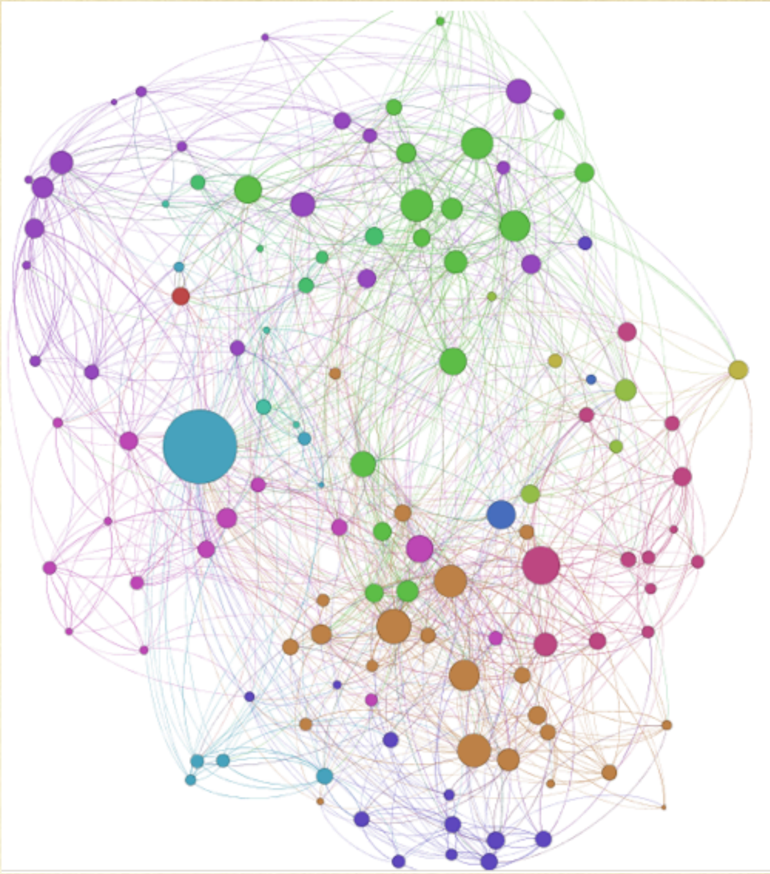


RelaViz

Graph Visualization of Learned Relations Between Entities

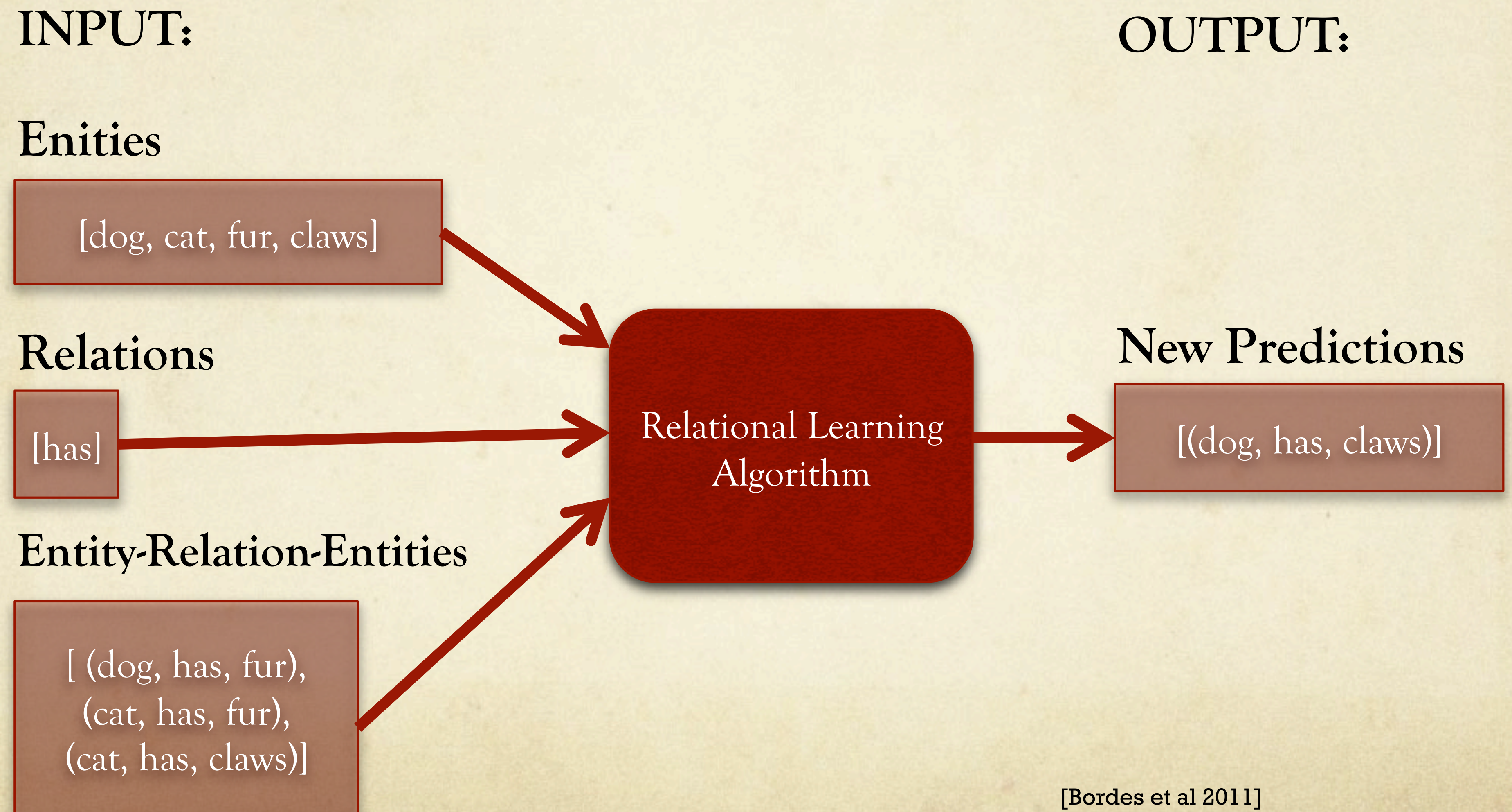
Joel Ferstay - joelaf@cs.ubc.ca

RelaViz: Design Study Project

- Apply Visualization to Assess the Performance of a Machine Learning Algorithm
- Type of Algorithm: **Relational Learning Algorithm**
- Why? Help algorithm designers verify whether or not the relations their algorithms predict **"make sense"**

Part I: The Domain

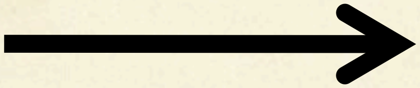
Overview: A Relational Learning Algorithm



[Bordes et al 2011]

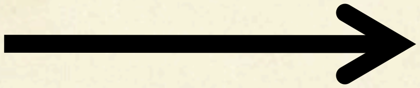
What's Relational Data?

Entity: Cat **Relation: Has** **Entity: Fur**



Relational Data is Directional

Entity: Cat **Relation: Has** **Entity: Fur**



Relational Data is Directional

Entity: Fur **Relation: Has** **Entity: Cat**

Relational Data is Directional

Entity: Fur **Relation: Has** **Entity: Cat**

NO

How Do We Encode Entity-Relation-Entity Data?

- As a triplet ($\mathbf{e}^l$, $\mathbf{r}$, $\mathbf{e}^r$)
- This is a compact, specific encoding which ensures the directionality of the relationship is preserved
- For instance: (cat, has, fur) YES ; (fur, has, cat) NO

[Bordes et al 2011]

What is Relational Learning?

- Given sets of **entities, relations,** and **entity-relation-entity** data, learn to predict new **entity-relation-entities**

We're Given:

Entity:

Cat

Relation: Has

Entity:

Fur

We’re Given:

Entity:

Cat

Relation: Has

Entity:

Fur

Entity:

Dog

Relation: Has

Entity:

Fur

We're Given:

Entity:

Cat

Relation: Has

Entity:

Fur

Entity:

Dog

Relation: Has

Entity:

Fur

Entity:

Cat

Relation: Has

Entity:

Claws

We Predict:

- Given the existing entity-relation-entity the algorithm learned from before, might predict:

Entity:

Dog

Relation: Has

Entity:

Claws

Why Do We Want To Learn Relations?

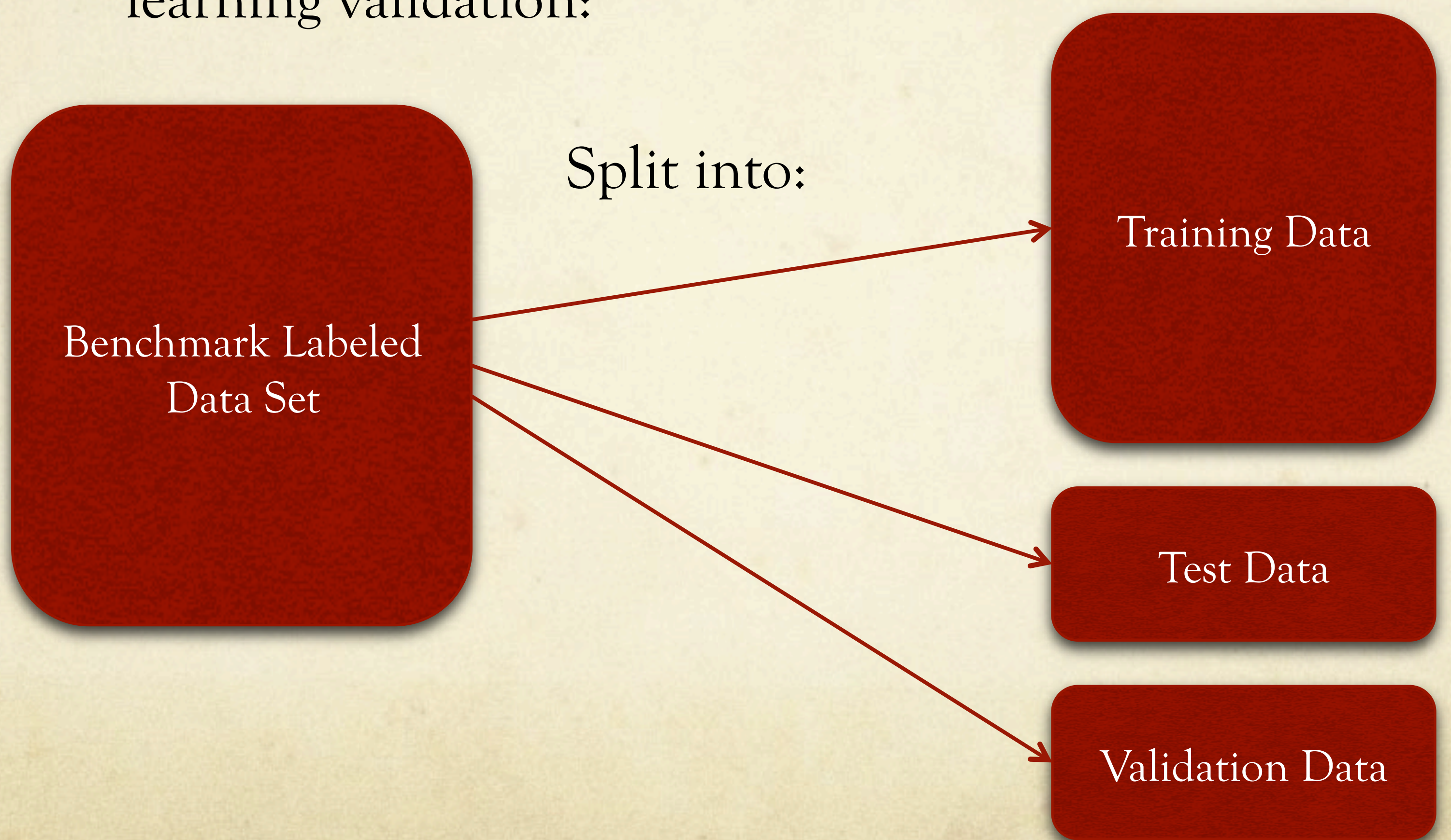
- Another step towards building thinking machines with "common sense"
- It is an advantage to be able to predict things ahead of time – we have predicted "furry things" usually have claws
- When our agent encounters an unfamiliar "furry thing", it does not have to walk up and inspect its paws to see if it has claws

[Bordes et al 2011]

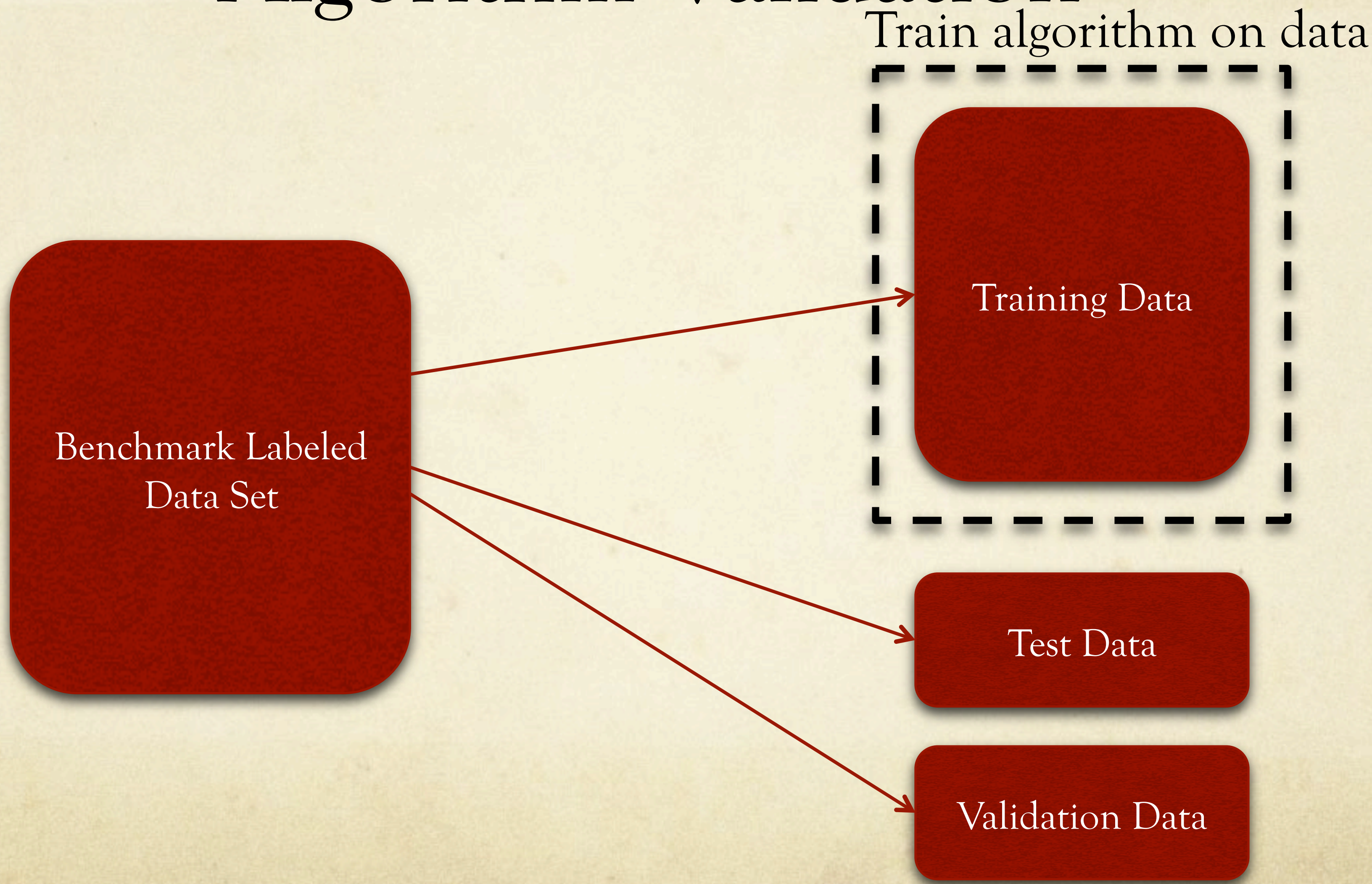
Part II: Why Visualization for Validation?

Current Machine Learning Algorithm Validation

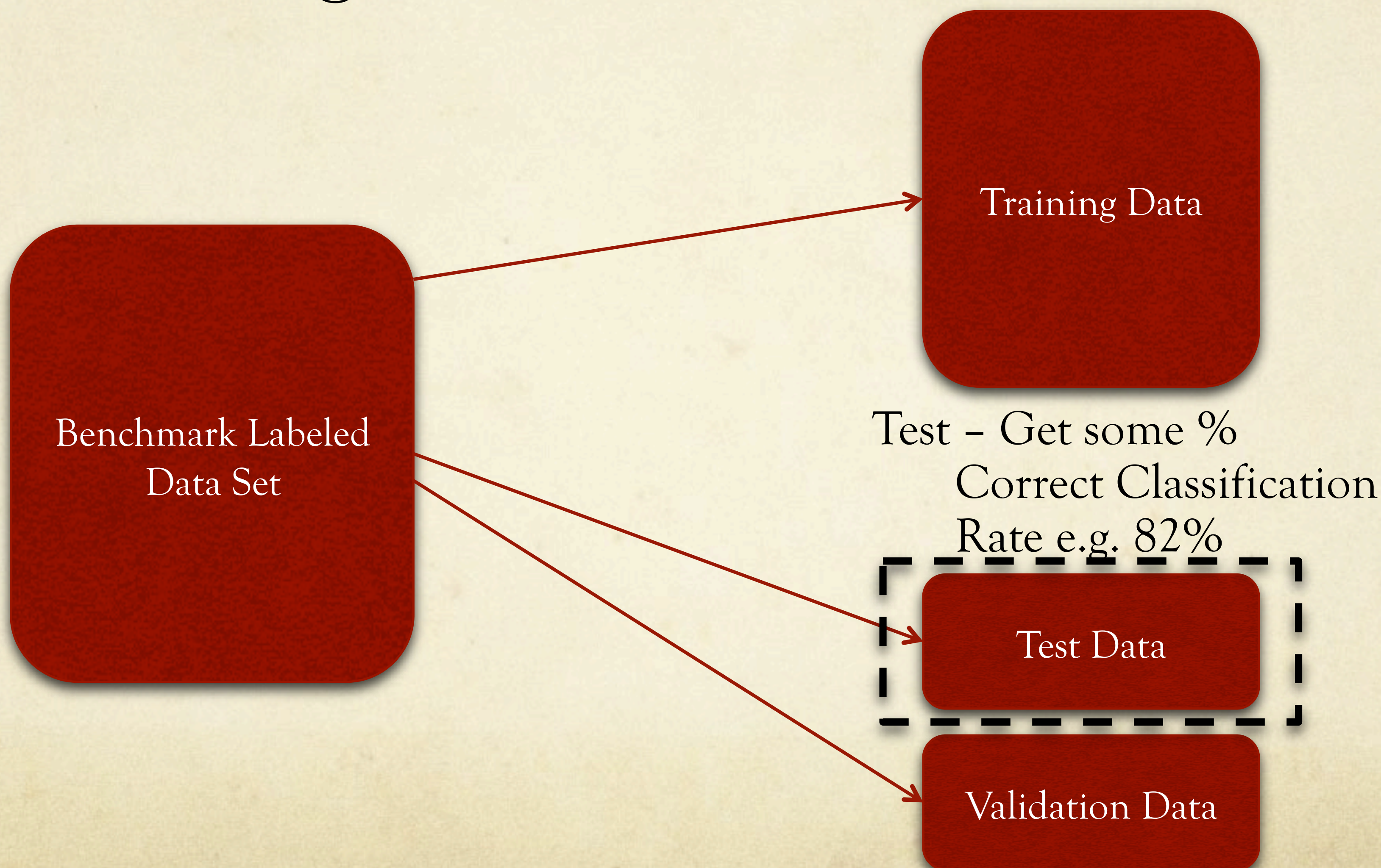
- Current state of affairs for conventional machine learning validation:



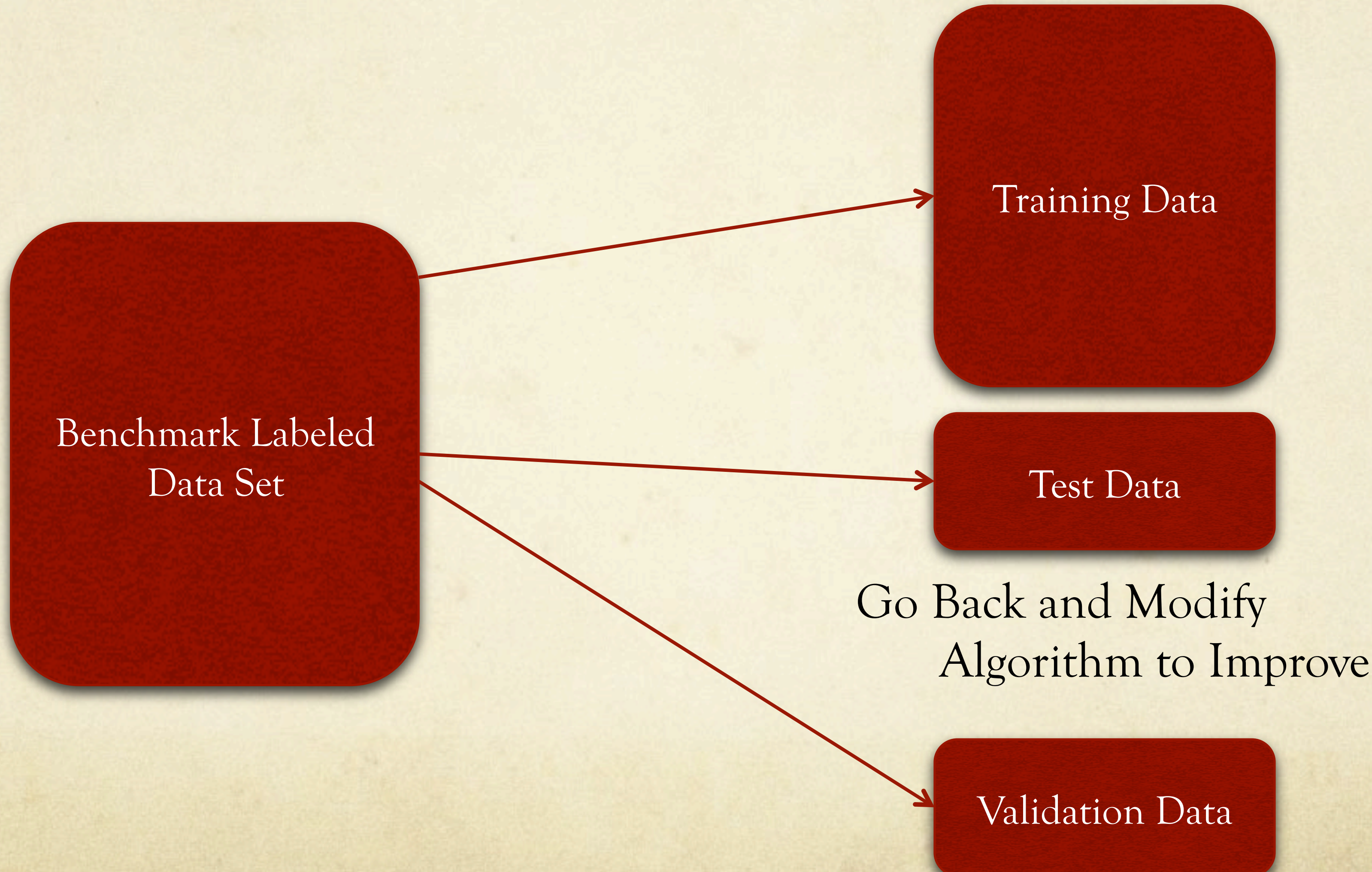
Current Machine Learning Algorithm Validation



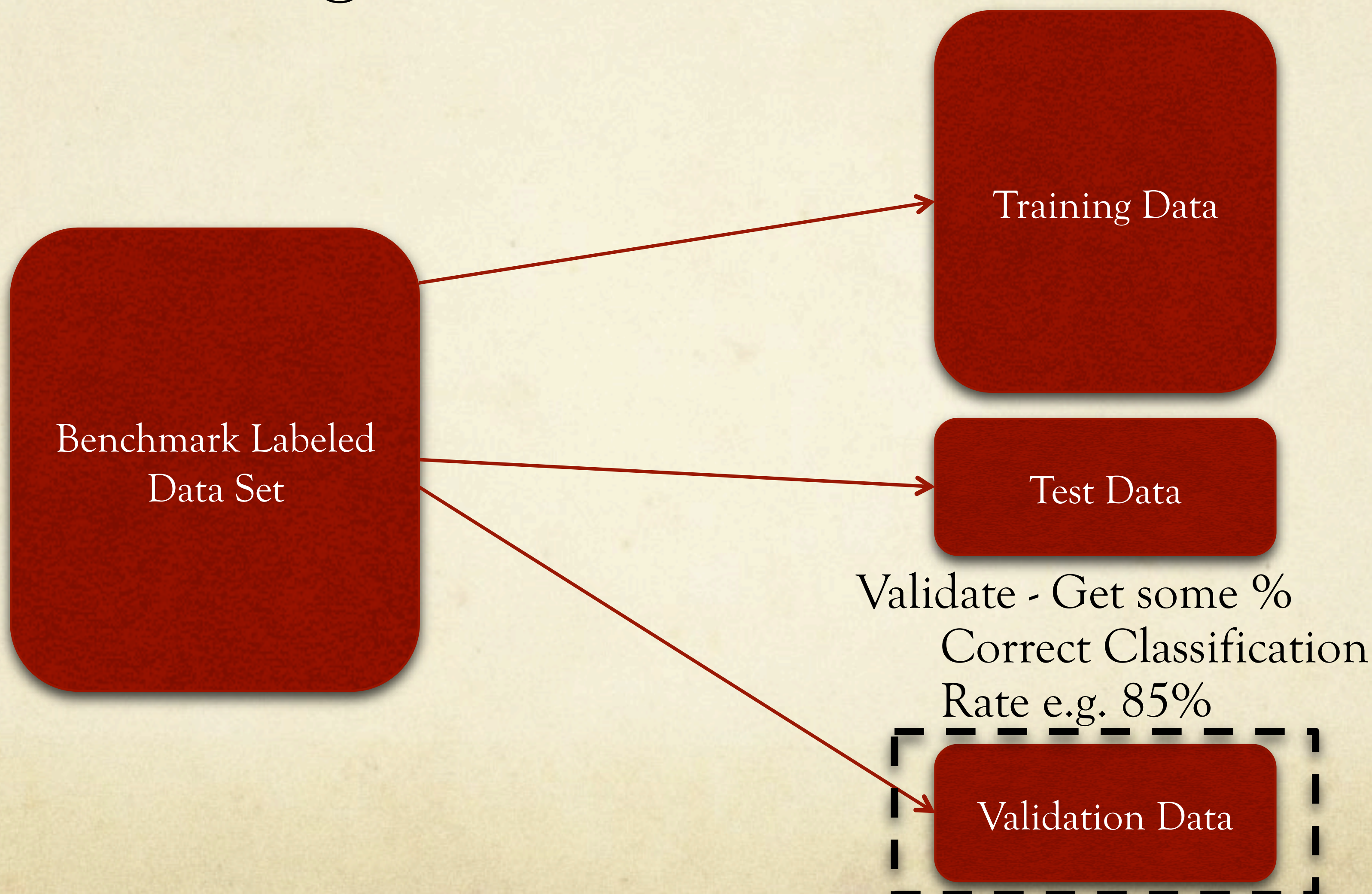
Current Machine Learning Algorithm Validation



Current Machine Learning Algorithm Validation



Current Machine Learning Algorithm Validation



Machine Learning Algorithm Validation

- Conventionally, we have an automated way of getting a quantitative, percentage measure of new relations we've learned that were "correct"
- E.g. Classification Rate 83.7 %
- So … what's the problem?

[Bordes et al 2011]

Why Visualization?

- There is no problem if we just care about assessing performance on a painstakingly annotated benchmark data set by only looking at a classification rate

- However, if we want a richer understanding of identity of the new relations, and the degree of uncertainty associated with those relations, we need something more…

Why Visualization?

- Here's a current approach for visualizing new relations learned by a relational learning algorithm:

Table 7: **Knowledge extraction**. Examples of lists of e^r predicted with the embeddings learnt out of raw text for e^l ="*people*". Lists are displayed by decreasing triplet probability density order.

e^l	people				
r	build	destroy	won	suffer	control
e^r	*livelihoods*	*icons*	*emmy*	*sores*	*rocket*
	homes	*virtue*	*award*	*agitation*	*stores*
	altars	*donkeys*	*everything*	*treatise*	*emotions*
	houses	*cowboy*	*standings*	*eczema*	*spending*
	ramps	*chimpanzees*	*pounds*	*copd*	*fertility*

[Bordes et al 2011]

Why Visualization?

- Presenting the data in table format will not scale up to the visual inspection of many entity-relation-entity data and many relations between entities.

Table 7: **Knowledge extraction**. Examples of lists of e^r predicted with the embeddings learnt out of raw text for e^l ="*people*". Lists are displayed by decreasing triplet probability density order.

e^l	people				
r	build	destroy	won	suffer	control
e^r	*livelihoods*	*icons*	*emmy*	*sores*	*rocket*
	homes	*virtue*	*award*	*agitation*	*stores*
	altars	*donkeys*	*everything*	*treatise*	*emotions*
	houses	*cowboy*	*standings*	*eczema*	*spending*
	ramps	*chimpanzees*	*pounds*	*copd*	*fertility*

[Bordes et al 2011]

Why Visualization?

- There is no truly expressive tools for assessing a relational learning algorithm's performance, at a fine grain of detail.
- There is no ability to automate validation of new, learned relations on un-encountered data sets
- Therefore, there is a need for an effective visualization tool

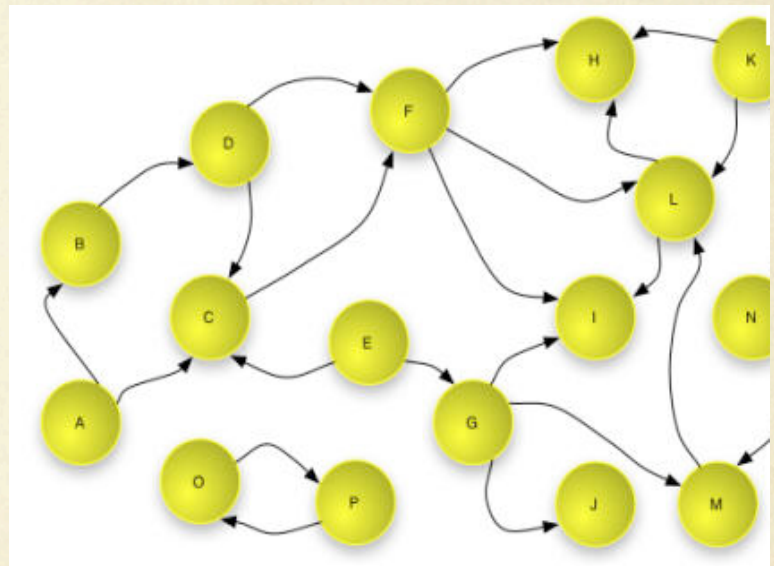
Part III: The Visualization Solution

Task Characterization

- Ideally, relational learning algorithm designers should be able to:
    - Examine the identity of learned relations between two entities
    - Examine the directionality of the relation between two entities
    - Examine the probability that a new, learned relation is true
    - Explore the difference between the known, training relations, and the new, learned relations for patterns

Data Abstraction

- Node-link Graphs
- Graphs are a natural choice for visualizing relational data
- High visibility for multiple relations, and directionality of relations between entities



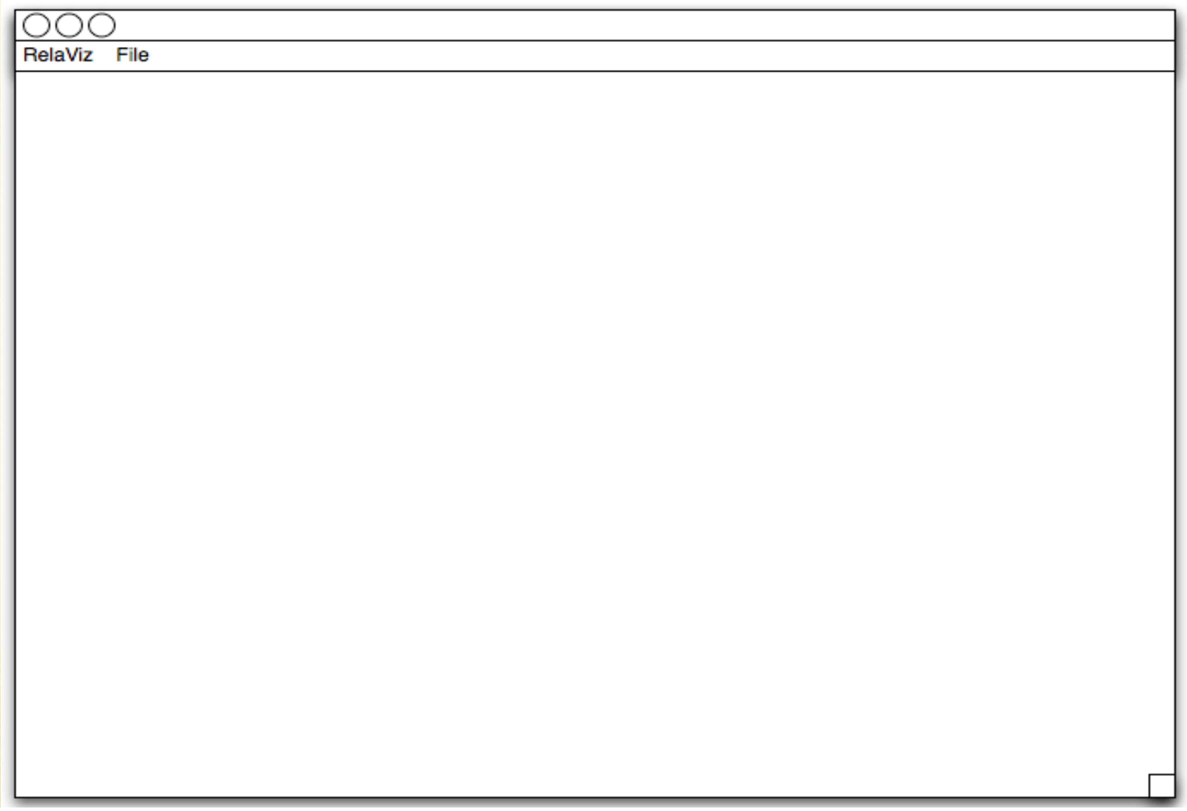
My Project: RelaViz Visualization Tool

- A visualization tool for inspecting predicted relations between entities produced by a machine-learning algorithm.

- This activity is necessary to determine whether relations match reality, and make intuitive sense.

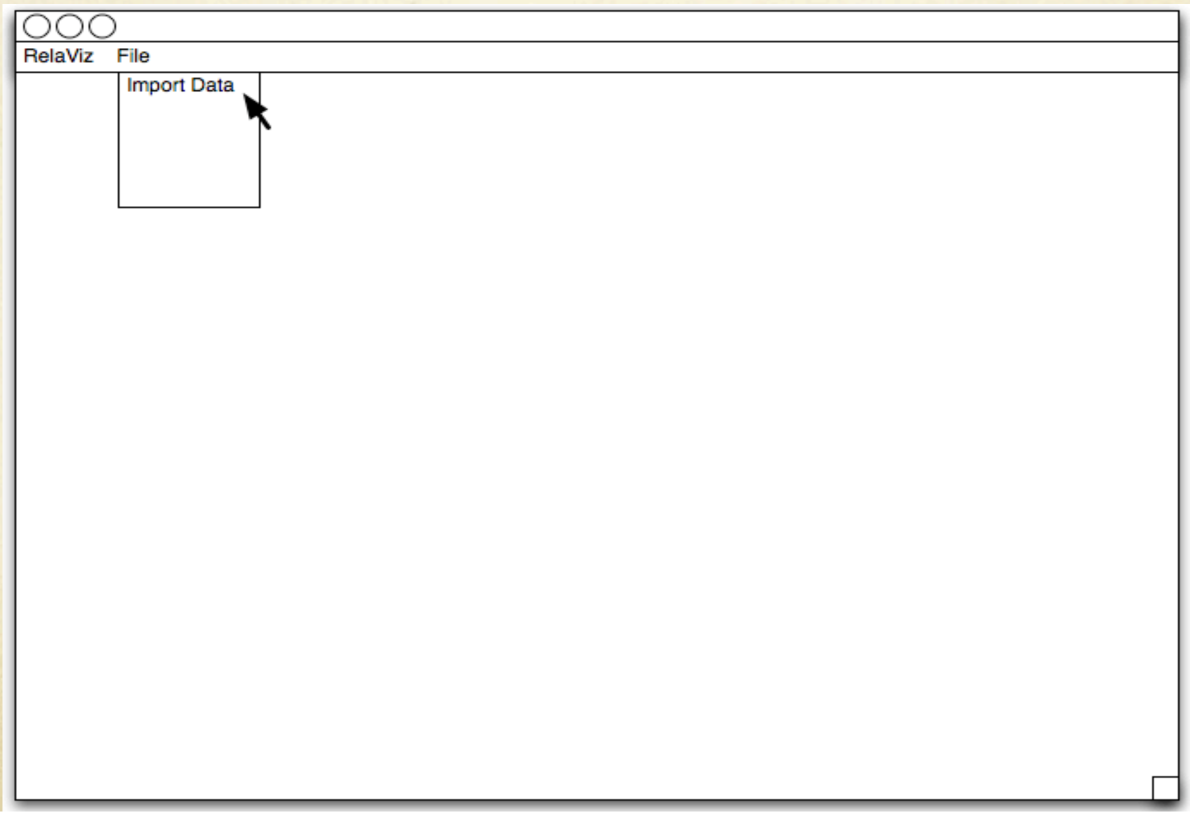
RelaViz: Scenario Of Use

- The goal is to inspect the relations between entities at a finer degree of resolution. The user will see the following screen upon initializing the system:



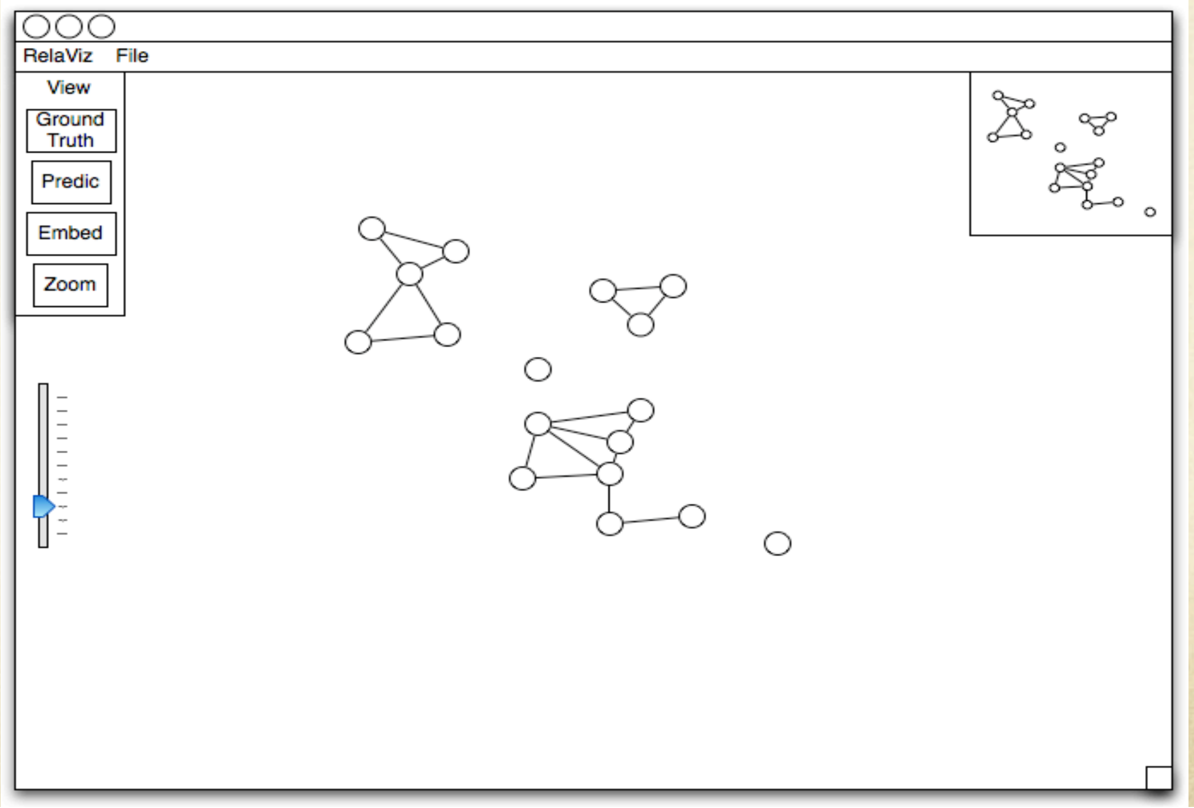
RelaViz: Scenario Of Use

- Select File > Import Data to evoke the OS's native file browser to load graph data:



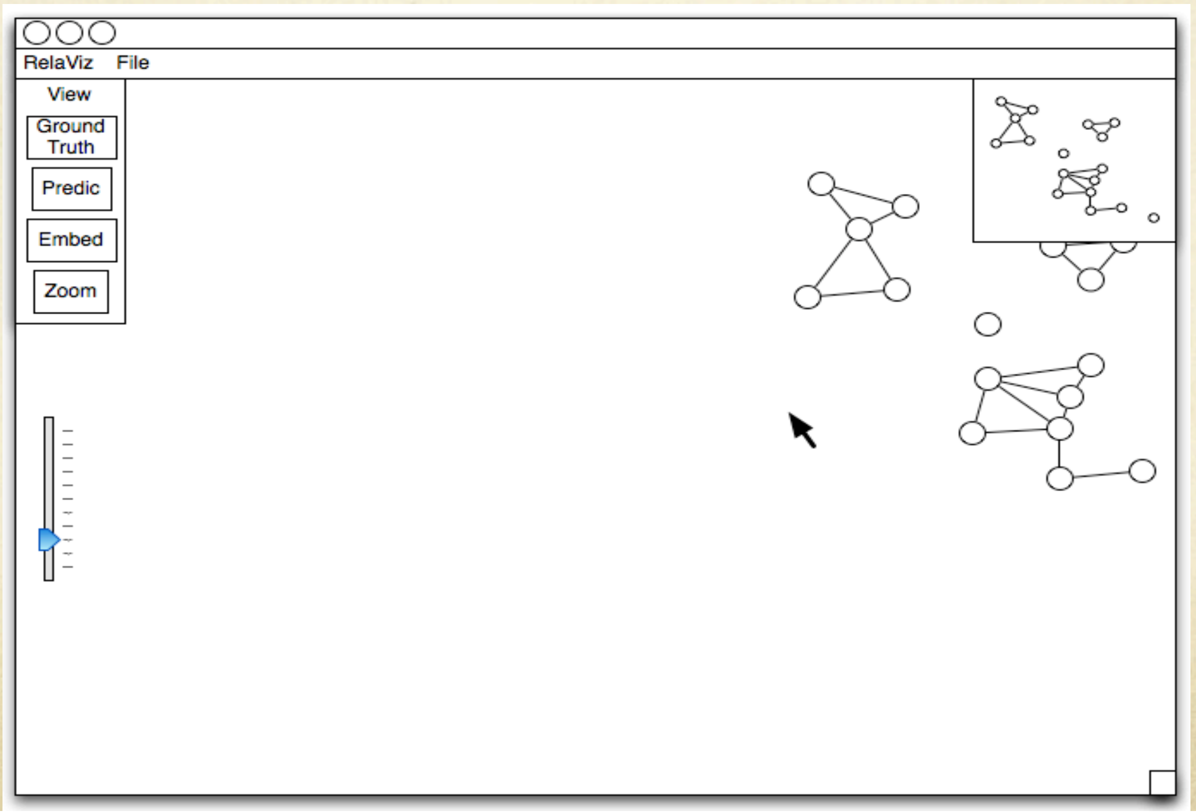
RelaViz: Scenario Of Use

- Graph will appear in the window, along with several options and a zoom slider controller to the left of the window. A graph overview is also present:



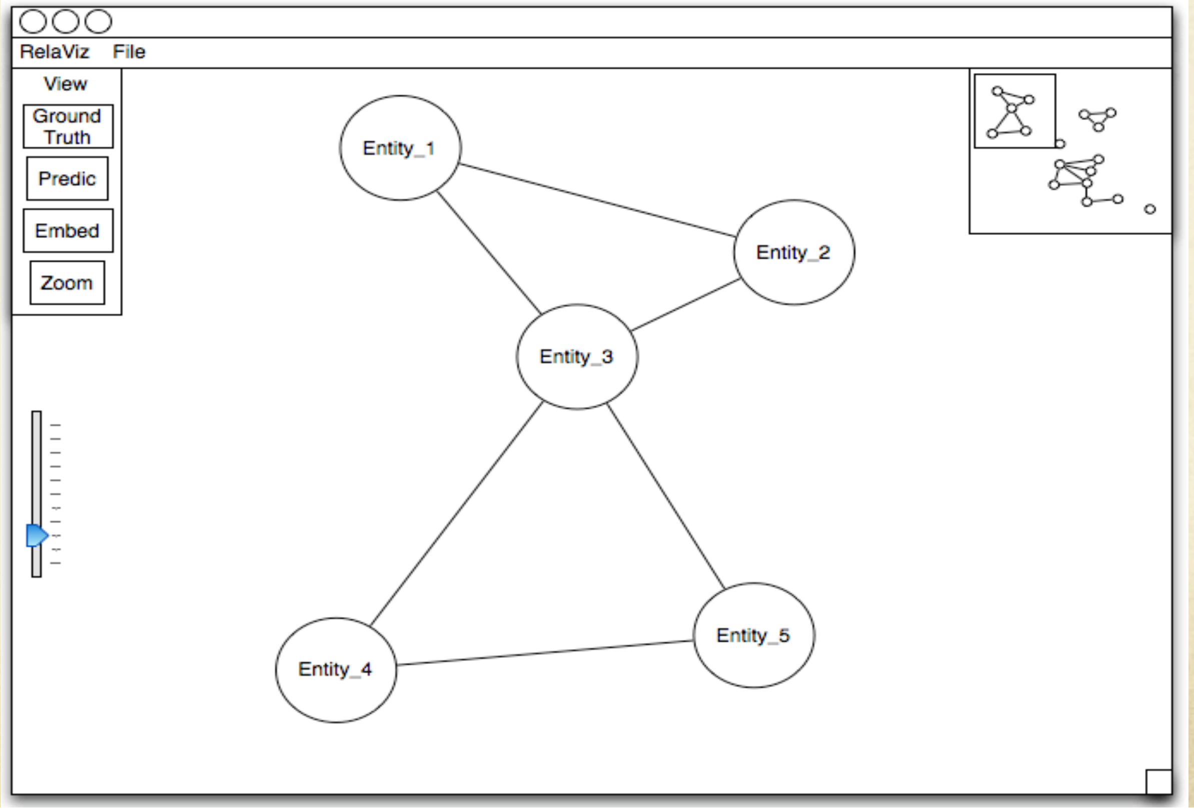
RelaViz: Scenario Of Use

- Can click and drag whitespace to navigate graph:



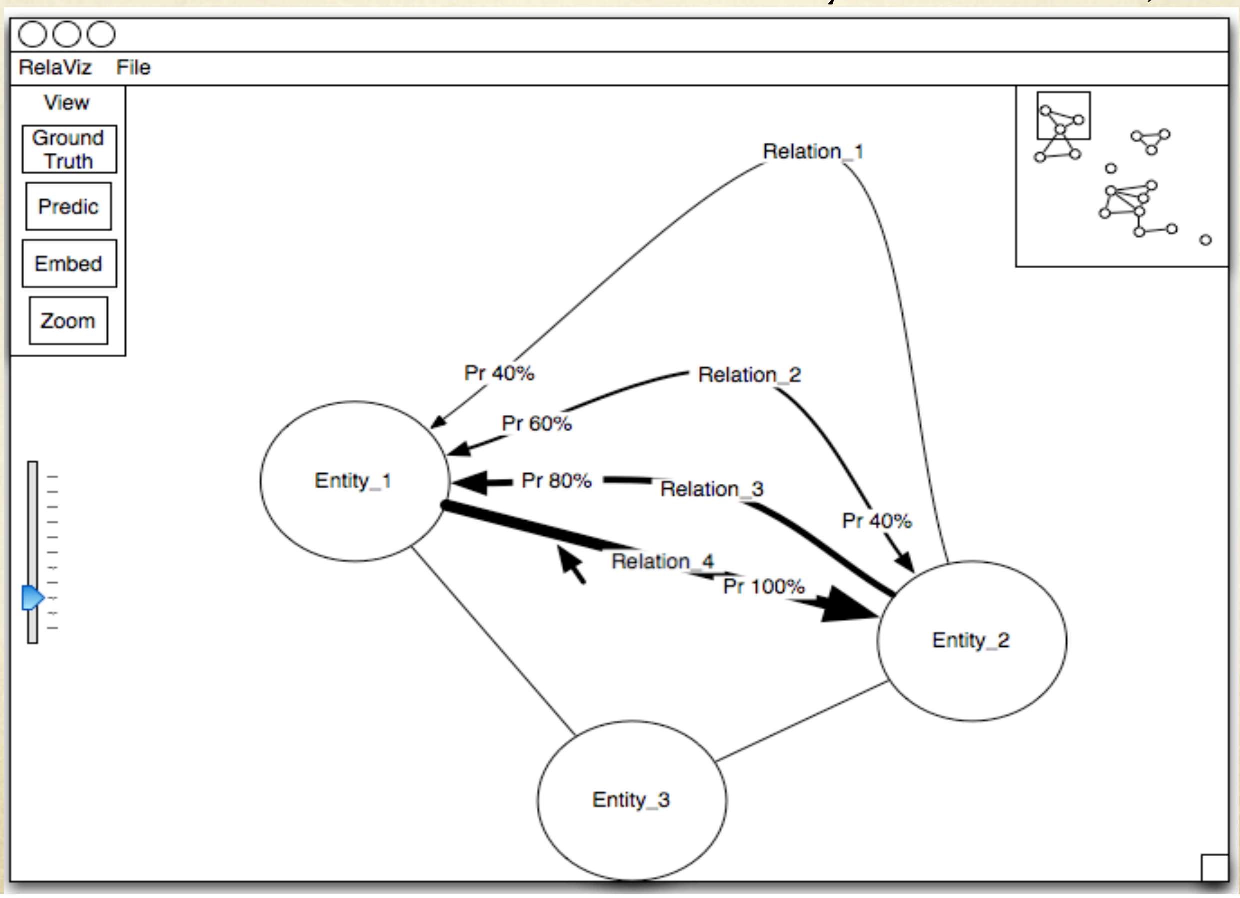
RelaViz: Scenario Of Use

- To zoom in on a particular area of interest, we can select the “Zoom” button on the left of the display, and click on an area of interest:



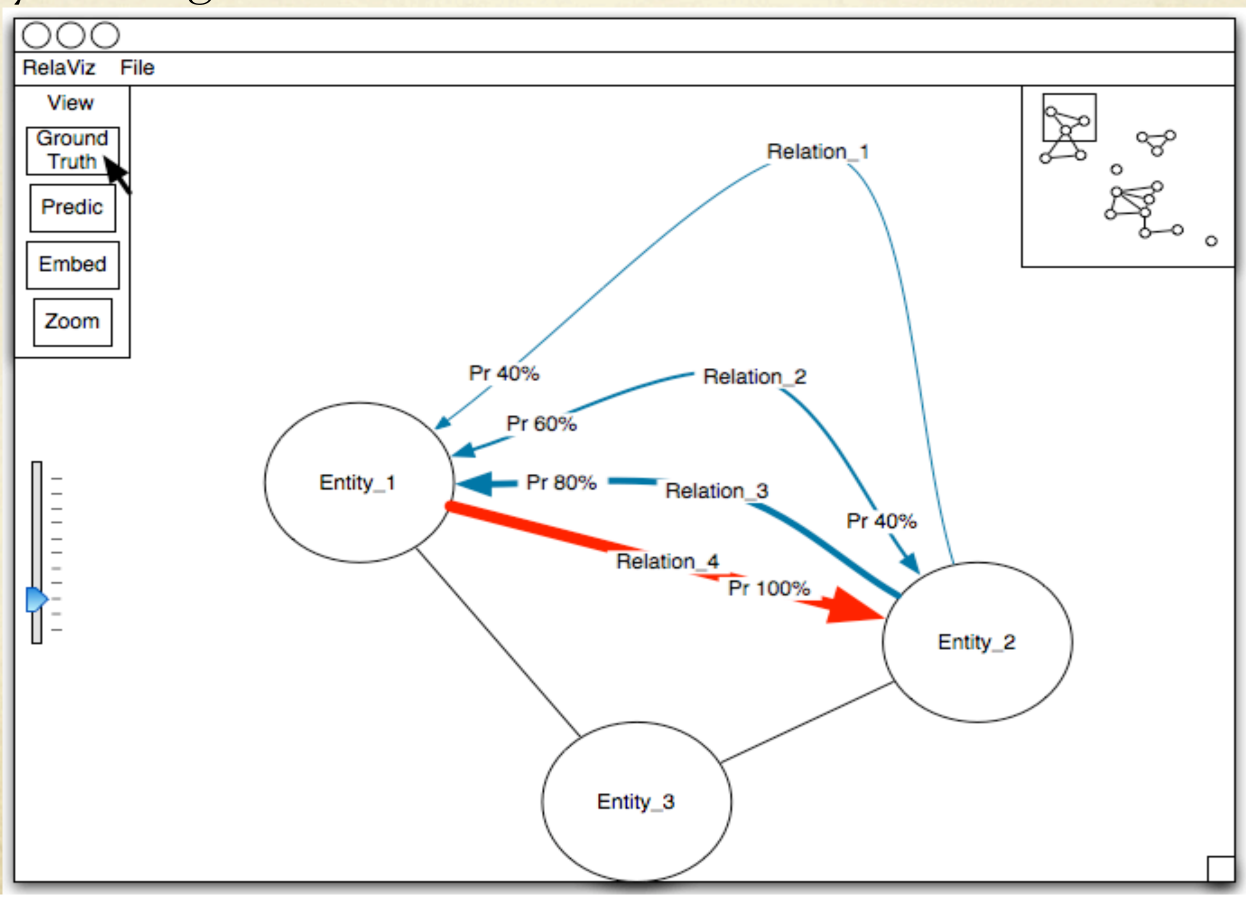
RelaViz: Scenario Of Use

- Placing the cursor on the link will split it into multiple links indicating the various relations between the two entities. This shows the uncertainty of relations, too.



RelaViz: Scenario Of Use

- Selecting “Ground Truth” indicates the true, known relational links in red, and shows the relations learned by the algorithm in blue.



Part III: Project Update

Project Update

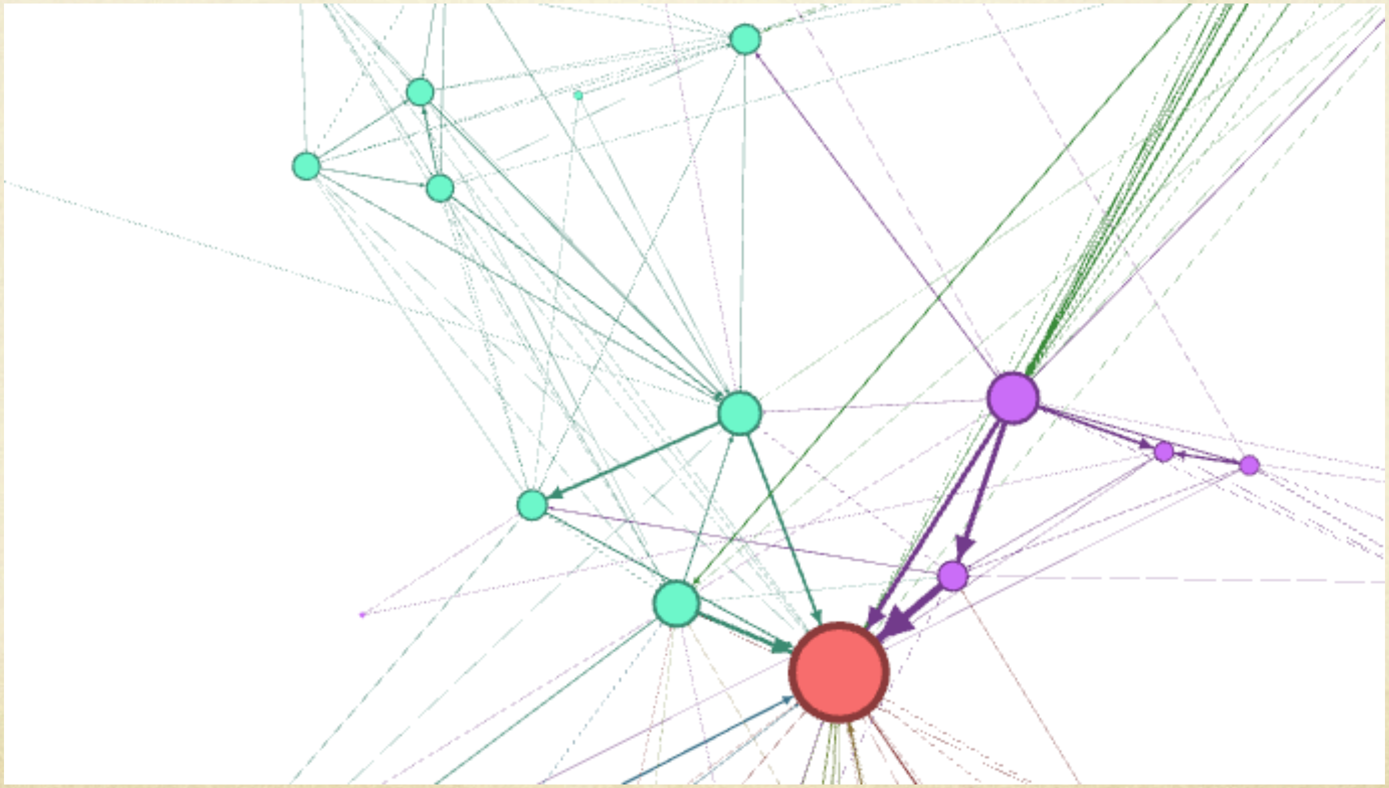
- **Note**: Joint project with CPSC 540: Machine Learning.
- Still in the process of ironing bugs out of my implemented relational learning algorithm
- To deal with this, I have produced synthetic data as a stand in for visualization in the meantime:
  - Example data point: [1,1,2,0.35]
  - Translation: Entity 1, Relation 1, Entity 2 with 35% confidence

Project Update

- Have assessed the strengths and weaknesses of the massive graph graph yoolkits GraphViz, Gephi, and Tulip
- Have selected Gephi as the toolkit for implementing this project
- Able to load synthetic data into Gephi, and display a directed graph in a simple display
- No link splitting, or entity and relation labeling yet

Gephi Toolkit Sample Implementation

- Example display of entities and directed relations between them using Gephi:

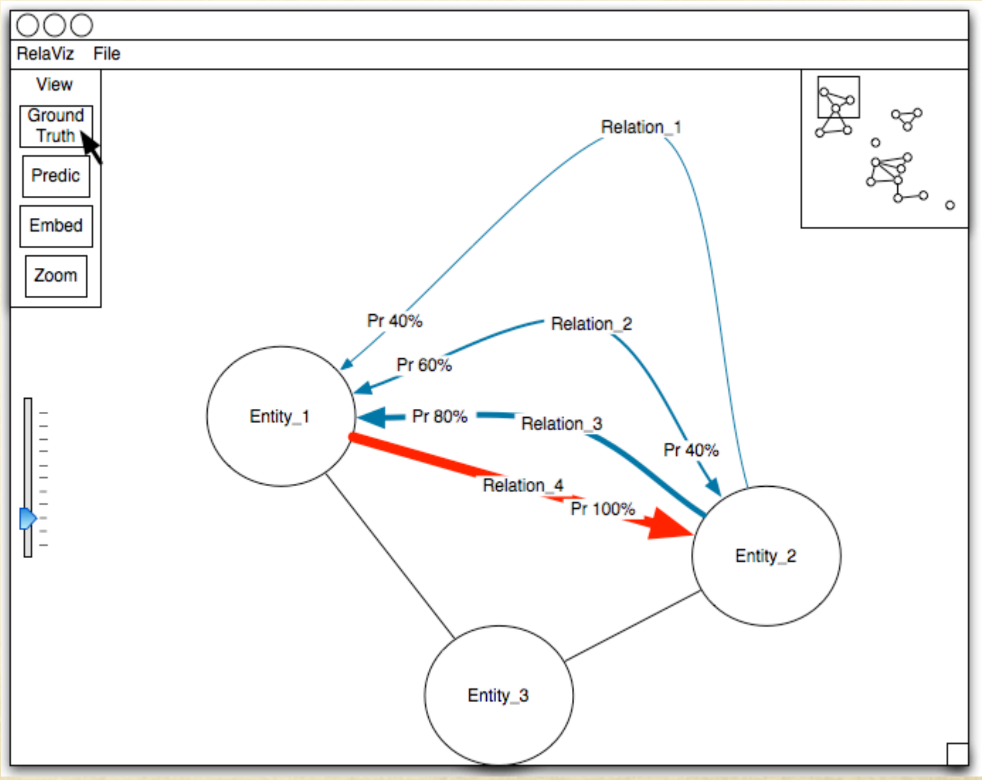


Part IV: Discussion of Current Challenges

Back To Visual Encodings

- Still iterating over how to visualize relational links

- Current Approach:



Back To Visual Encodings

- **BUT**, In a recent paper evaluating link types, TAPERED and ANIMATED edges were found to be the most effective over standard arrow:

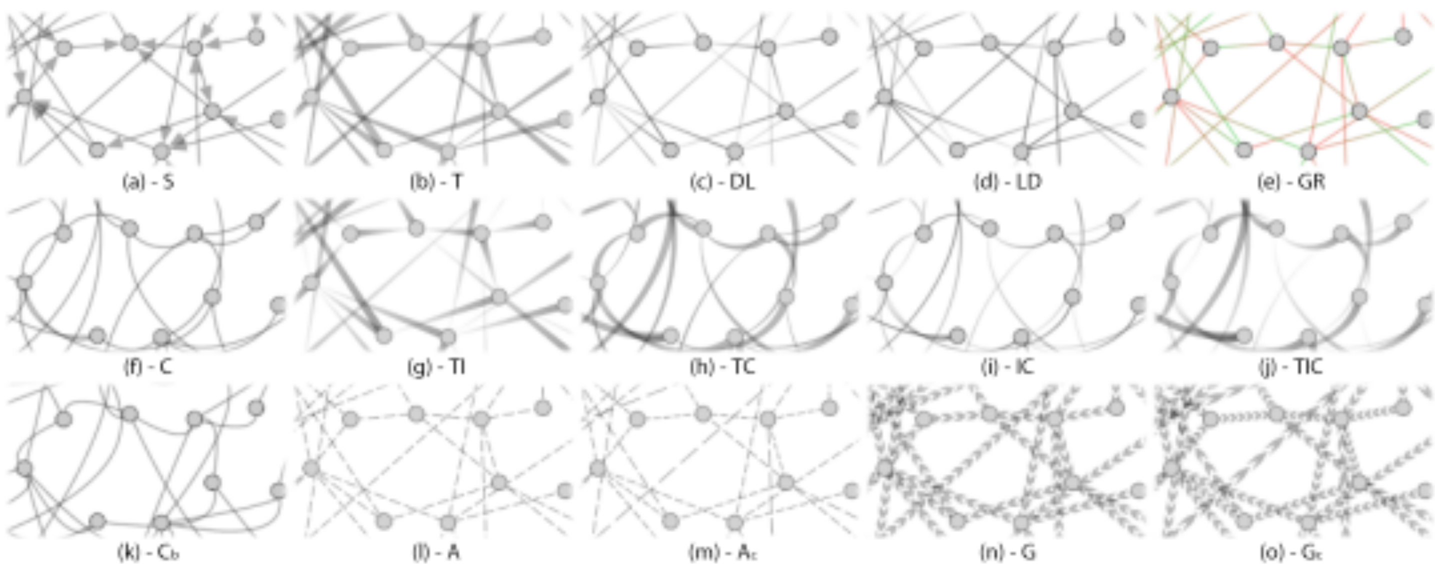


Figure 1: All directed-edge representations used in our initial (a to j), follow-up (b, k, l), and current study (b, l, m, n, o). (a) standard arrow – S, (b) tapered – T, (c) dark-to-light – DL (a.k.a intensity – I), (d) light-to-dark – LD, (e) green-to-red – GR, (f) curvature – C, (g) tapered-intensity – TI, (h) tapered-curvature – TC, (i) intensity-curvature – IC, (j) tapered-intensity-curvature – TIC, (k) biased curvature – C_b, (l) animated – A, (m) animated compressed – A_c, (n) glyph – G, and (o) glyph compressed – G_c.

[Holten et al 2011]

Back To Visual Encodings

- BUT, In a recent paper evaluating link types, **TAPERED** and ANIMATED edges were found to be the most effective over standard arrow:

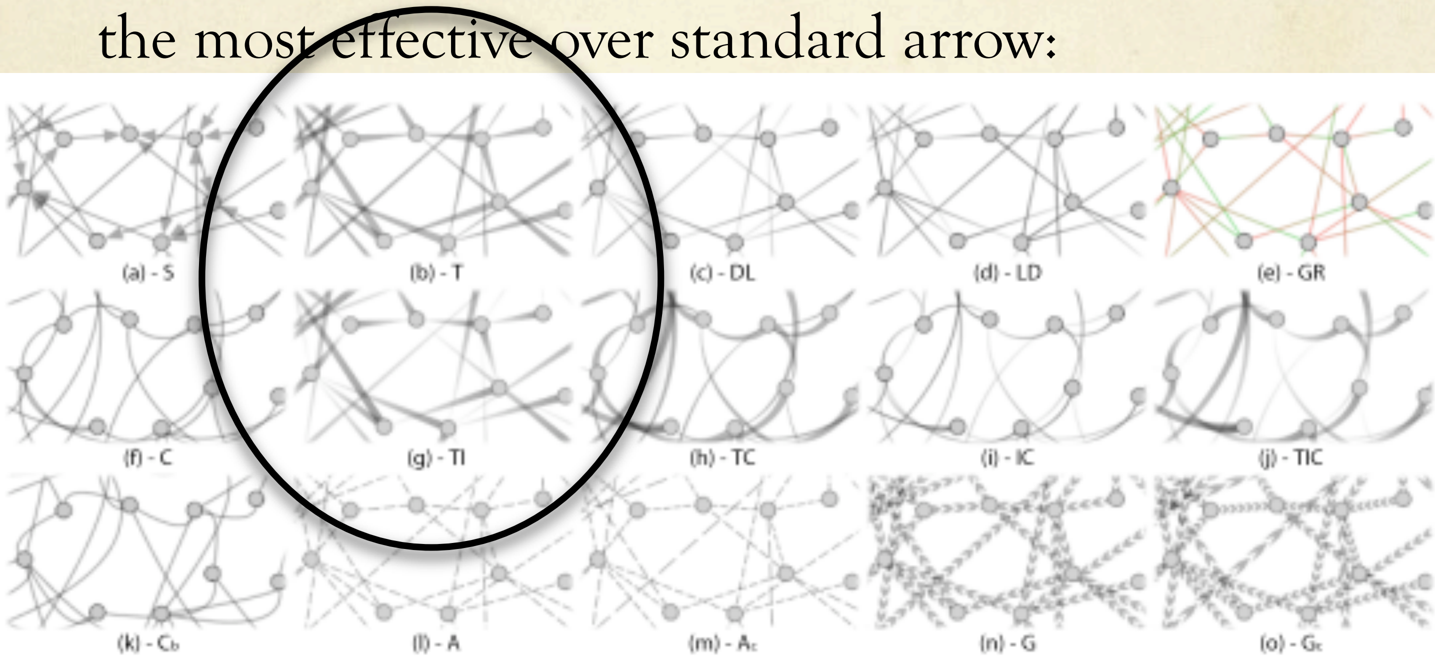


Figure 1: All directed-edge representations used in our initial (a to j), follow-up (b, k, l), and current study (b, l, m, n, o). (a) standard arrow – S, (b) tapered – T, (c) dark-to-light – DL (a.k.a intensity – I), (d) light-to-dark – LD, (e) green-to-red – GR, (f) curvature – C, (g) tapered-intensity – TI, (h) tapered-curvature – TC, (i) intensity-curvature – IC, (j) tapered-intensity-curvature – TIC, (k) biased curvature – C_b, (l) animated – A, (m) animated compressed – A_c, (n) glyph – G, and (o) glyph compressed – G_c.

[Holten et al 2011]

Back To Visual Encodings

- BUT, In a recent paper evaluating link types, TAPERED and **ANIMATED** edges were found to be the most effective over standard arrow:

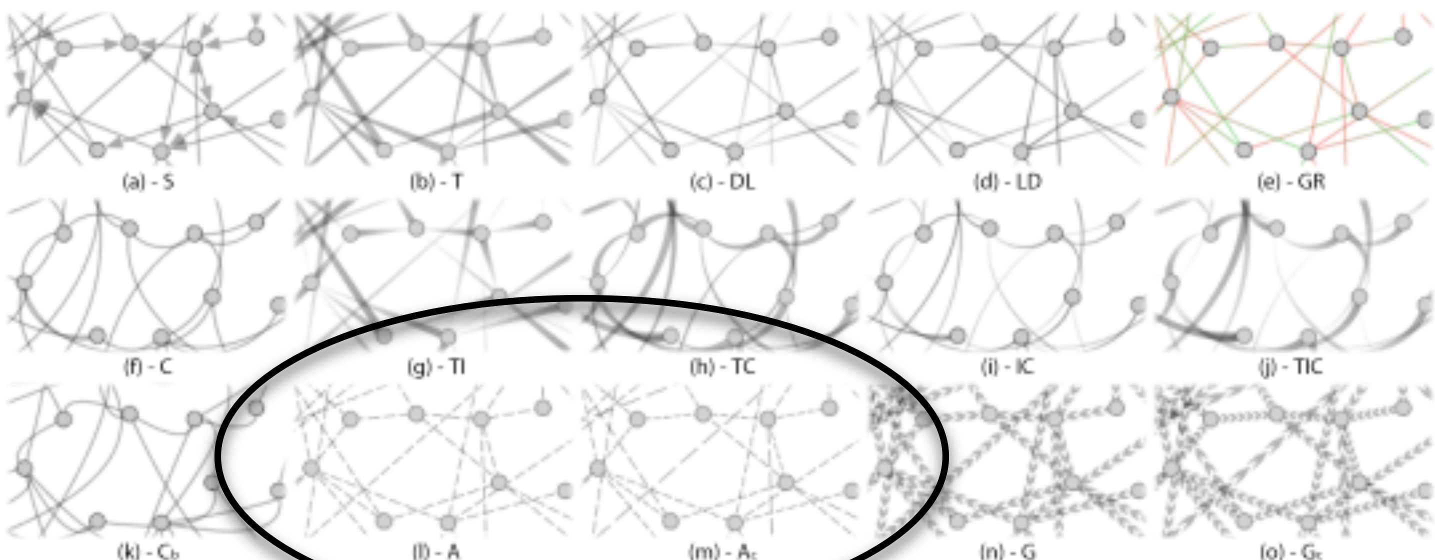
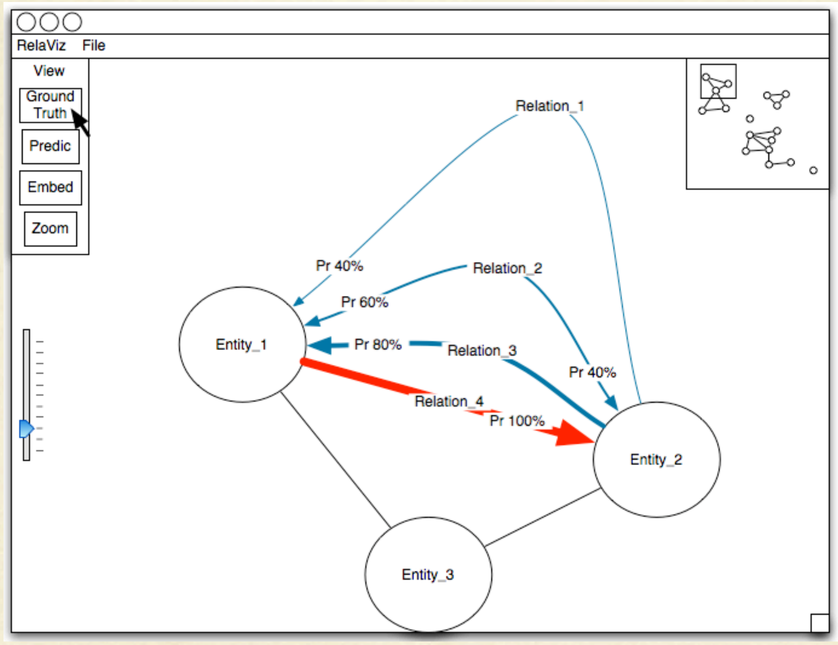


Figure 1: All directed-edge representations used in our initial (a to j), follow-up (b, k, l), and current study (b, l, m, n, o). (a) standard arrow – S, (b) tapered – T, (c) dark-to-light – DL (a.k.a intensity – I), (d) light-to-dark – LD, (e) green-to-red – GR, (f) curvature – C, (g) tapered-intensity – TI, (h) tapered-curvature – TC, (i) intensity-curvature – IC, (j) tapered-intensity-curvature – TIC, (k) biased curvature – C_b, (l) animated – A, (m) animated compressed – A_c, (n) glyph – G, and (o) glyph compressed – G_c.

[Holten et al 2011]

Back To Visual Encodings

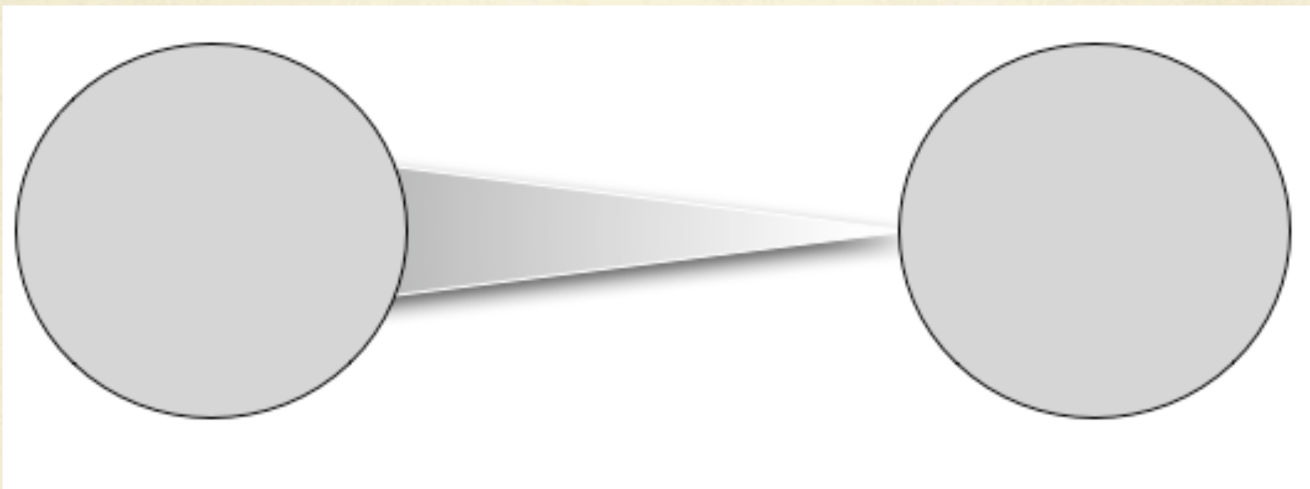
- I've unfortunately chosen **standard arrow** with **curve**, which, separately, scored relatively less in the evaluation paper:



- My problem now may be to find bi-directional analogues for the tapered and animated links they favor, or look elsewhere

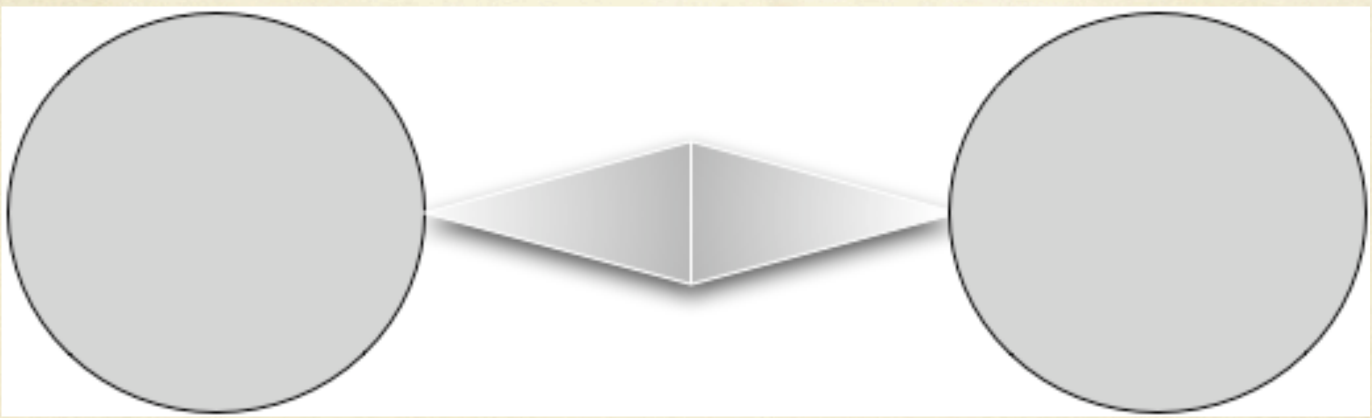
Back To Visual Encodings

- What's your opinion?
- Given that this is a directed edge:



Back To Visual Encodings

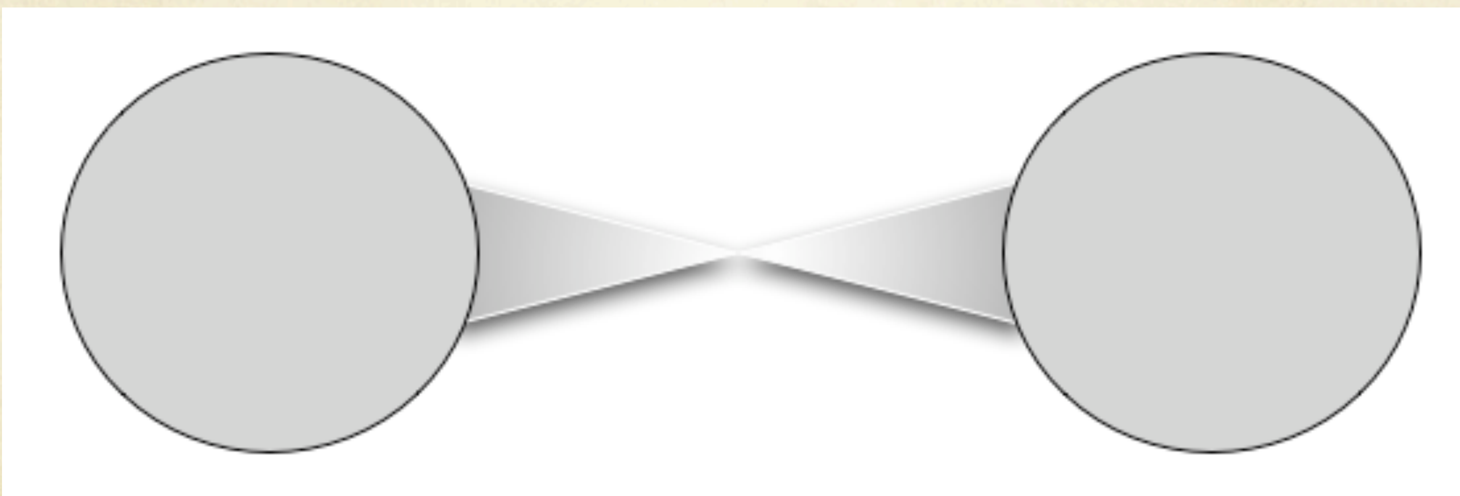
- What's your opinion?
- Should this be the bi-directional analogue?



Back To Visual Encodings

- What's your opinion?

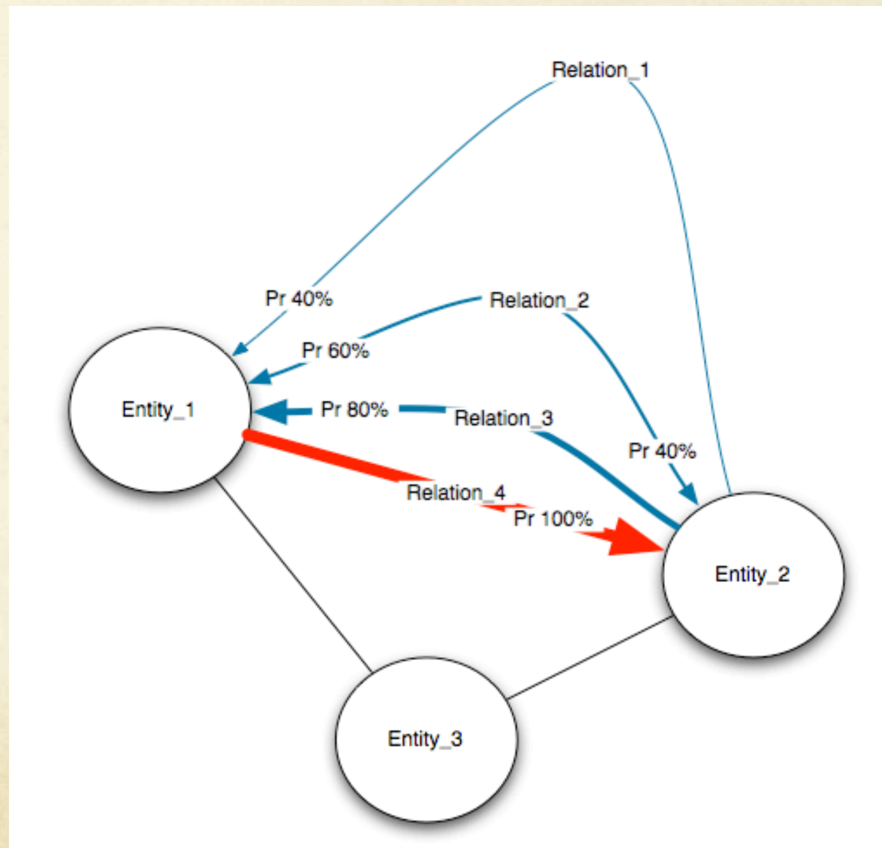
- Should this be the bi-directional analogue?



Back To Visual Encodings

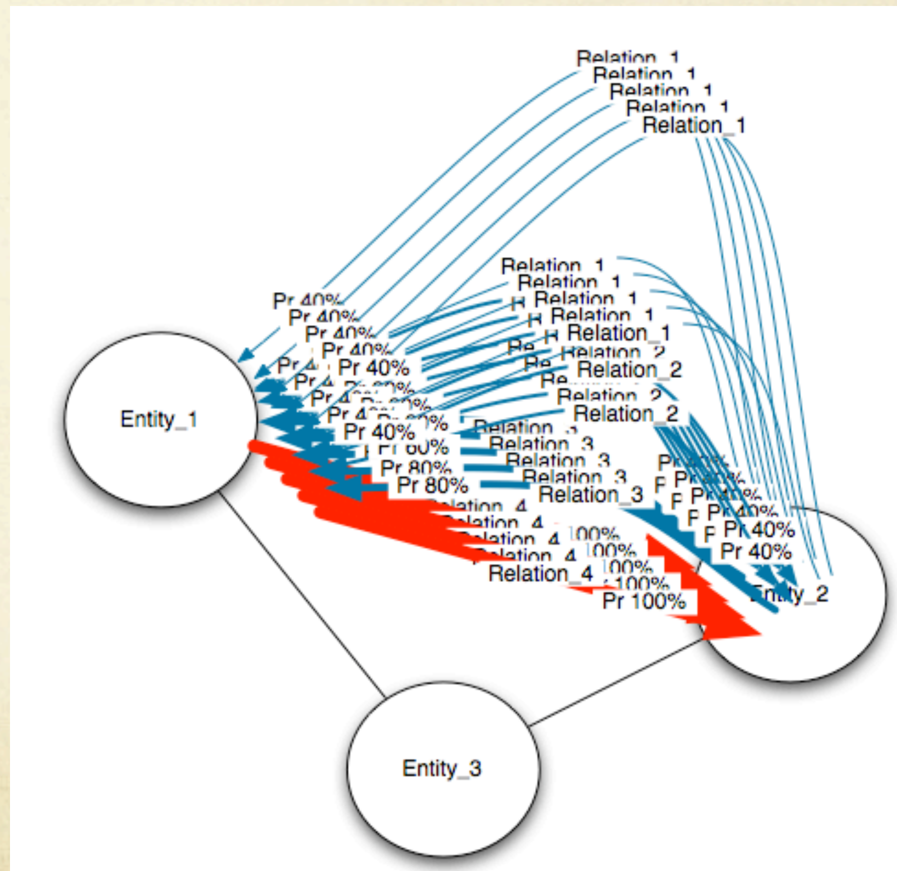
- Another problem, **channel capacity**.

- We're doing okay with a few relations, but...



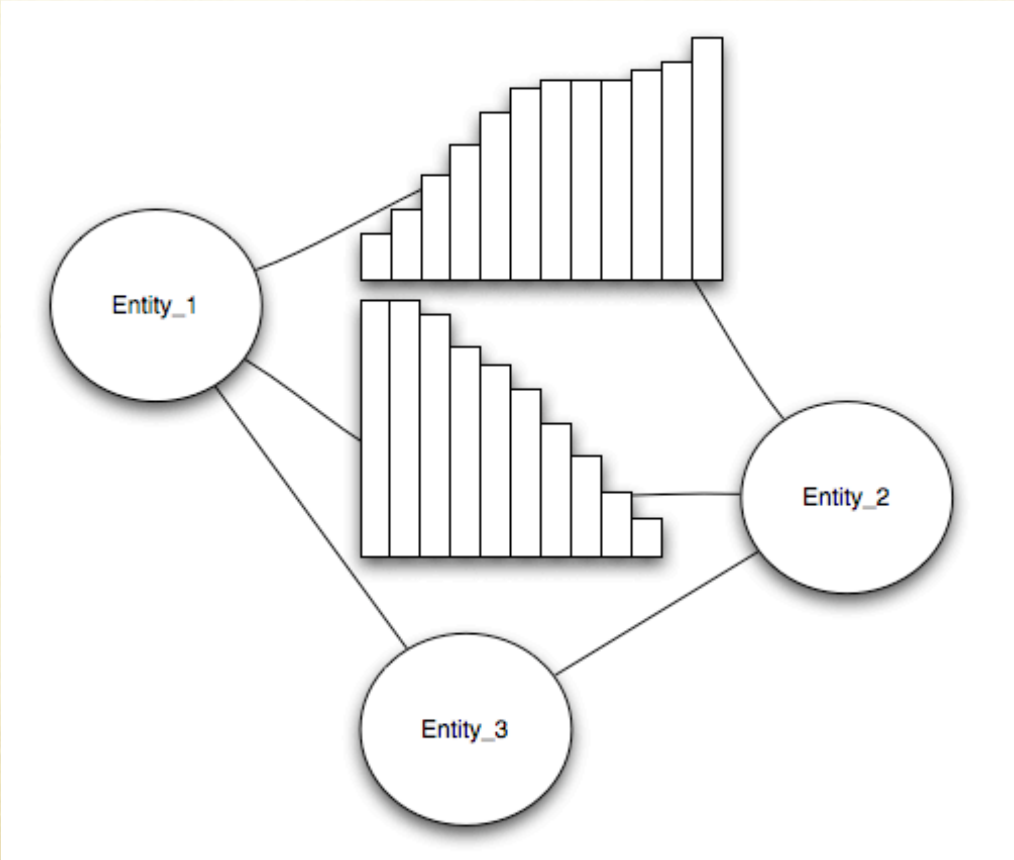
Back To Visual Encodings

- Another problem, **channel capacity**.
- Many?
- Might become illegible!



Back To Visual Encodings

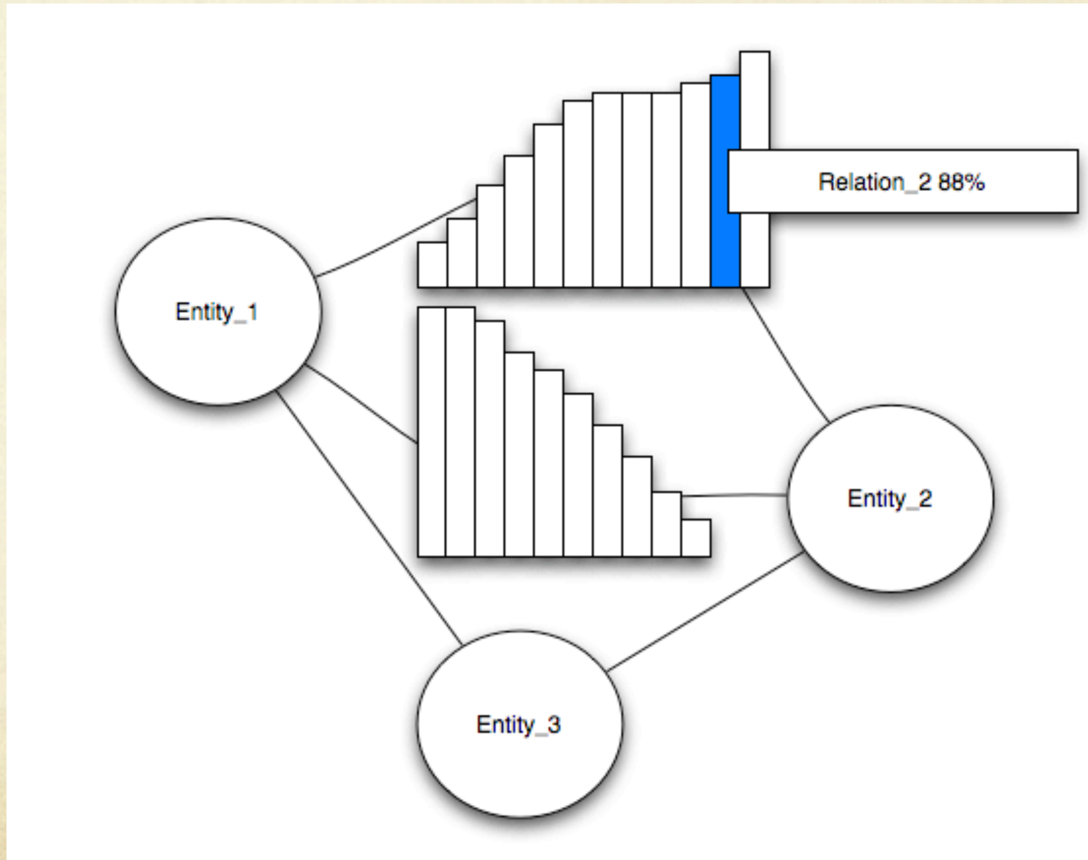
- Another problem, **channel capacity**.
- Possible Solution:
- Bars indicate a **relation,** and bar height is the **level of uncertainty** associated with each relation



Back To Visual Encodings

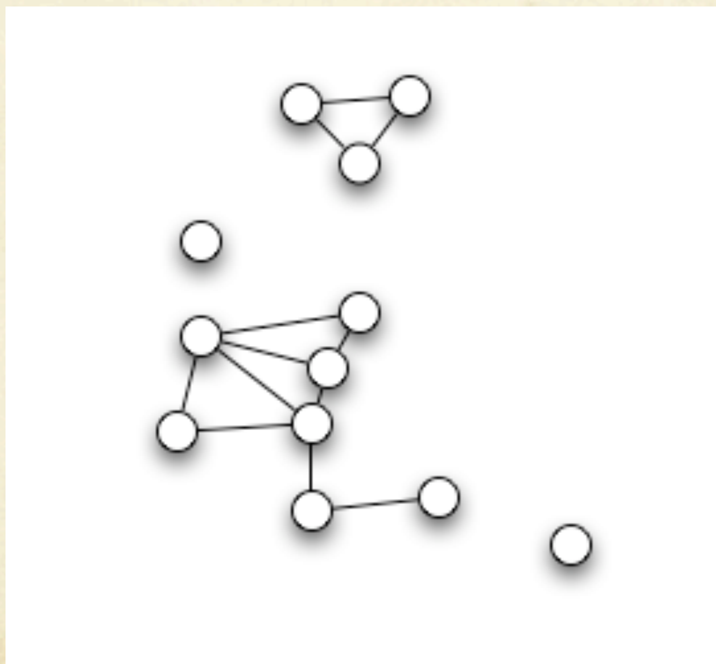
- Another problem, **channel capacity**.

- Hovering the cursor over the bar reveals the relation's identity and the uncertainty associated with it



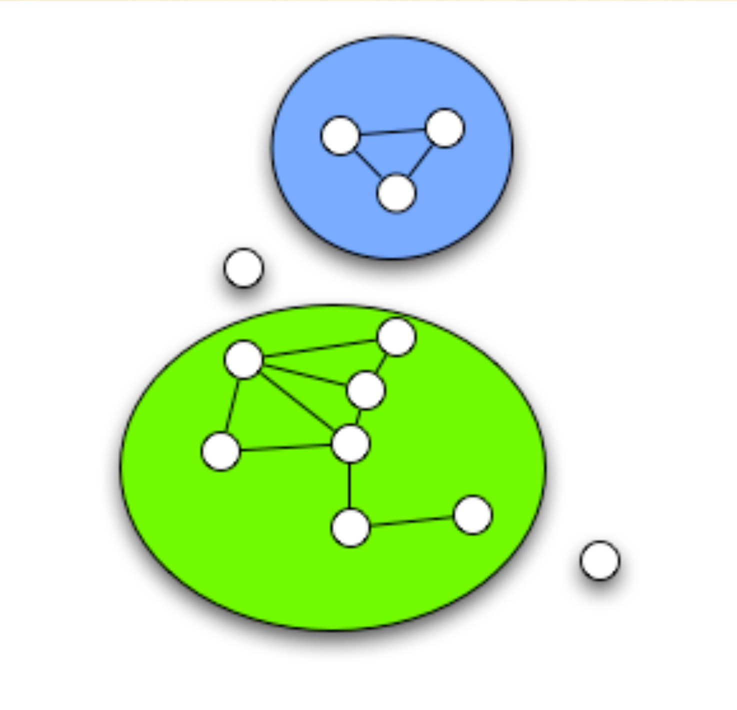
Further Questions

- What kinds of information should be accessible from an overview of the graph?



Further Questions

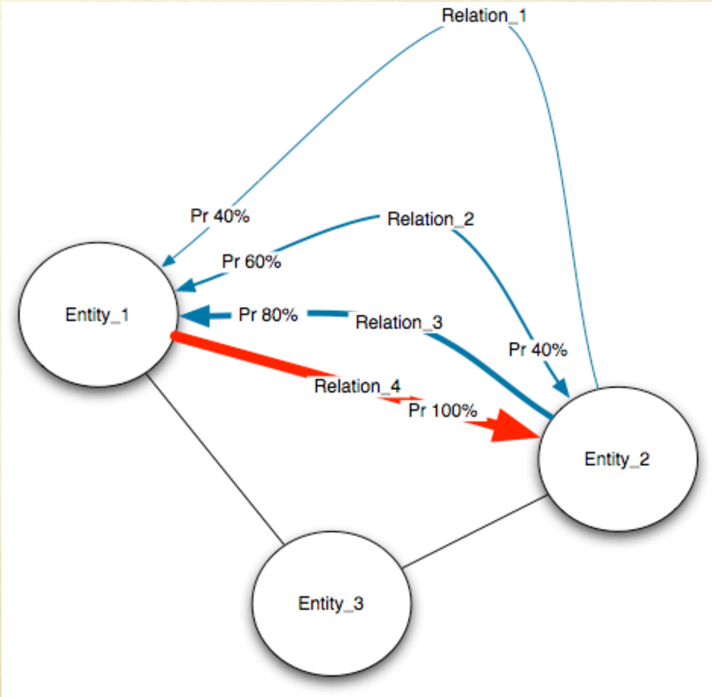
- Group encoding for quick browsing and showing large scale relationships?



Further Questions

- Allow for interactive data labeling so an algorithm designer can mark relations that they “approve” as making sense?

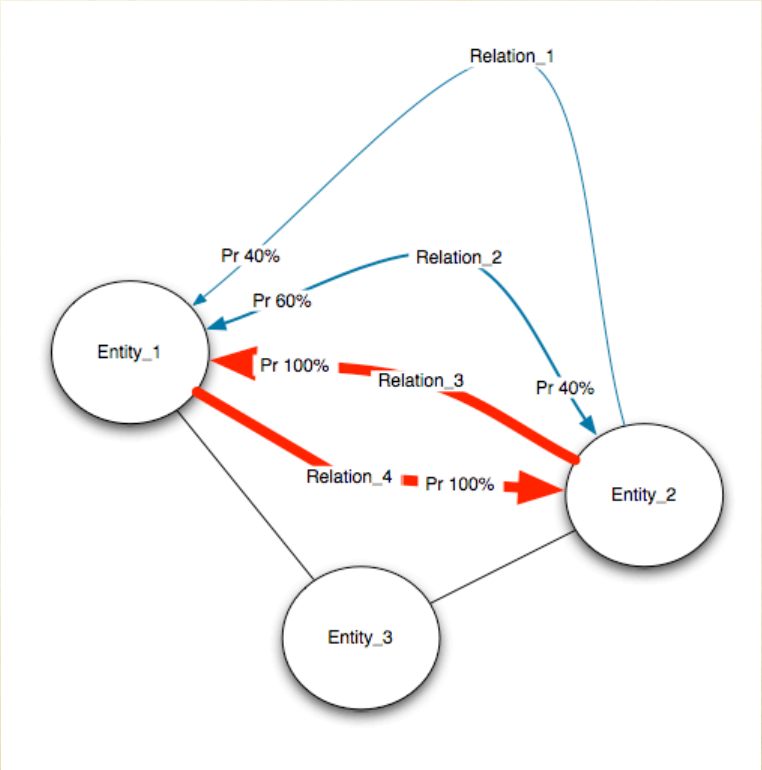
- Here Relation 4 is a ground truth, but suppose predicted Relation 3 is correct, too:



Further Questions

- Allow for interactive data labeling so an algorithm designer can mark relations that they "approve" as making sense?

- Here Relation 4 is a ground truth, but suppose Relation 3 is correct, too:



Thank You

- This completes the status update for the RelaViz Project.