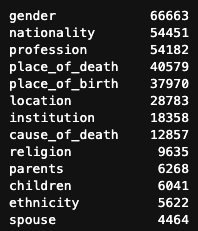
# Constructing Common Sense Benchmark

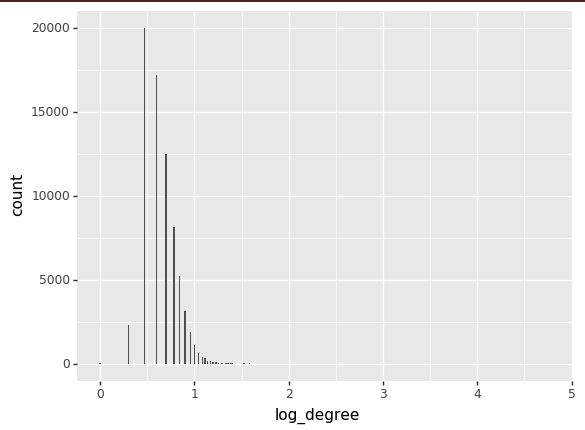
## Exploratory analysis of FB13 data

Number of edges in the graph of each relationship type:



### Overall network properties

Degree distribution:



Connected components: the graph is a single connected component. However, **if you remove the ‘male’ and ‘female’ nodes, you then get 64 distinct components**!

### Family

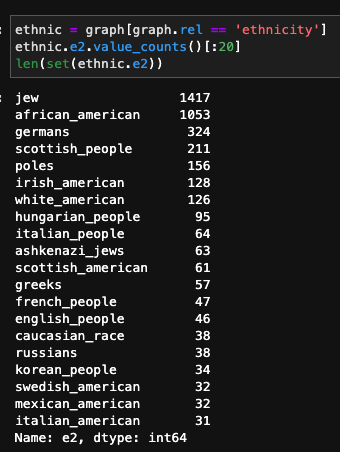
* For the ‘parents’ relationship, I observe that the tail entity is the parent, and the head entity is the child, based on checking three different edges.
* **Among the 6268 ‘parents’ edges, 5629 (90%) have the symmetric “children” relationship.**
* 3853 out of 6268 parents (~60%) have at least one spouse relationship
* **Most spouse edges are also reciprocated** (92% or 4112 out of 4464, which means 2056 married couples represented in the 4112 reciprocated edges).

### Gender

* **Most entities (66663 out of 75043) have genders**. And not all these entities are people, so most people have genders in the data.
* **The most highly connected node is ‘male’ with degree 58593. The ‘female’ node has degree 8070. (A gender bias!)**
* **Among the 4077 spouse edges where both spouses have gender annotations, 2% (95) are homosexual, and 98% are heterosexual (heterosexual bias)**

### Ethnicity

There are 5622 `ethnicity` edges to 211 distinct entities, as listed below. Note that almost 50% of these edges are either “jew” or “african\_american”. Let’s not use this, as the coverage is low and it somehow feels non-PC given that the most oppressed ethnic groups are so over-represented in the data.



### Nationality

Nationality has much stronger coverage and is a neutral attribute.



## Plan for commonsense benchmark

→ Construct various commonsense benchmarks based on natural logical rules.

* **parents:** A--[parents]-->B explained by:
  + 1-hop: *child*, B--[children]-->A. Deterministic.
  + 2-hop: Non-deterministic but highly likely.
    - *spouses\_child* {B--[spouse]-->X--[children]-->A
    - *parents\_spouse* A--[parents]-->X--[spouse]-->B}.
* **spouse:** A--[spouse]-->B explained by:
  + 1-hop: *spouse*, B--[spouse]-->A Deterministic.
  + 2-hop: *co-child*. Inferential but highly likely.
    - B--[children]-->X--[parents]-->A
    - A--[children]-->X--[parents]-->B
* **nationality** A--[nationality]-->B explained by:
  + 1-hop (all inferential):
    - A--[location]-->X
    - A--[place\_of\_birth]-->X
    - A--[place\_of\_death]-->X
  + 2-hop: (all inferential)
    - parent nationality
    - child nationality
    - spouse nationality
* **location** via:
  + 1-hop: nationality, place\_of\_birth, place\_of\_death
  + 2-hop: parent location, child location, spouse location, institution location

→ For location and nationality benchmarks, could consider additional analysis showing that richer explanations are correlated with more confident model scores.

* **parents:** A--[parents]-->B explained by:
  + 1-hop: *child* B--[children]-->A
    - **Deterministic →** No need to even check if the edge is in the graph, since this is a symmetric relationship by definition, so we’ll assume that asymmetric representations just correspond to missing data.
    - Check whether it is in train, valid, test, or none.
    - Assert that the sum of fractions is 1
  + 2-hop: *spouse* {B--[spouse]-->X--[children]-->A, A--[parents]-->X--[spouse]-->B}
    - Logic: need both in order for the explanation to make any sense. Therefore, check if both hops are in the graph, and if so, add it as an explanatory feature.
    - Check what fraction is in train, valid, and test, and assert this adds to 1.
    - Note that it is possible to have multiple paths of either type if someone has multiple spouses. But probably less likely.
    - Note also that you can infer A’s spouse based on other co-children, and this can be used as well. But using inferred links, this can still be a 2-hop feature, so let’s not worry about it.

Questions:

* How does link prediction even make sense with such a sparse graph?
* How should this interplay with the cross-validation split?
  + Start by assessing performance on the explanation benchmarks, distinguishing whether the explanation paths are fully in the training data, partially, or not at all. If there is an affect, reassess whether we should construct the benchmark in a way that somehow respects the cross-validation split

To think about:

* + Distinguish simple cases to explain (single path in the data which is the correct path) vs. more highly connected nodes

## Loading data into Neo4j

Files to be imported must be in this Import directory: /Users/rhodos/Library/Application Support/Neo4j Desktop/Application/neo4jDatabases/database-a664751e-d7f6-4bc3-b91c-37e1d0a69027/installation-3.5.26/import

See neo4j/README.txt for more information.

## TO DO

* Note that the location edge can link one person to multiple locations.
* Compute connectivity or other graph features that might explain variation in performance (in either the base link prediction or in the explanation performance). Maybe hop node counts?

## Splitting the benchmark data

Goals:

* We would like to have multiple splits of the explanation benchmarks, divided into validation and testing purposes.
* We would also like the number of examples to be relatively manageable, at least as we prototype everything, so that Gustavo and Niraj can iterate quickly. How about we randomly split into 200 valid, and the remainder test. And then I’ll take the first 25 of each into quick\_valid.txt and quick\_test.txt.

We are also interested in questions of whether the quality of the explanations vary depending on whether:

* the query triple was in the training set
* the explanation paths were in the training set (note that the set of paths could have full, partial, or no presence in the training data)

For the first question, we are already setup to answer this based on our current split. \*\*The only thing we would need here is an explanation benchmark where query triples are coming from from the training set.

For the second question, let’s look into the original train/test split, and the new one.

1. Can I confirm that currently, the explanatory paths are entirely within train? **Yes, this is true for both parents and location!**
2. If so, why is this? Are the specific relations that we are focused on just not in the test or valid set? **Yes! Actually the data were not split correctly the first time. The training edges were just copied into the test set. Also the validation set doesn’t contain the types of relations we care about.**
3. The plan for re-splitting:
   1. Take the entire graph.txt, shuffle the triples, and split between train, valid and test (80/10/10)
   2. Extract the largest connected component based on the training triples, and then restrict the valid and test to only include those entities

