

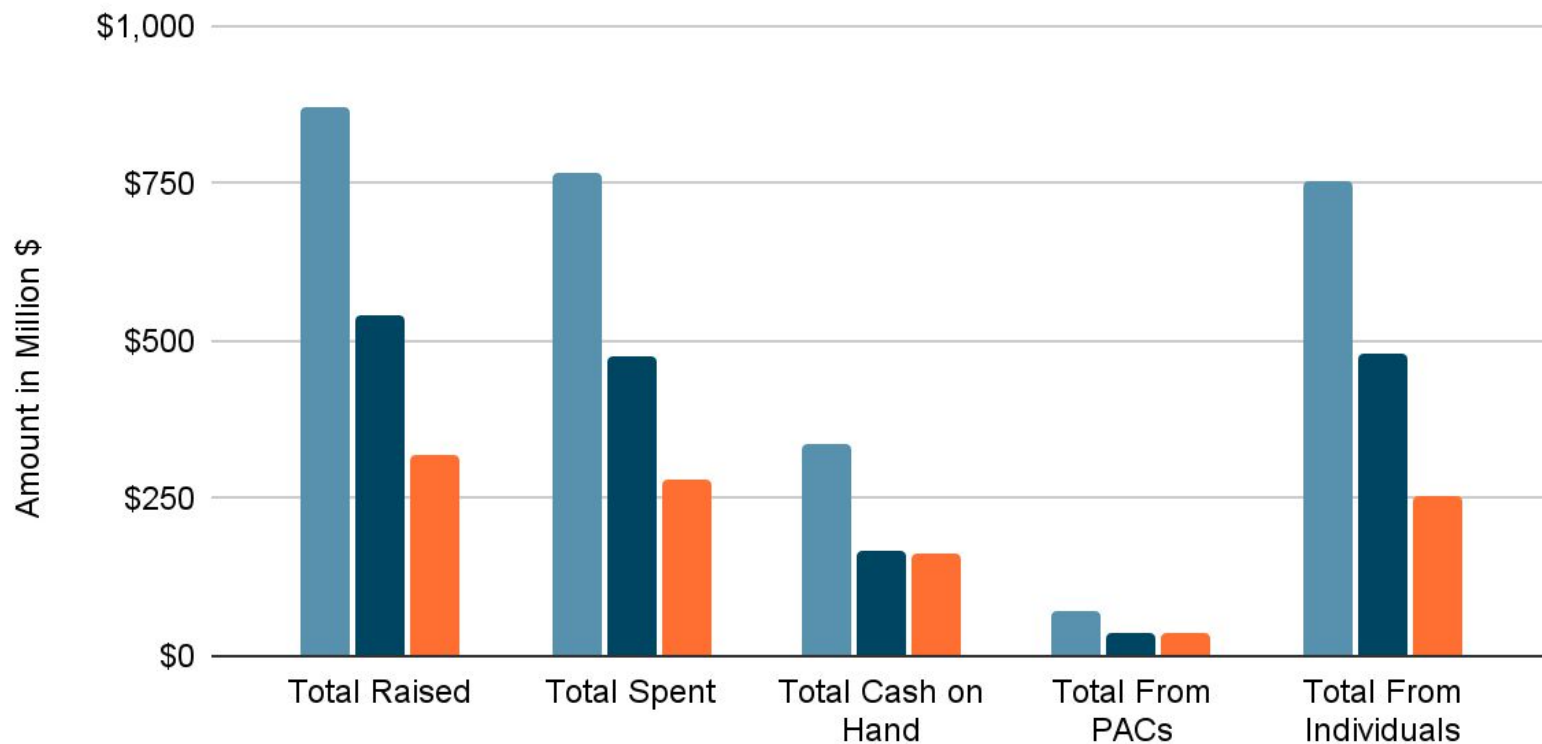
# S117 US Senators Knowledge Graph

Aidan Keogh, Willy Markuske, and Zach  
Wallace



# Senator Fundraising/Spending 2022

All Dems Repubs





# Introduction

The purpose of our knowledge graph is to allow users to map financial contributors and vote history of senators in the 117th Senate with their publicly available statements via Twitter.

The graph uses publically available vote data, financial contribution data, and senator statements on twitter to allow a user to investigate how senators vote compared to public statements and their financial contributors. By using this graph it is possible to determine what entities a senator tweets about and how that corresponds to their voting record. Or it can be seen how much money individuals donated to campaigns for senators that voted specific ways on votes that mention topics.



# Data Sources

Voteview.com



- csv files with Senate session 117 information
  - Senator biographical data - ICPSR ID, State, Name
  - Senator party data - Name, Party
  - Senate rollcall votes - Roll Number, Date, Yea Count, Nay Count, Description
  - Senator vote record - ICPSR, Roll Number, Cast
- Provides unique Senator ICPSR ID to identify unique individuals

OpenSecrets.org

- csv files with contribution amounts for the latest election that a senator was elected in by states
  - Name, Contributor, Individual Contributions, PAC Contributions, Total Contributions
- Contributors listed are the sum total of contributions from individuals that are associated with an organization
- Provides both corporate and political action committee contributions



# Data Sources



## Twitter

- csv listing from Kaggle.com of all current senator twitter handles to link individuals with tweets
  - Name, Handle
- json file of tweets from the list of twitter handles, about 10K tweets total
  - handle:
    - created\_at:
    - text:
    - public\_metrics:
      - retweet\_count:
      - reply\_count:
      - like\_count
      - Quote\_count
- Twitter API tutorial found at <https://github.com/twitterdev/getting-started-with-the-twitter-api-v2-for-academic-research>
- Limited in the tweet timeframe for free API, up to 3200 most recent tweets from user's timeline



# Preparing Senators & Contributors

- 1) Extract a listing of all senators with a unique ICPSR ID from voteview biographical data
- 2) Generate a distinct set of nodes for senators with their ID, name, and party affiliation
- 3) Entity match firstname and lastname values from the master senators list with contribution csv for each senate race and confirm against their state
- 4) Find all unique contributors using names across all senate race files using entity matching
- 5) Create an edge linking each unique contributor to each unique senator with the edge titled DONATED\_TO
  - a) Store the donation amount, number of individual from organization who donated, number of pacs from the donor, and race that was donated to as edge attributes



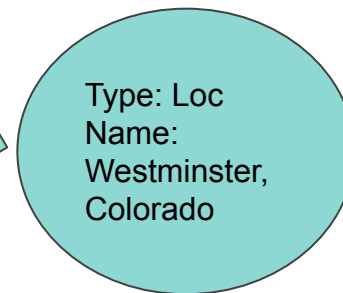
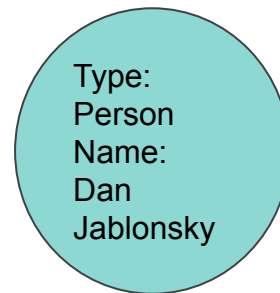
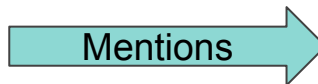
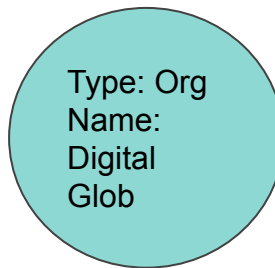
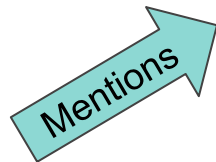
# Preparing the Tweets with Entities

- 1) Match Senators to Twitter Accounts
  - a) Match on first name and last name, manually curate the results
- 2) Pull Tweets:
  - a) Use the twitter API to pull recent tweets on the senator's user timeline, max 100 tweets per senator twitterID, store as a JSON file
  - b) From JSON file, save the tweet itself and public metrics (likes, replies, retweets, quotes)
- 3) Tweet NLP:
  - a) Conduct a NER on each tweet using Spacy, mapping each tweet to an entity type and entity name
- 4) Load the tweets in to Neo4j as individual nodes, with tweet text and public metrics as node attributes
- 5) Load the name entities as individual nodes, with the entity type and the entity name as node attributes
- 6) MATCH the tweet nodes to their corresponding entity node if the tweet mentions that named entity creating the edge MENTIONS



Tweet Node

## Entity Nodes







# Preparing Roll Calls with Entities

- 1) Process the S117 senate roll call information from [voteview.com](http://voteview.com) (S117\_rollcalls.csv)
  - a) Save the bill number, date of bill, yea count, nay count, margin (yea - nay), result, and bill description
- 2) Load in as individual node type to Neo4j with above attributes
- 3) Roll call NLP:
  - a) For the the bill descriptions, conduct a NER analysis to map descriptions to named entities
- 4) Load in the named entities from the roll call NER results as additional named entity nodes (with entity type and name as node attributes), merging with the ones from the twitter NER results
- 5) MATCH the roll call nodes to the corresponding named entity for which the bill description mentions, linking with the edge MENTIONS

## Example Data from Roll Calls

rollnumber	date	yea_count	nay_count	vote_result	vote_desc		
1	1/6/21	6	93	Objection Not Sustained			
2	1/7/21	7	92	Objection Not Sustained			
3	1/20/21	84	10	Nomination	Avril Danica Haines, of New Y		
4	1/21/21	69	27	Bill Passed	A bill to provide for an except		
5	1/22/21	93	2	Nomination	Lloyd James Austin, of Georgi		
6	1/25/21	84	15	Nomination	Janet Louise Yellen, of Califor		
7	1/26/21	78	22	Nomination	Antony John Blinken, of New		
8	1/26/21	55	45	Motion to Table	Agreed to		
9	1/26/21	83	17	Resolution A	A resolution to provide for rel		
10	1/28/21	55	42	Cloture Mot	Alejandro Nicholas Mayorkas,		
11	2/2/21	86	13	Nomination	Peter Paul Montgomery Butti		
12	2/2/21	56	43	Nomination	Alejandro Nicholas Mayorkas,		
13	2/2/21	50	49	Motion to P	A concurrent resolution settin		
14	2/4/21	90	10	Amendment	To establish a deficit-neutral		
15	2/4/21	50	50	Amendment	To establish a deficit-neutral		
16	2/4/21	100	0	Amendment	To establish a deficit-neutral		
17	2/4/21	50	50	Amendment	To establish a deficit-neutral		
18	2/4/21	58	42	Amendment	To establish a deficit-neutral		
19	2/4/21	99	1	Amendment	To establish a deficit-neutral		

**NER on  
vote\_desc  
column**

**Entity  
Nodes**



# Linking Named Entities To Contributors

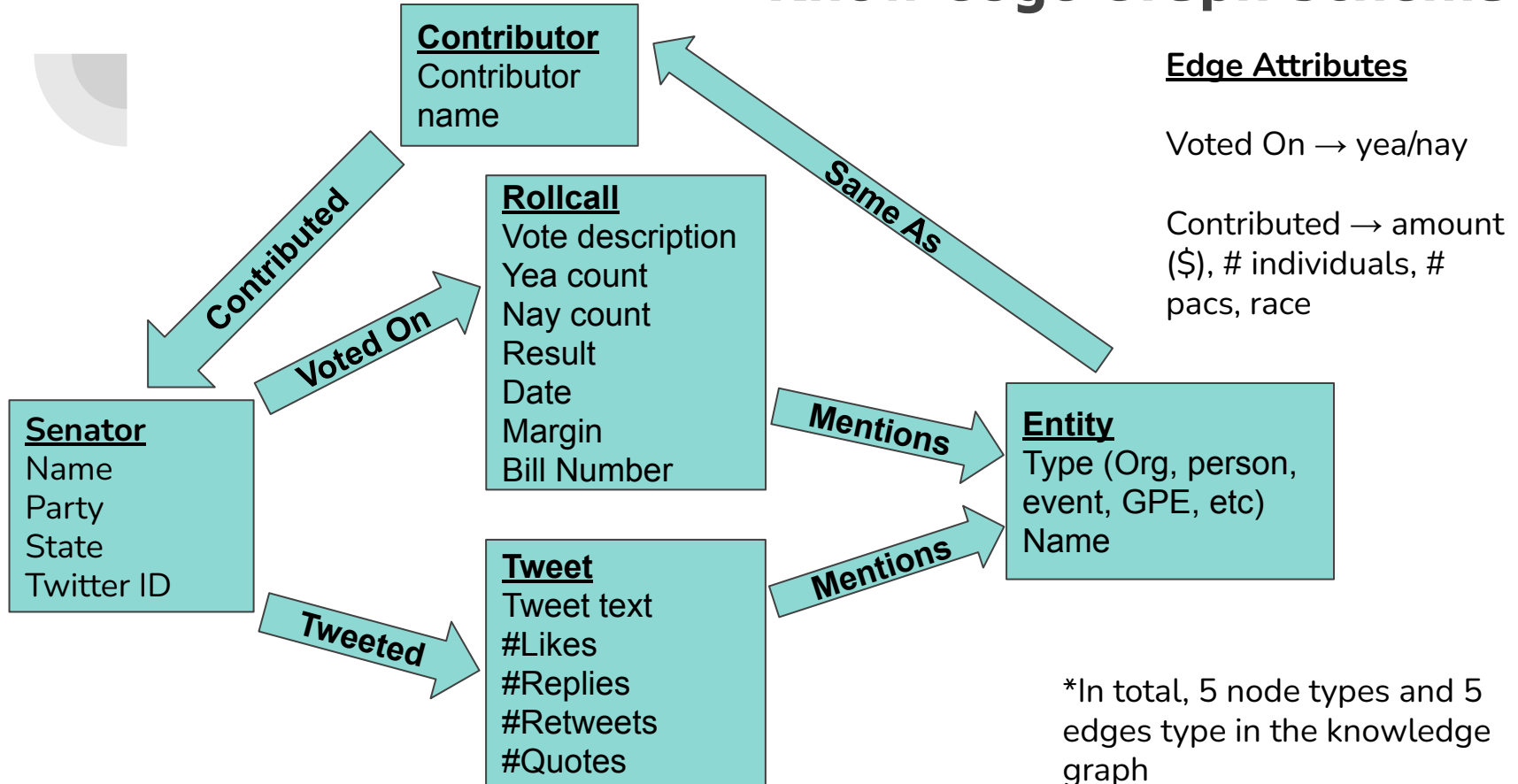
- 1) Process contributor names in contributors CSV file
  - a) Create data with removed punctuation, all lowercase, and removing org, inc, or llc
- 2) Merge roll call NER results file with twitter NER results file on the named entity
  - a) Create data with removed punctuation, all lowercase, and removing org, inc, or llc
- 3) Match the processed contributor names with the processed named entities, saving as a file
- 4) Load in the named entity - contributor matching results as relationships to the Neo4j knowledge graph
  - a) Create the edge SAME\_AS if a named entity is also a contributor



# Linking Senators to their Votes and Tweets

- 1) Using S117 senator information from voteview.com (S117\_members.csv), save the name, party, and state of each senator
- 2) Merge senator names with the Twitter account mapping
- 3) Load in the senators as their own nodes with name, party, state, and twitter account as node attributes
- 4) Senator Tweet Relationship:
  - a) MATCH each senator node to their tweet nodes creating an edge labeled TWEETED
- 5) Senator Votes on Roll calls:
  - a) Using the S117 voteview.com information (S117\_votes.csv), MATCH each senator node to the roll call node they voted on creating an edge labeled VOTED
  - b) Store whether the senator voted yea or nay as an edge attribute

# Knowledge Graph Schema





# Next Steps

- Get an API key for followthemoney, which has more detailed contributor data such as industry. Or get contributor data manually from somewhere like dbpedia.
- Increase date range of tweets to provide a larger variety of public statements
- Curate NER results further
  - There are overlaps to what may be called an Organization or what may be called a Geopolitical entity, for example
- Add sentiment predictions on tweets to allow for searching entities that are positively mentioned vs negatively mentioned
- Conduct more rigorous matching of named entities to contributors
- Add news articles mentioning bills/senators into the knowledge graph
- Figure out a way to incorporate tweets that mention other users



## Demo

We will now give a demo of our senator knowledge graph in Neo4j ...



The full implementation of our knowledge graph can be found at  
<https://github.com/Aidankeogh/SenatorGraph>

Thank you!