph data
  - Gephi

Basic Idea of a Graph

- "A" is related to "B"
  - Three things we want to represent
  - Item, person, concept, thing: "A"
  - Item, person, concept, thing: "B"
  - The "is related to" relation

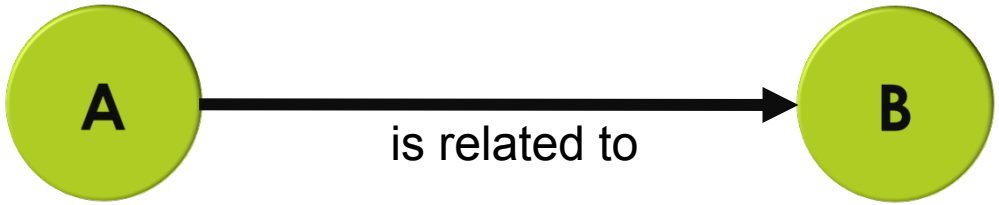
Basic Idea of a Graph

- "A" is related to "B"
  - Three things we want to represent
  - Item, person, concept, thing: "A"
  - Item, person, concept, thing: "B"
  - The "is related to" relation



Basic Idea of a Graph

- "A" is related to "B"
  - Three things we want to represent
  - Item, person, concept, thing: "A"
  - Item, person, concept, thing: "B"
  - The "is related to" relation
    - Relation can be directed, directional



Who Retweets Whom?

- AidanKellyUSA: RT @BreeSchaaf: Quick turn on @NBCSports Men's Singles Luge final! @mazdzer @AidanKellyUSA @TuckerWest1 are laying it all on the line tonig…
- Setrice93: RT @NBCOlympics: #Gold for @sagekotsenburg! First gold at #Sochi2014 and first-ever Olympic gold in snowboard slopestyle! http://t.co/0F8ys…
- adore_knob: RT @drdloveswater: I have waited 4 years to do this. Thank you @NBCOlympics & all your interns for such awesome coverage. #Sochi2014 http:/…
- MattJanik: RT @NBCOlympics Yeah, it's not good for your health.
- LisaKSimone: RT @robringham: Tired of @nbc / @NBCOlympics holding the Olympics hostage. Time for them to lose exclusivity. #NBCFail
- TS_Krupa: RT @NBCOlympics: RT @RedSox: Go Team USA!! @USOlympic #Sochi2014 http://t.co/anvneh5Mmy

Who Retweets Whom?

- AidanKellyUSA: RT @BreeSchaaf: Quick turn on @NBCSports Men's Singles Luge final! @mazdzer @AidanKellyUSA @TuckerWest1 are laying it all on the line tonig…

- Setrice93: RT @NBCOlympics: #Gold for @sagekotsenburg! First gold at #Sochi2014 and first-ever Olympic gold in snowboard slopestyle! http://t.co/0F8ys…

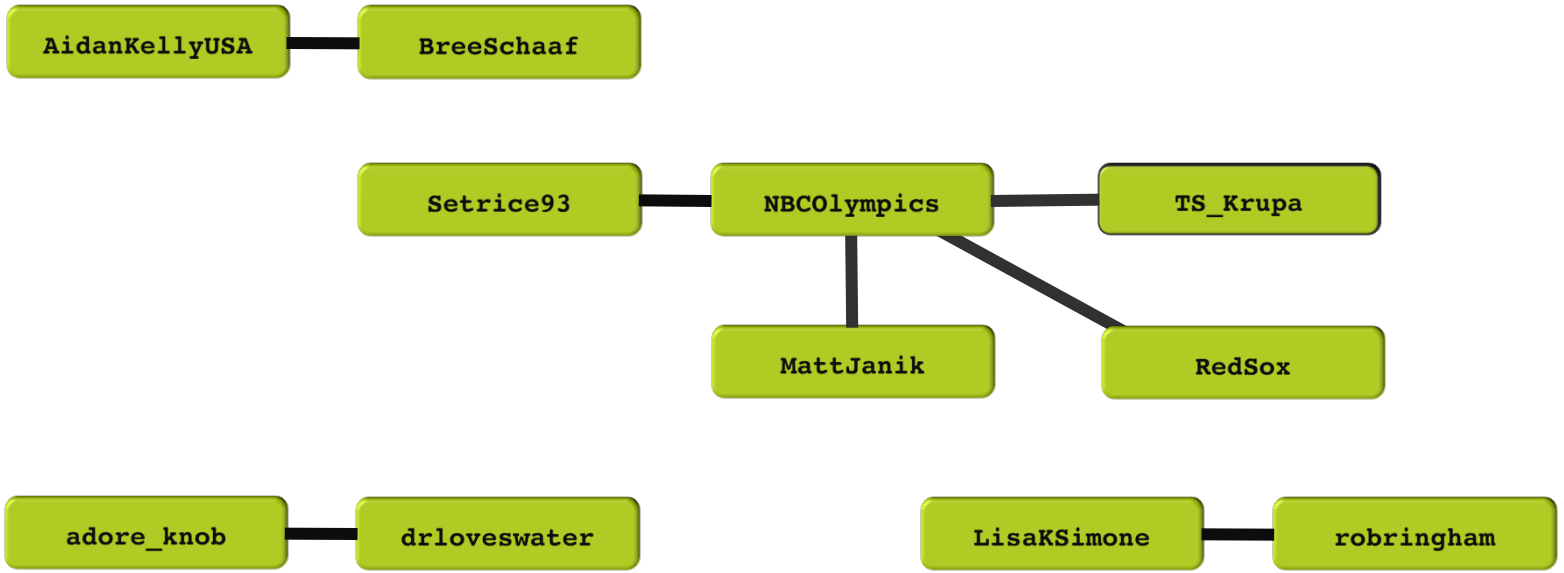
- adore_knob: RT @drdloveswater: I have waited 4 years to do this. Thank you @NBCOlympics & all your interns for such awesome coverage. #Sochi2014 http:/…

- MattJanik: RT @NBCOlympics Yeah, it's not good for your health.

- LisaKSimone: RT @robringham: Tired of @nbc / @NBCOlympics holding the Olympics hostage. Time for them to lose exclusivity. #NBCFail

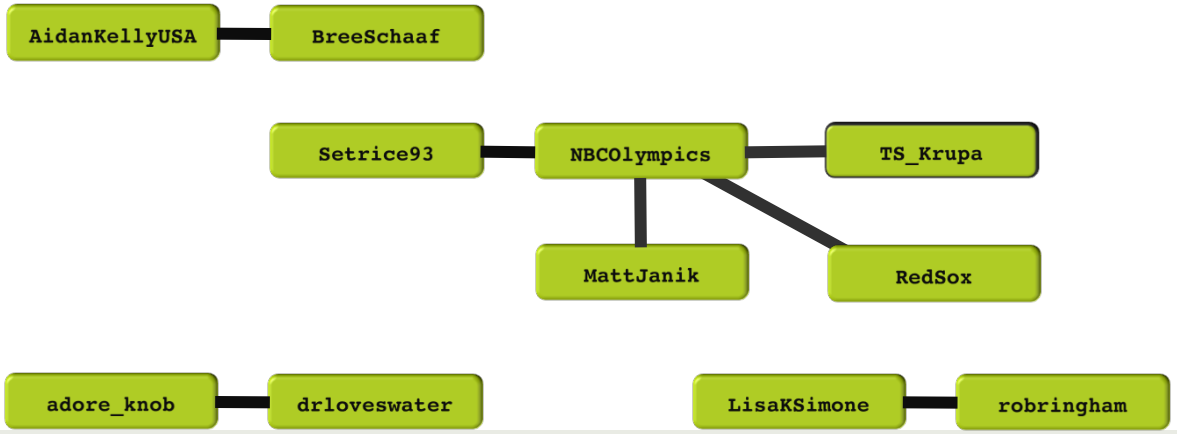
- TS_Krupa: RT @NBCOlympics: RT @RedSox: Go Team USA!! @USOlympic #Sochi2014 http://t.co/anvneh5Mmy

Who Retweets Whom?



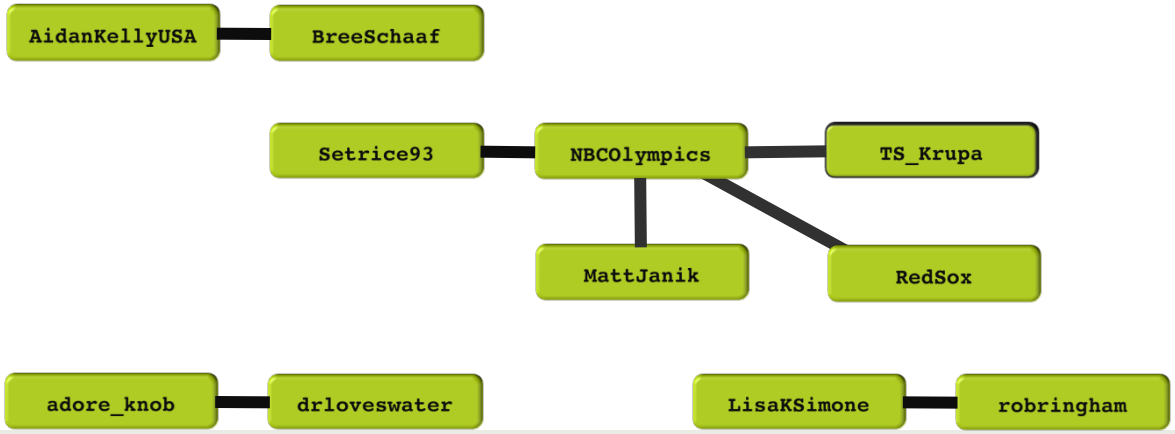
Some graph concepts

- Nodes – related items
  - Number
- Edges – the relations
  - Number



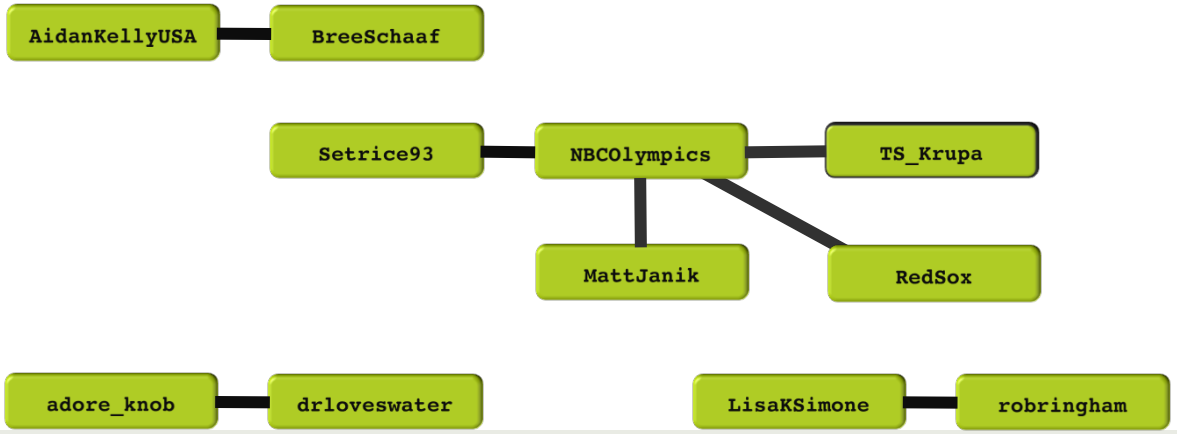
Some graph concepts

- Component (connected component)
  - A connected "chunk" of the whole thing
    - Example is one graph with four connected components
- Subgraph
  - Graph that can be found within another graph



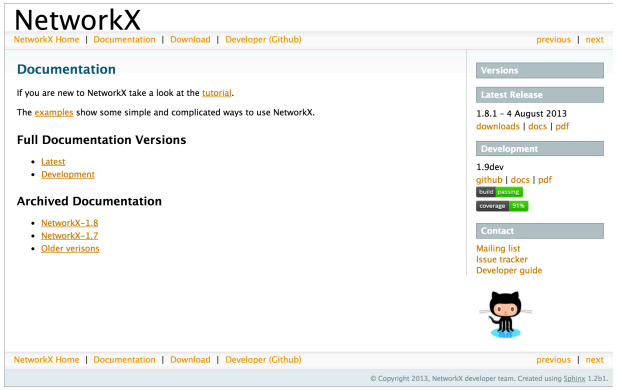
Some graph concepts

- Complete Graph
  - A graph where every node is connected to every other node
- Clique
  - A complete subgraph, a complete graph found within a graph

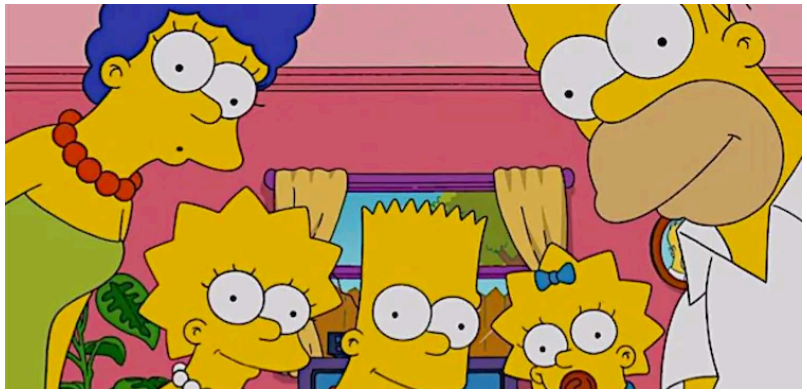


Networkx

- Networkx, a python graph creation and analysis tool
  - http://networkx.github.io/
- Good Documentation
  - https://networkx.github.io/documentation/stable/



How are we related?



Simpson's Social Network

```
import networkx as nx
g = nx.Graph() ## create a new undirected graph
# Add nodes and edges
g.add_edge("bart","marge")
g.add_edge("homer","marge")
g.add_edge("lisa","marge")
g.add_edge("maggie","marge")
g.add_edge("patty","marge")
g.add_edge("selma","marge")
g.add_edge("homer","lisa")
g.add_edge("homer","maggie")
g.add_edge("homer","bart")
g.add_edge("ned","todd")
g.add_edge("ned","rod")
g.add_edge("ned","maude")
g.add_edge("todd","maude")
g.add_edge("rod","maude")
```

Networks demo (nodes, edges)

```
# Print the number of nodes in the graph
print len(g.nodes())
# Print nodes - Just a list of the node names
print g.nodes()
# Print edges - A list of *node pairs*
print g.edges()
# Find all edges incident on one node - node pairs
print g.edges("marge")
# get the subgraph of all nodes around marge
nl = [ n[1] for n in g.edges("marge") ]
nl.append("marge")
sg = nx.Graph(g.subgraph(nl))
print sg.nodes()
print sg.edges()
```

Networks demo (calculations)

```
# some basic graph info
print nx.info(g)
# edge calculations
print nx.degree(g,"marge")
print nx.density(g)
# some centrality measures
print nx.degree_centrality(g)
print nx.betweenness_centrality(g)
print nx.eigenvector_centrality(g)
# find cliques
gclique = list(nx.find_cliques(g))
print gclique
# find connected components
comps = nx.connected_components(g)
print len(comps)
print comps[0]
print comps[1]
```

Tweet Networks

- Want code to build a retweet network
  - Collect tweets
  - For each tweet find if it's a retweet
  - Link the retweeting user to the retweeted user

  - Based on the text retweet convention
    - RT @dwmcphd <some tweet text>
- Command line usage
- Code walk through

Command Line

- Samples are in the directories
  - hcde/data/election_2012
  - hcde/data/election_2016
  - hcde/data/oscar_2016

```
python build_graph.py
USAGE: build_graph.py -date <date> -save <filename> [-dur <days>] [-digraph] [-
weighted] [-edge_cut] [-comp] [-dot | -graphml] [-report | -no_report]
```

Create a Graph (election 2012)

- `python build_graph.py -date 20121015 -save class_sample -edge_cut -graphml –report`
  - October 15, 2012 (election data)
  - Saving filename as 'class_sample'
  - Perform a single edge cut, remove singleton edges, nodes
  - Write the file in GraphML
  - Report activity to the screen (who is retweeting who)

```
< ... lots of text scrolls by ... >

Graph has 62751 nodes and 92841 edges.
Performing recursive single edge cut.
Made 6 passes through the graph, cut 45223 edges and 50905 nodes.
Graph has 11846 nodes and 47618 edges.
Writing GraphML file: class_sample-20121015-dur01-edge_cut.graphml
```

build_graph.py

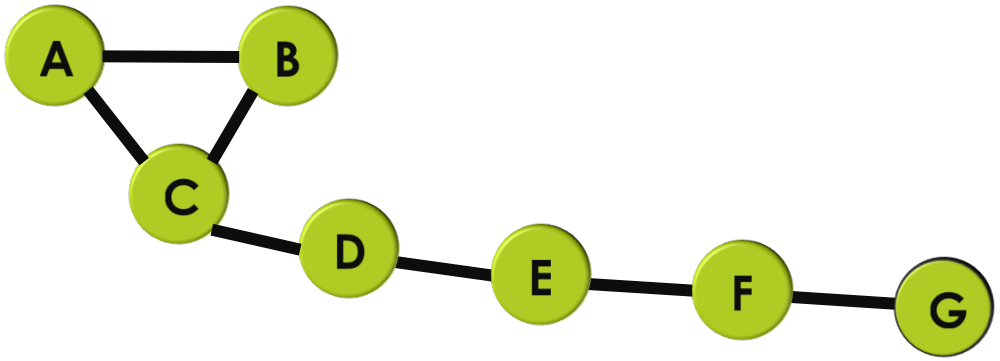
- Looking through the code

Caveats

- Can build directed or undirected graph
- Any/all retweets create a connection between users
- Single edge cut (recursive)

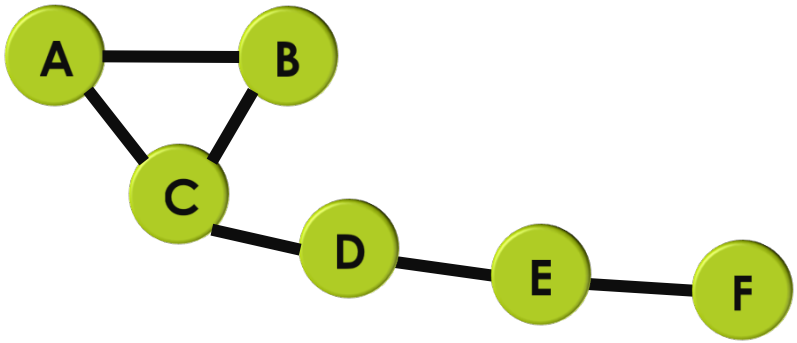
Single Edge Cut

- Do we want long chains of retweets, or main clump of people?



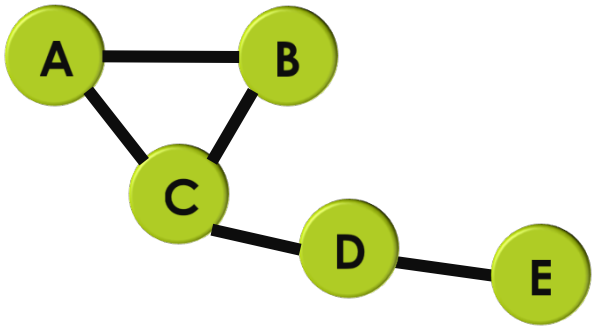
Single Edge Cut

- Do we want long chains of retweets, or main clump of people?



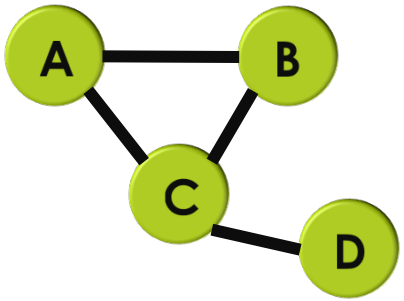
Single Edge Cut

- Do we want long chains of retweets, or main clump of people?



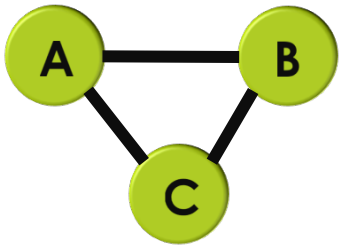
Single Edge Cut

- Do we want long chains of retweets, or main clump of people?



Single Edge Cut

- Do we want long chains of retweets, or main clump of people?

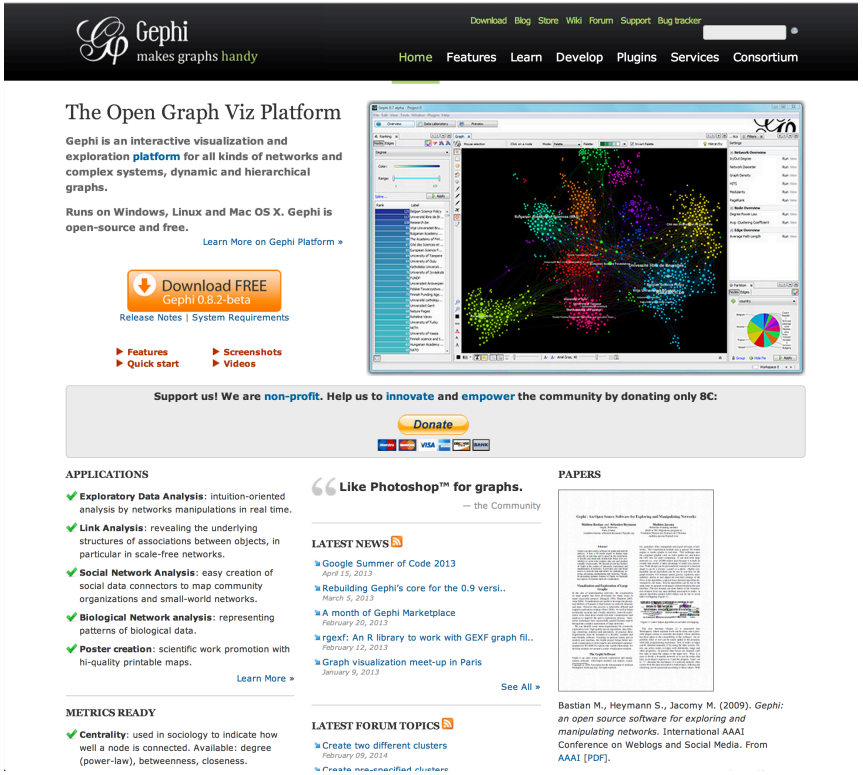


Visualizing graph data

- build_graph.py
  - Can dump GraphML (Graph Markup Language)
    - Good for Gephi (static picture, desktop app)
  - Can dump a "dot" file
    - Good for GraphVis (old, crufty, command line tool)
- Possible modifications to build_graph.py
  - Could be modified to use JSON output in Networkx
    - Maybe useful for Plotly or D3

Gephi

- https://gephi.org/
- Great tool – useful
- Possible to use Plotly
  - https://plot.ly/python/network-graphs/



Gephi Tutorial

- https://gephi.org/users/
- Tutorial is very good
  - You should do it

Visualization

Possible Modifications

- build_graph.py
  - Network of @mentions (who mentions who)
    - Possibly directed graph
  - Network of #hashtag use (who uses which hashtags)
    - Is what we call a 2 mode network
  - Extract, save Component
  - Extract, save Clique