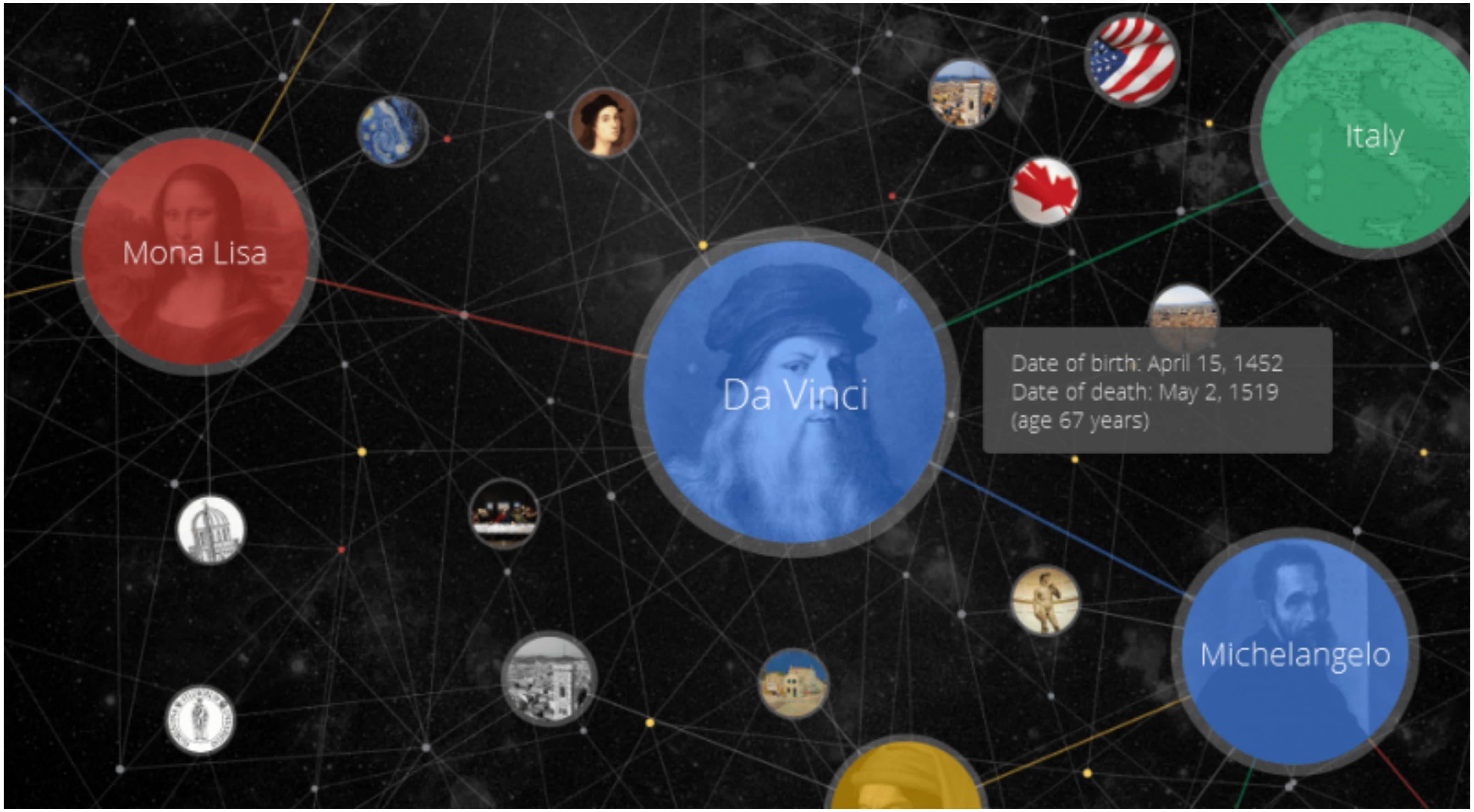
Random Walk Inference Over Knowledge Bases and Text

11-805 class presentation

Matt Gardner

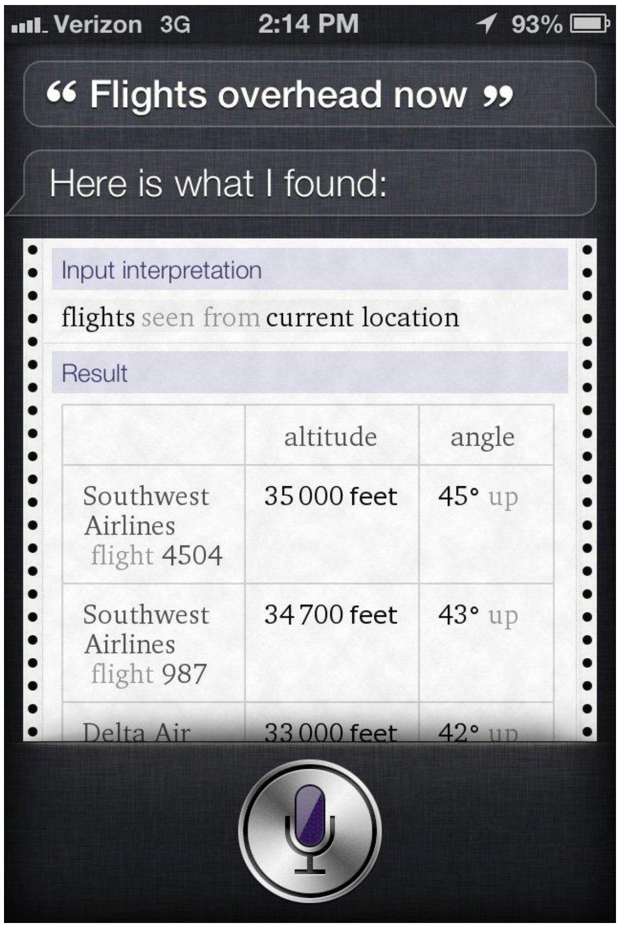
work done by: Ni Lao, Matt Gardner, and collaborators

What is a knowledge base?

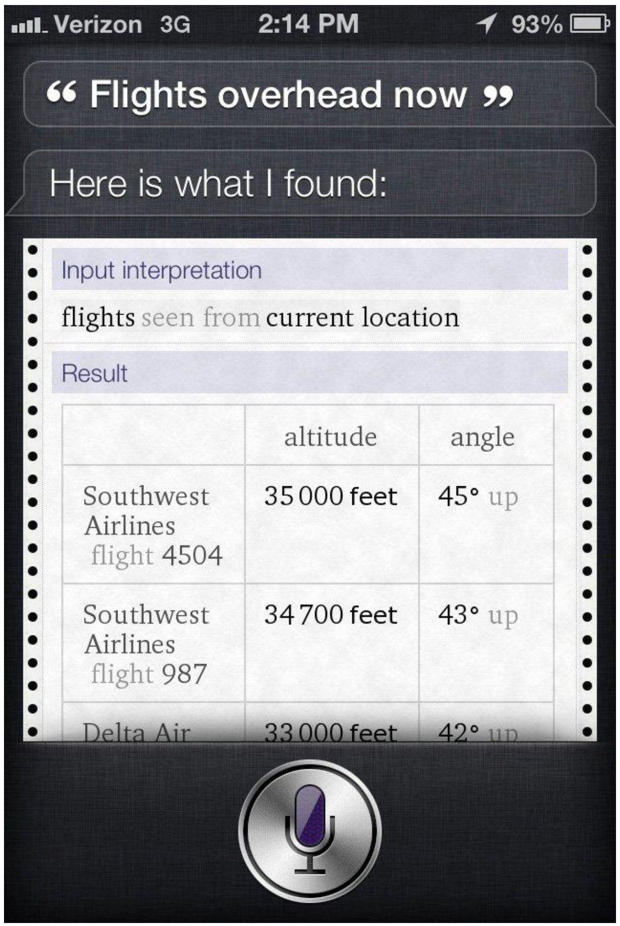


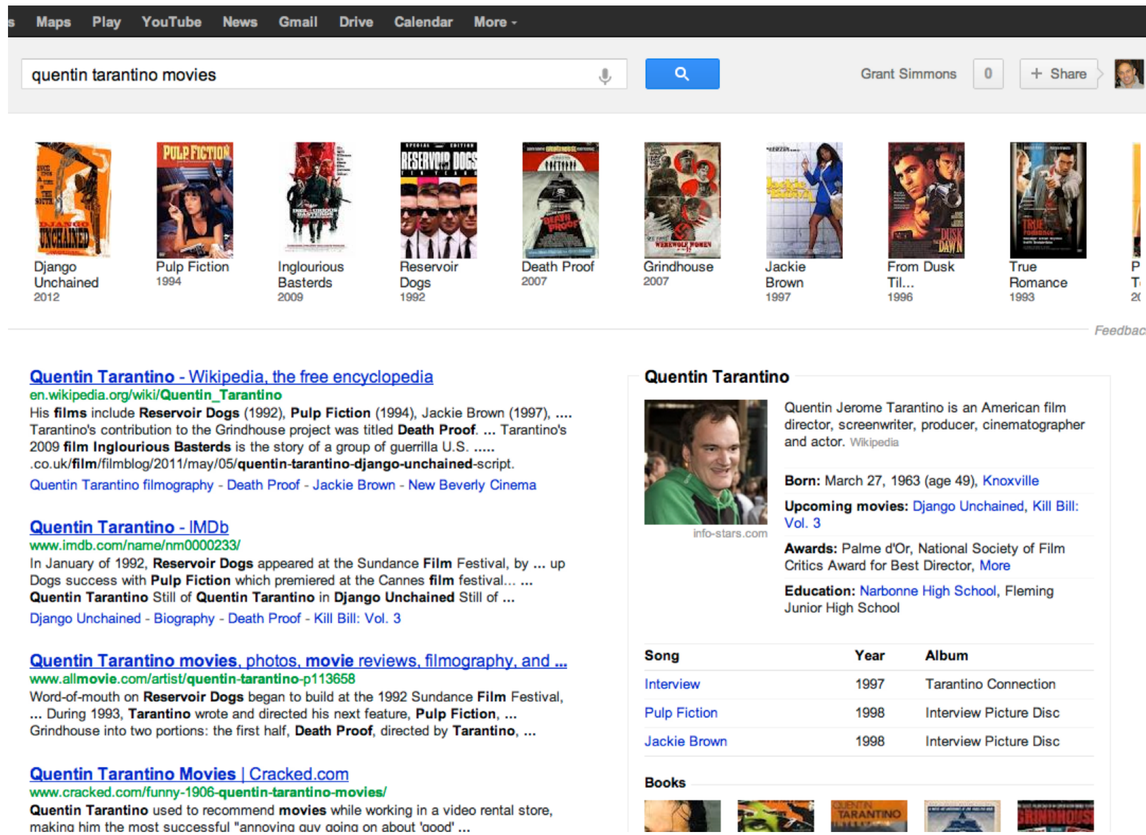
Why knowledge bases?

Why knowledge bases?



Why knowledge bases?





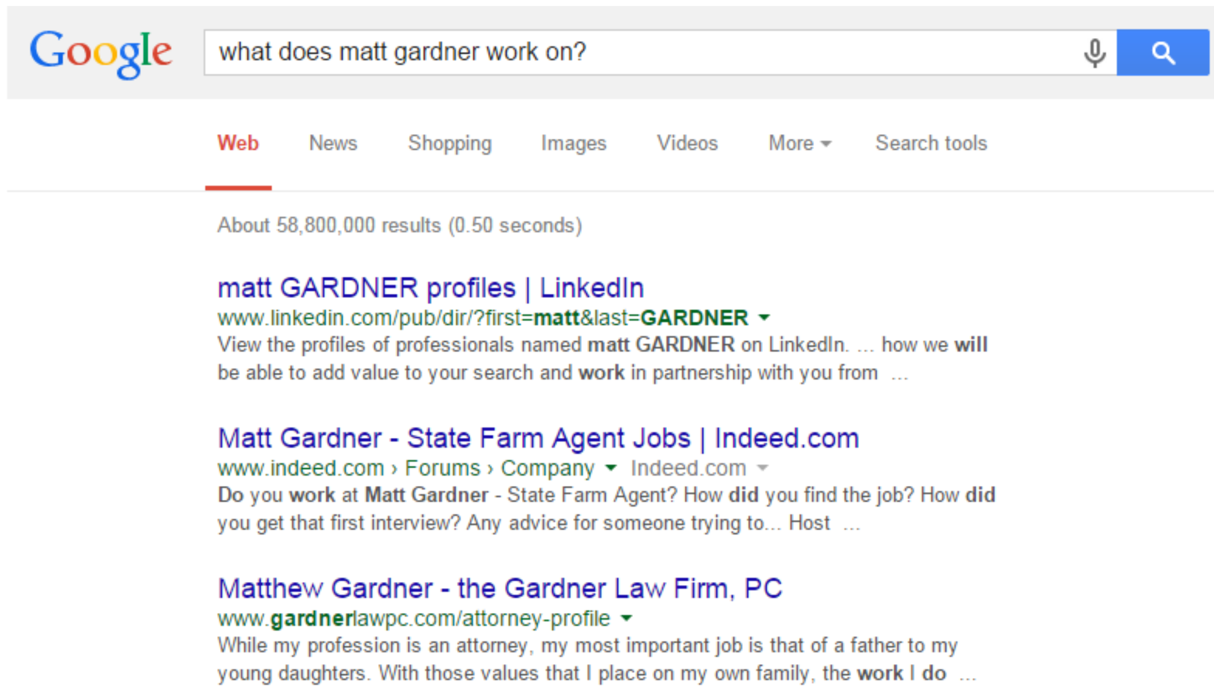
But ...

But ...


4:46
What would you like to do?
Who is the CEO of Samsung
I'm sorry, I don't have the answer. Would you like to search the web?
Search the web

But ...

4:46
What would you like to do?
Who is the CEO of Samsung
I'm sorry, I don't have the answer. Would you like to search the web?
Search the web



Google
what does matt gardner work on?
Web
News
Shopping
Images
Videos
More
Search tools
About 58,800,000 results (0.50 seconds)
matt GARDNER profiles | LinkedIn
www.linkedin.com/pub/dir/?first=matt&last=GARDNER
View the profiles of professionals named matt GARDNER on LinkedIn. ... how we will be able to add value to your search and work in partnership with you from ...
Matt Gardner - State Farm Agent Jobs | Indeed.com
www.indeed.com › Forums › Company
Indeed.com
Do you work at Matt Gardner - State Farm Agent? How did you find the job? How did you get that first interview? Any advice for someone trying to... Host ...
Matthew Gardner - the Gardner Law Firm, PC
www.gardnerlawpc.com/attorney-profile
While my profession is an attorney, my most important job is that of a father to my young daughters. With those values that I place on my own family, the work I do ...

So we do inference

So we do inference

- Predict missing facts given what we know

So we do inference

- Predict missing facts given what we know

- Lots of ways to do this, today we’ll talk about random walk inference, or the Path Ranking Algorithm (PRA)

So we do inference

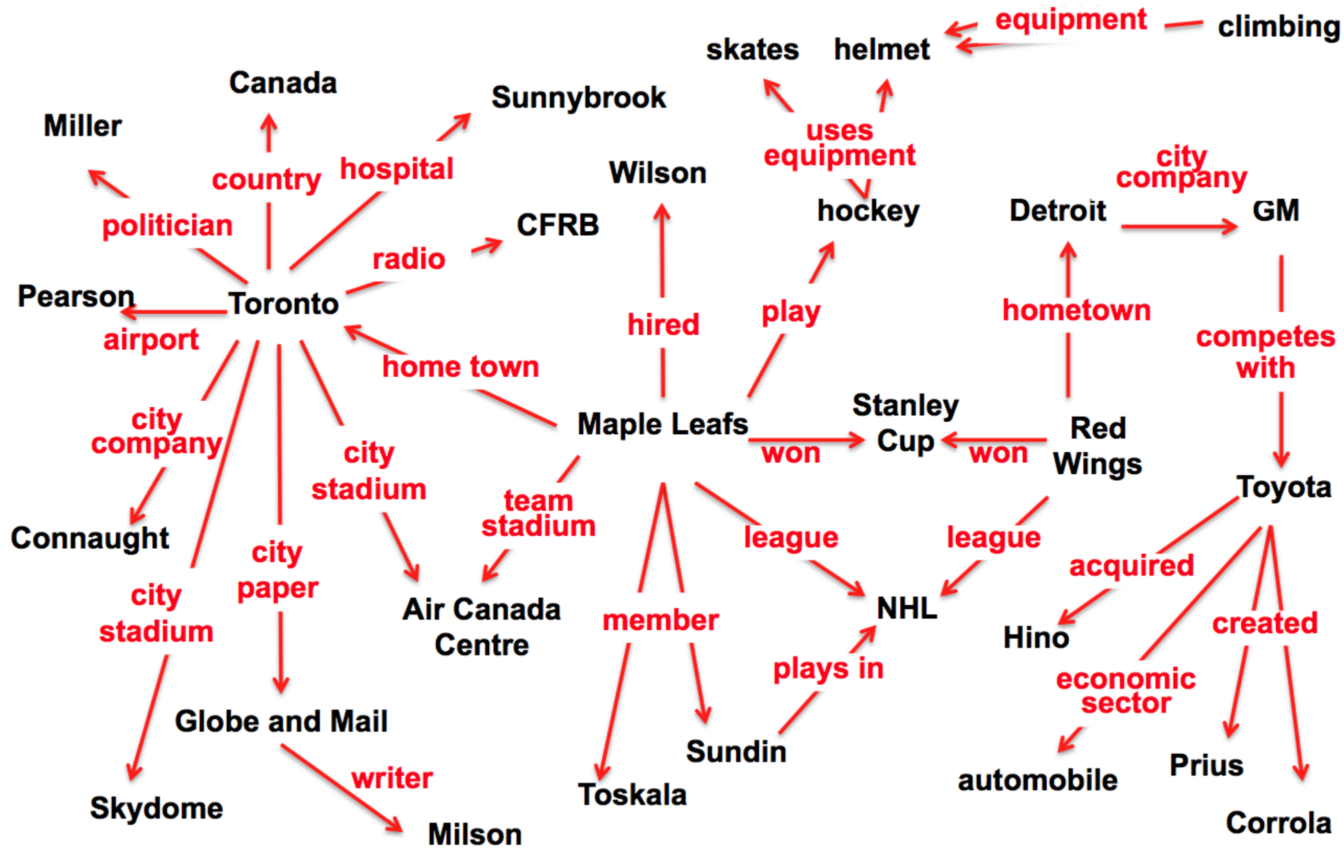
- Predict missing facts given what we know

- Lots of ways to do this, today we'll talk about random walk inference, or the Path Ranking Algorithm (PRA)

- [Lao, Mitchell, Cohen, EMNLP 2011]

Inference

[Lao et al, *EMNLP* 2011]

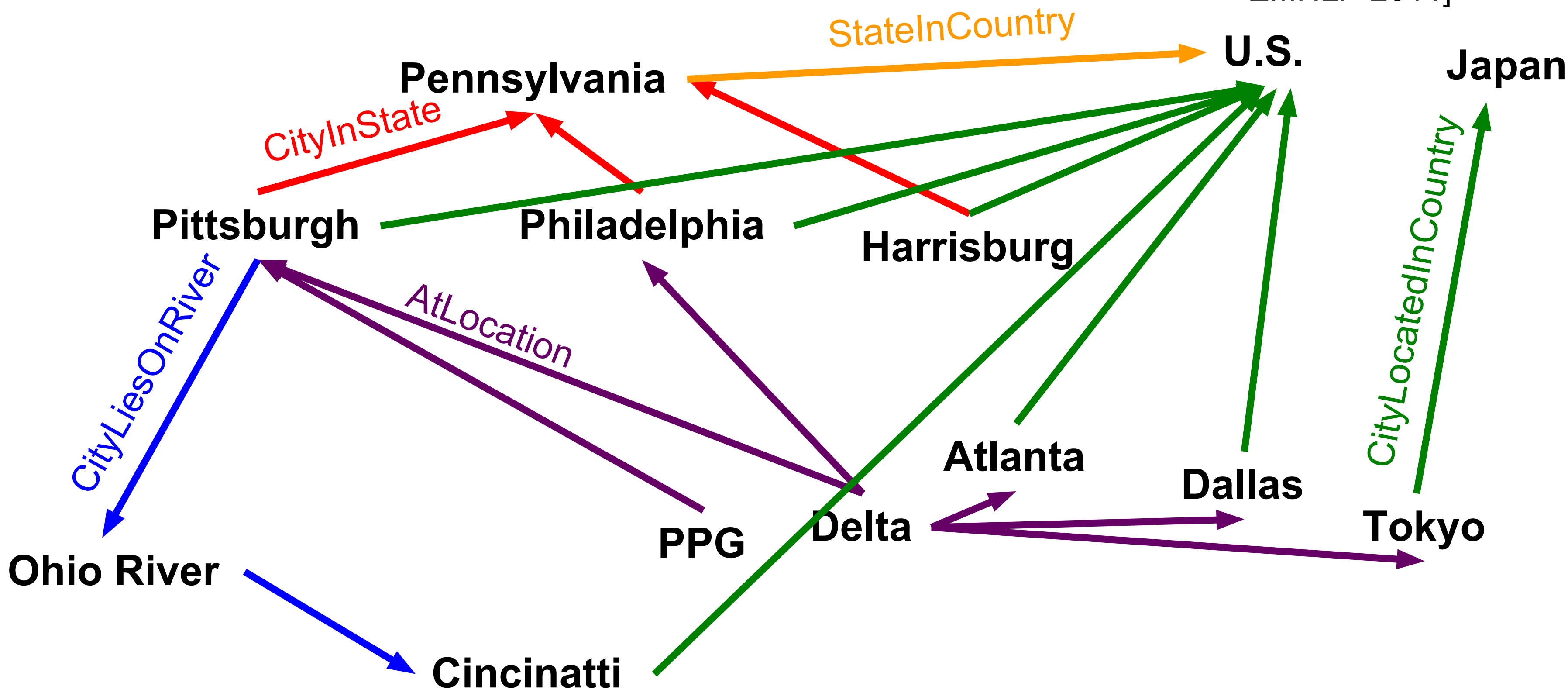


If: x1 — competes with (x1,x2) → x2 — economic sector (x2, x3) → x3

Then: economic sector (x1, x3)

CityLocatedInCountry - Selecting path features

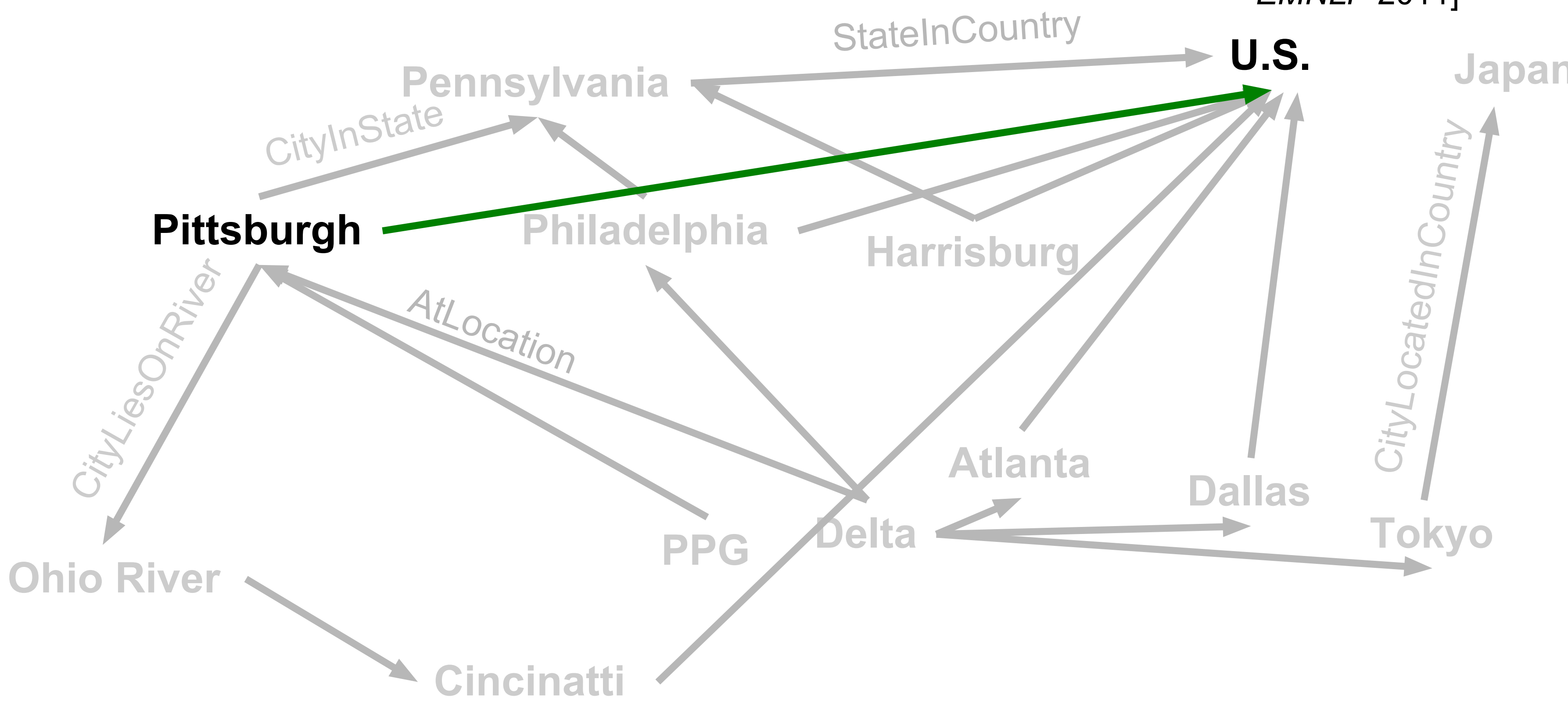
[Lao et al, *EMNLP* 2011]



Path type	**Count**

CityLocatedInCountry - Selecting path features

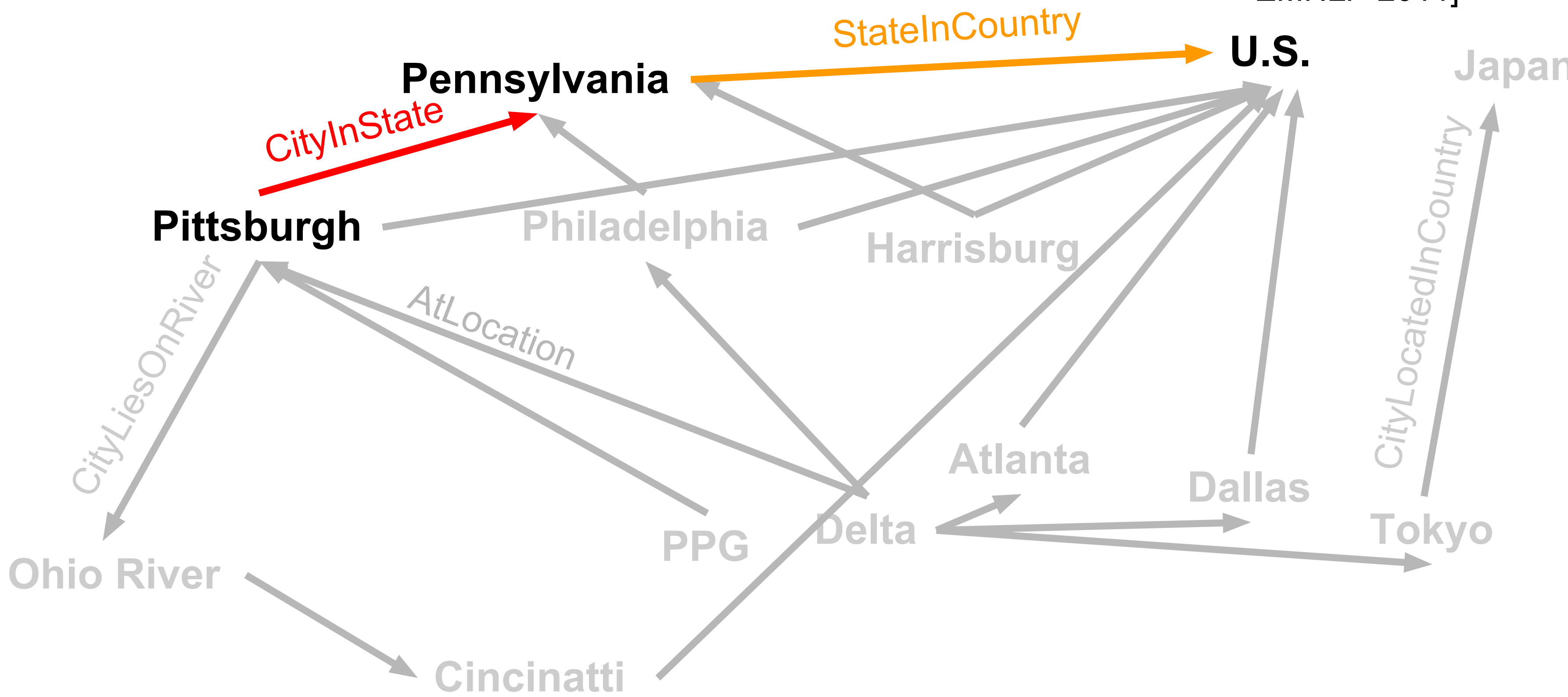
[Lao et al, *EMNLP* 2011]



Path type	**Count**

CityLocatedInCountry - Selecting path features

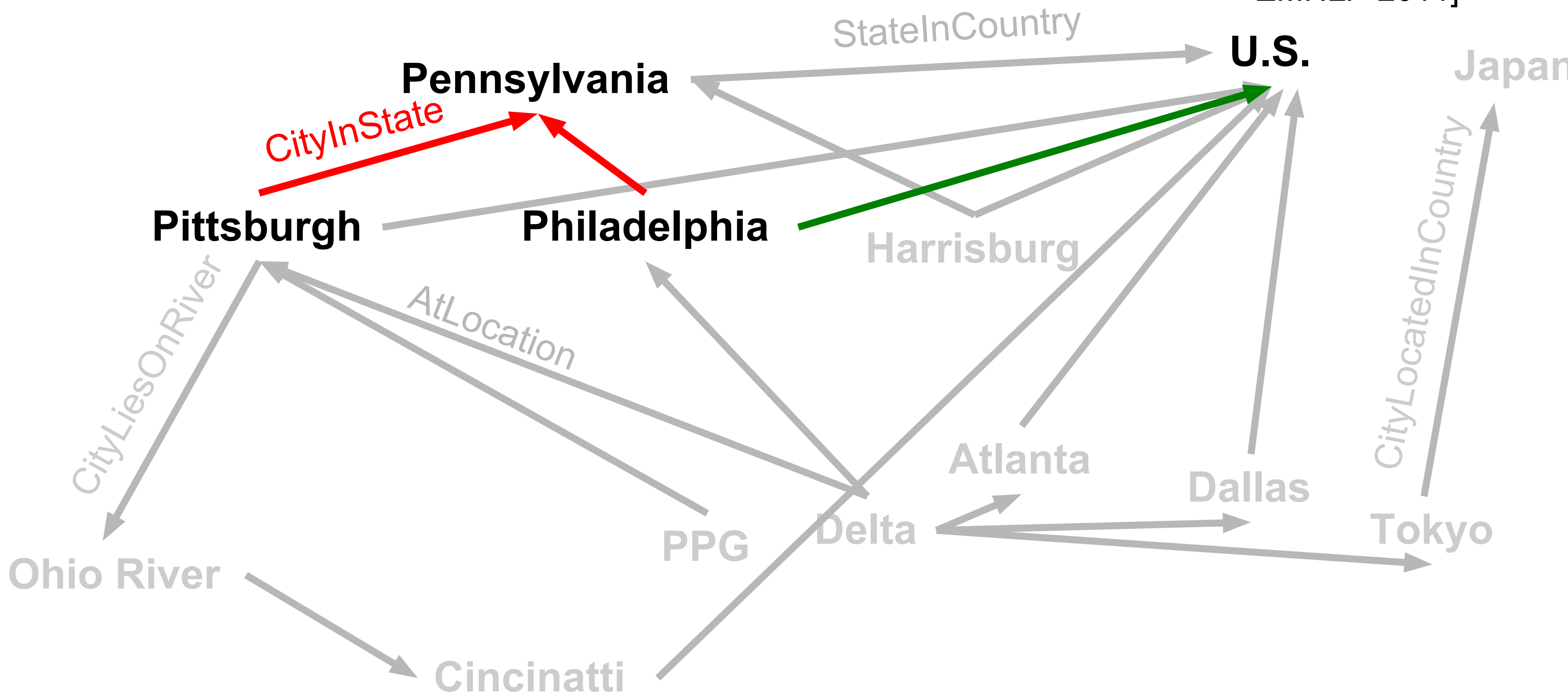
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	1

CityLocatedInCountry - Selecting path features

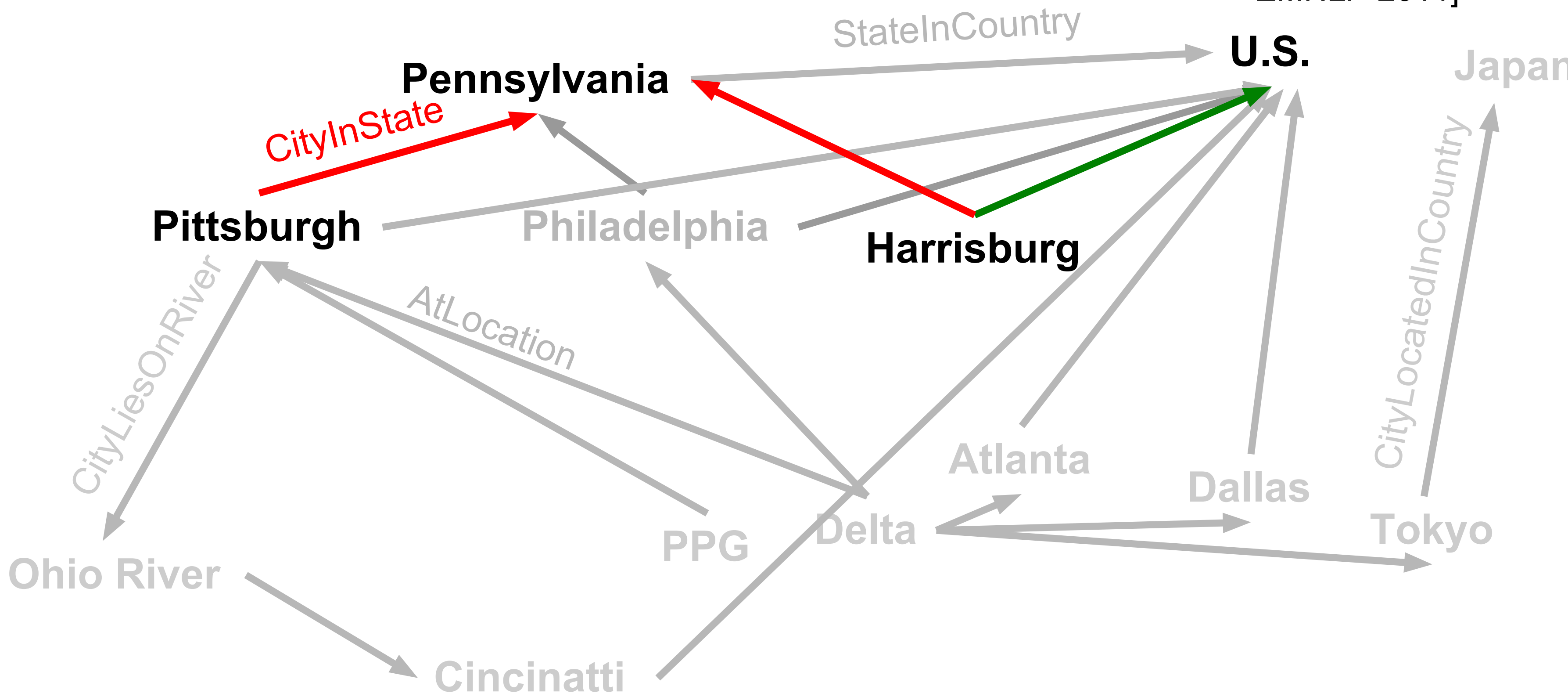
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState^{-1}, CityLocatedInCountry	1

CityLocatedInCountry - Selecting path features

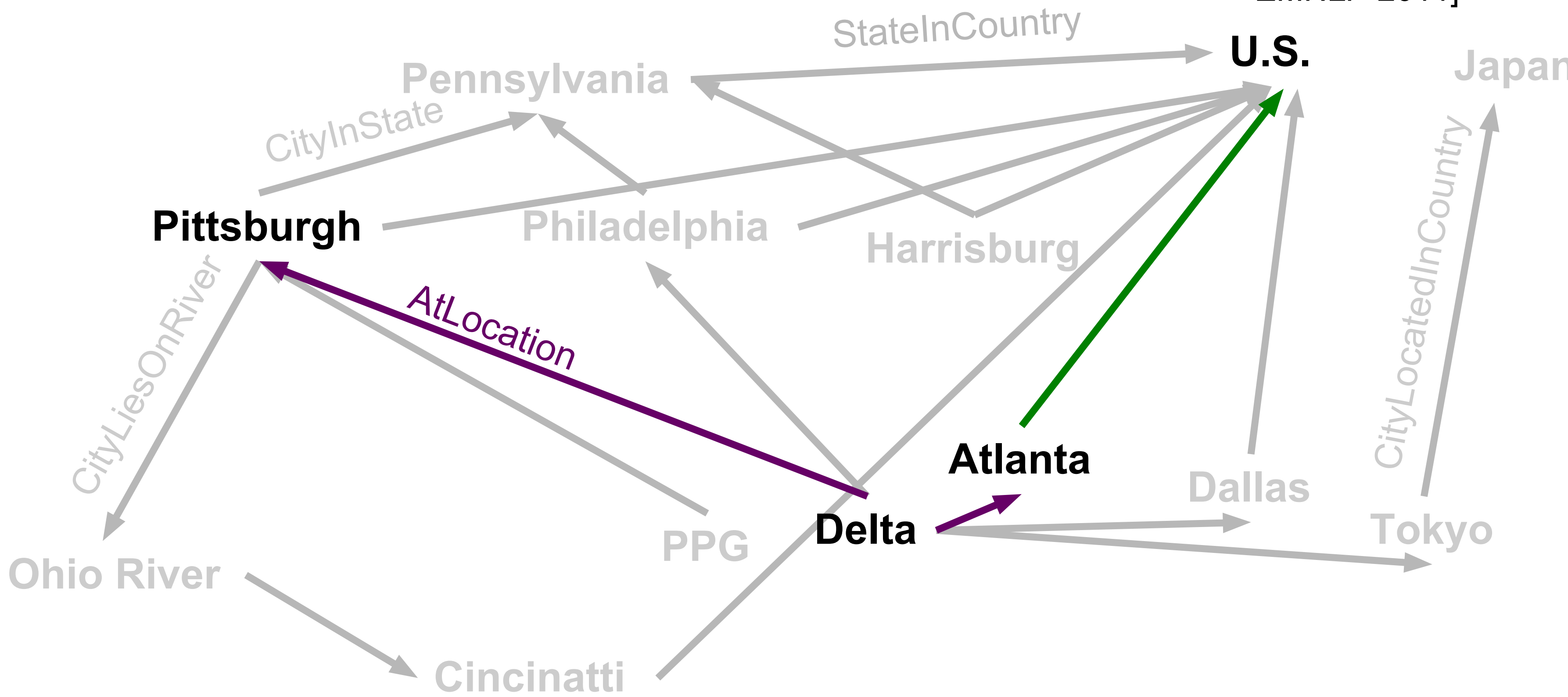
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState^{-1}, CityLocatedInCountry	2

CityLocatedInCountry - Selecting path features

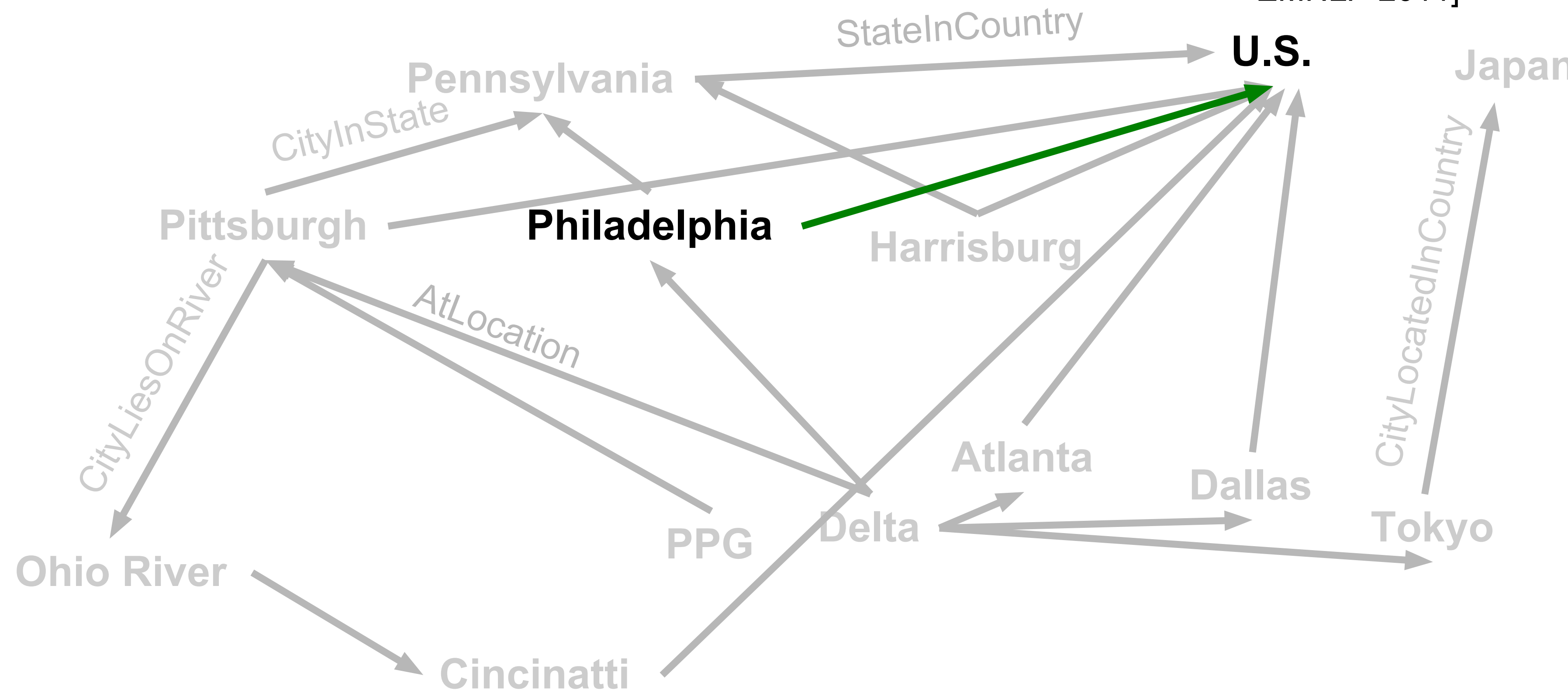
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState^{-1}, CityLocatedInCountry	2
AtLocation^{-1}, AtLocation, CityLocatedInCountry	1

CityLocatedInCountry - Selecting path features

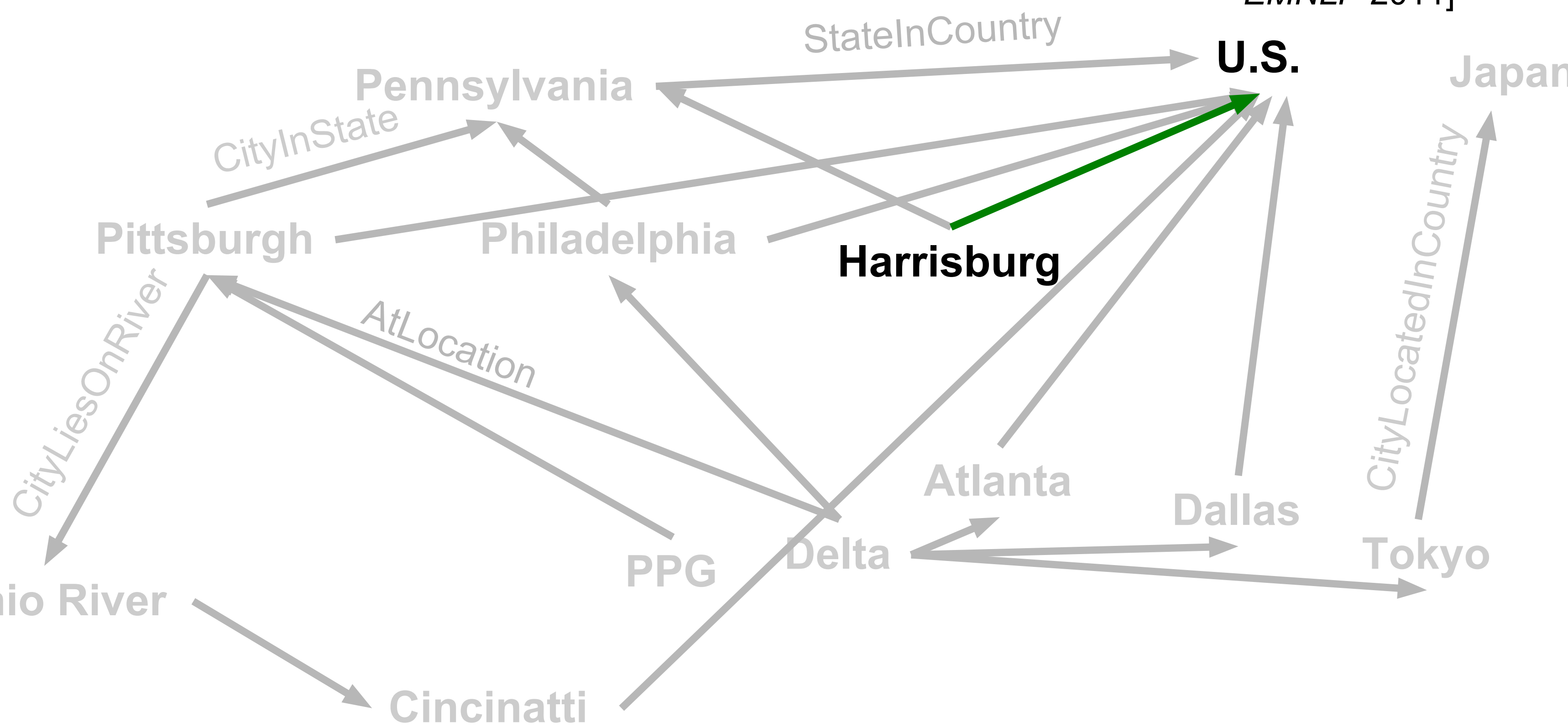
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	2
CityInState, CityInState^{-1}, CityLocatedInCountry	24
AtLocation^{-1}, AtLocation, CityLocatedInCountry	10
CityLiesOnRiver, CityLiesOnRiver^{-1}, CityLocatedInCountry	1
…	…

CityLocatedInCountry - Selecting path features

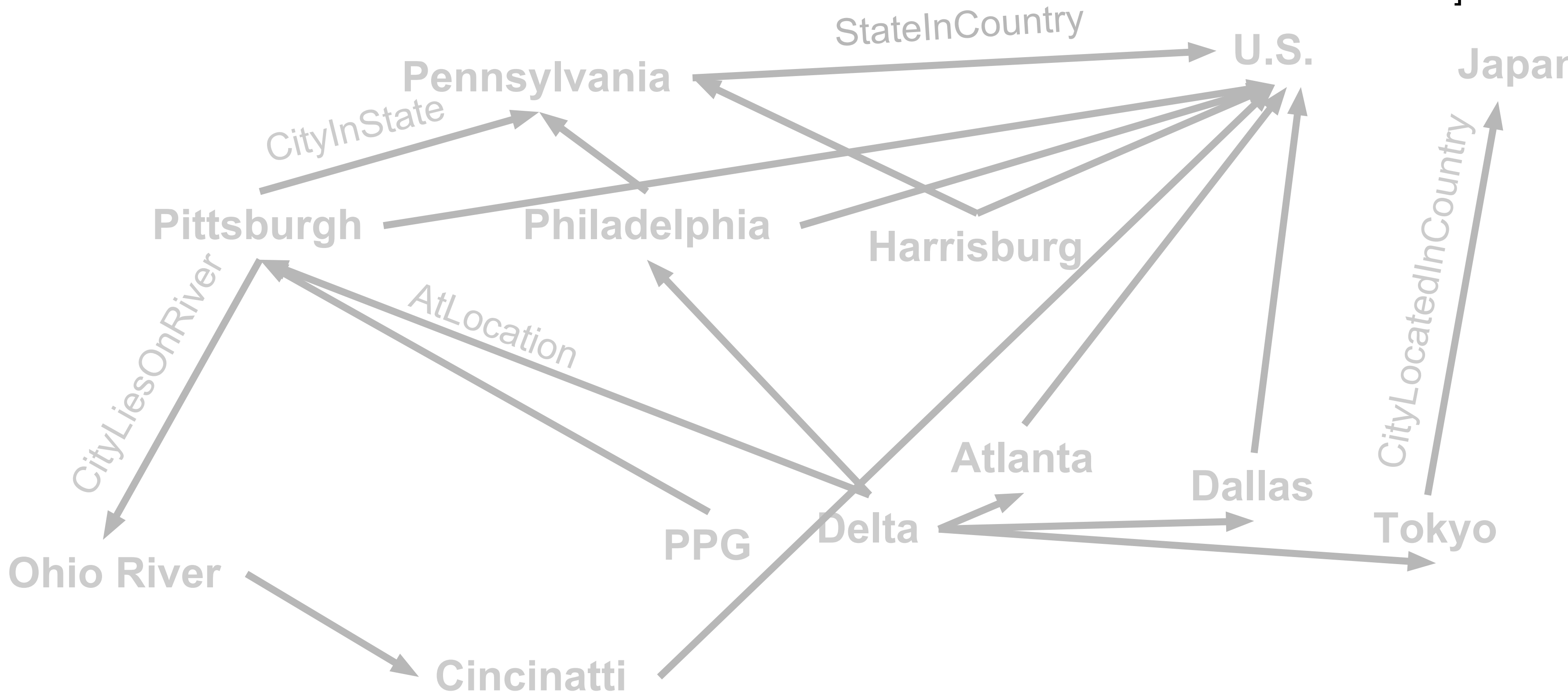
[Lao et al, *EMNLP* 2011]



Path type	Count
CityInState, StateInCountry	3,892
CityInState, CityInState^{-1}, CityLocatedInCountry	234
AtLocation^{-1}, AtLocation, CityLocatedInCountry	1,543
CityLiesOnRiver, $\text{CityLiesOnRiver}^{-1}$, CityLocatedInCountry	123
…	…

CityLocatedInCountry - Selecting path features

[Lao et al, *EMNLP* 2011]



Path type	**Count**
CityInState, StateInCountry	8,172
CityInState, CityInState^{-1}, CityLocatedInCountry	2,234
AtLocation^{-1}, AtLocation, CityLocatedInCountry	5,273
CityLiesOnRiver, $\text{CityLiesOnRiver}^{-1}$, CityLocatedInCountry	298
…	…

Select the most frequent path types, and keep them as features in the model

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, *EMNLP* 2011]

Pittsburgh

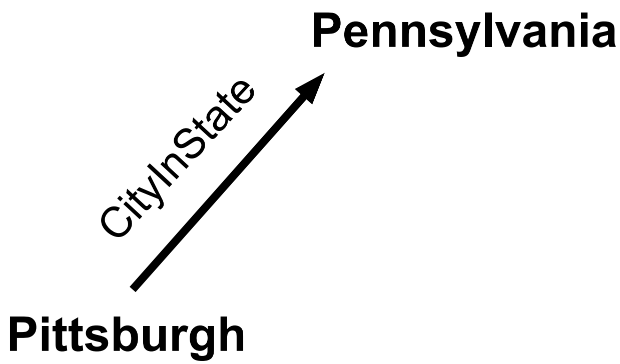
<u>Feature = Typed Path</u>

CityInState, CityInState^{-1}, CityLocatedInCountry

<u>Feature Value</u>

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, *EMNLP* 2011]



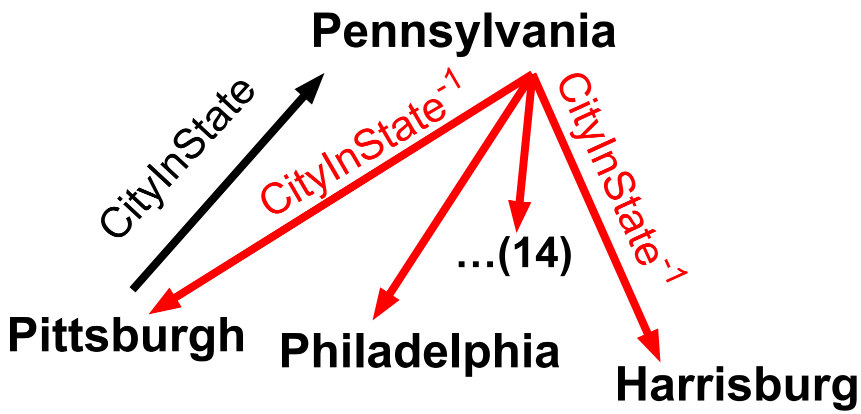
Feature = Typed Path

CityInState, CityInState^{-1}, CityLocatedInCountry

Feature Value

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

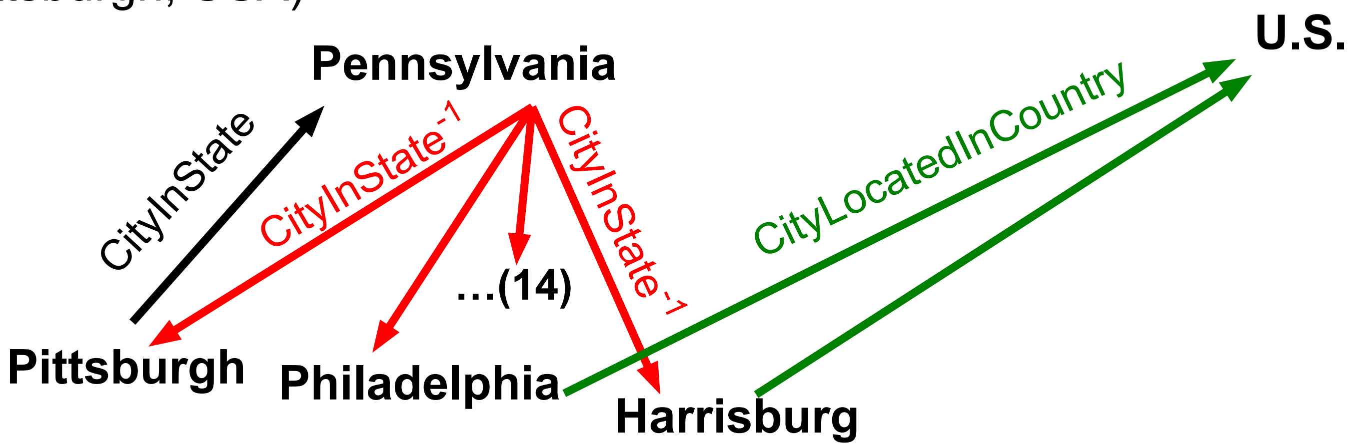
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, *EMNLP* 2011]



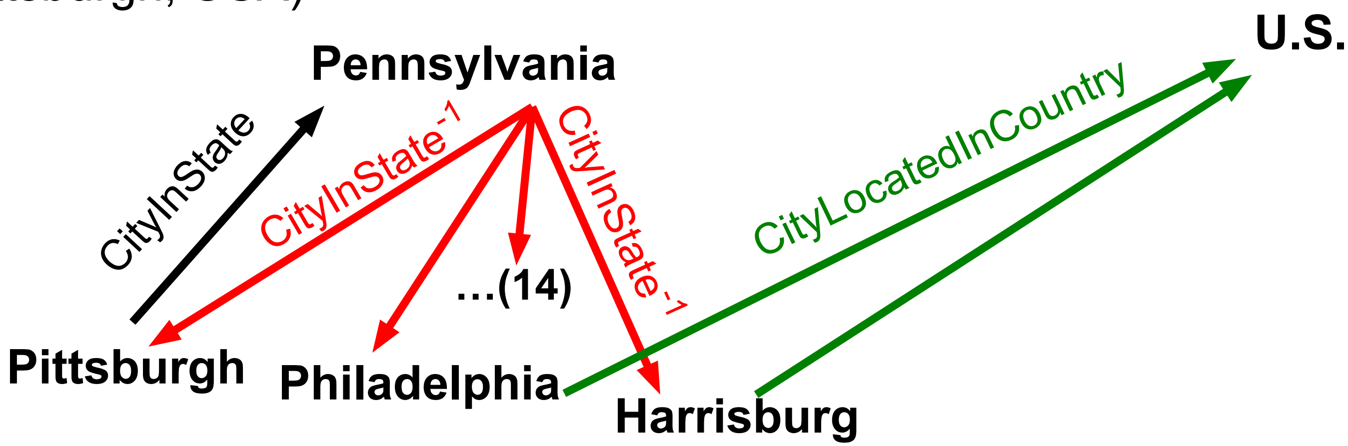
Feature = Typed Path

CityInState, CityInState^{-1}, CityLocatedInCountry

Feature Value

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, *EMNLP* 2011]

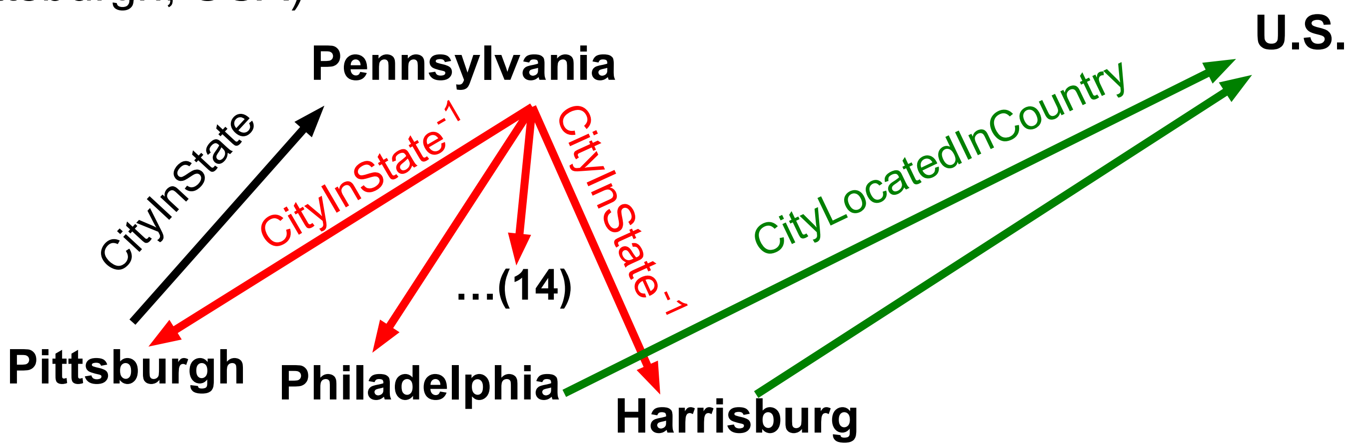


Pr(U.S. | Pittsburgh, TypedPath)

Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

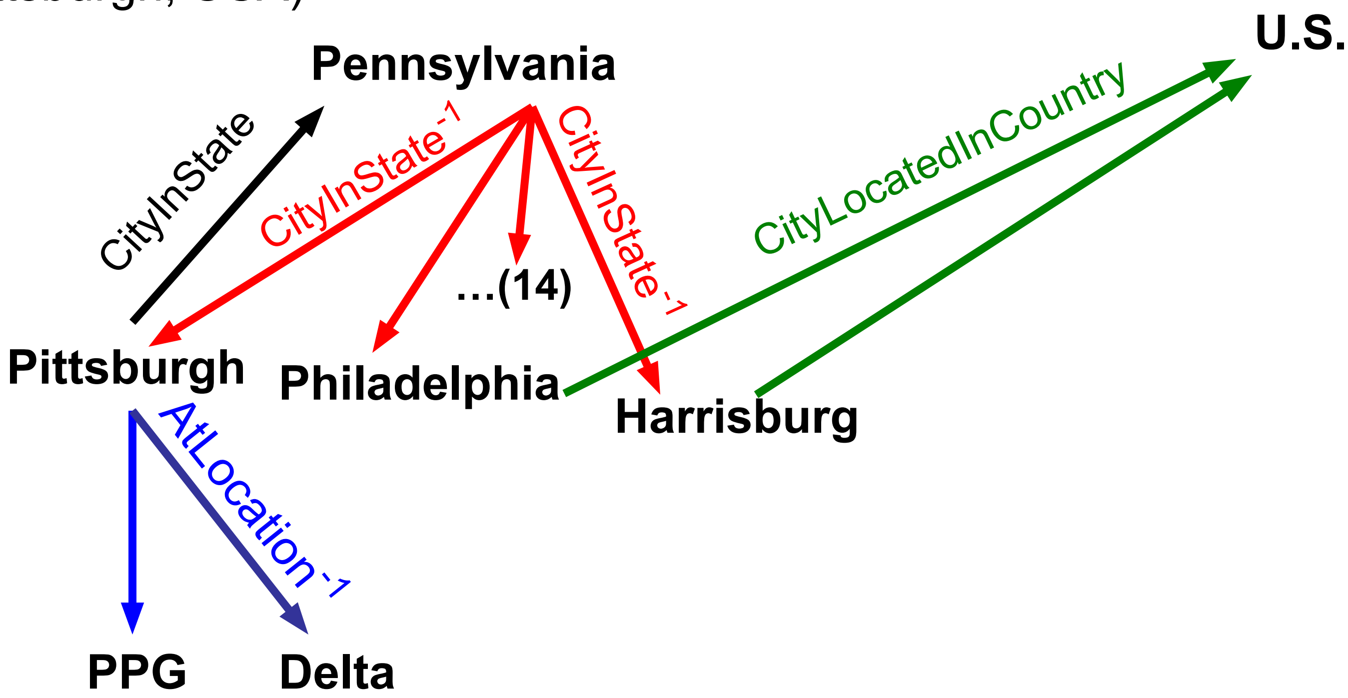
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0
AtLocation^{-1}, AtLocation, CityLocatedInCountry	

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

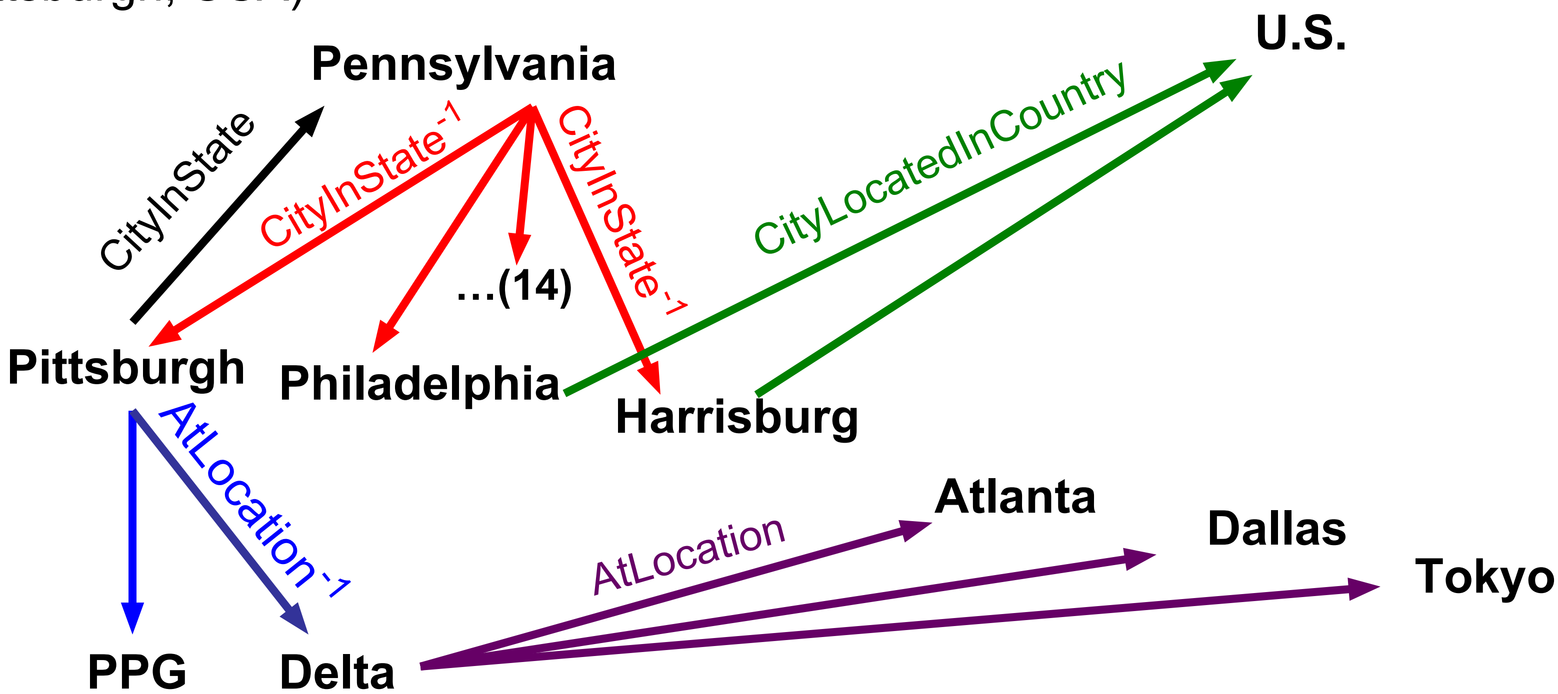
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0
AtLocation^{-1}, AtLocation, CityLocatedInCountry	

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

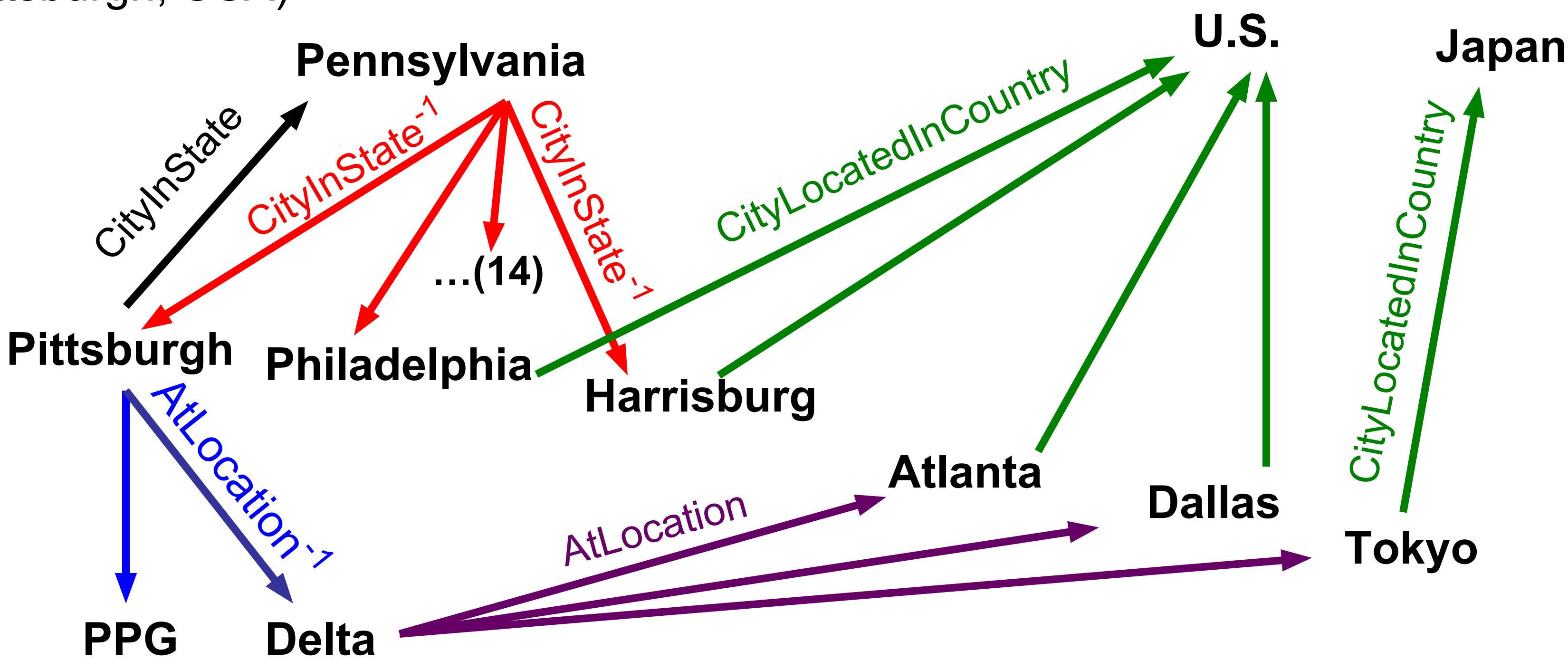
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0
AtLocation^{-1}, AtLocation, CityLocatedInCountry	

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

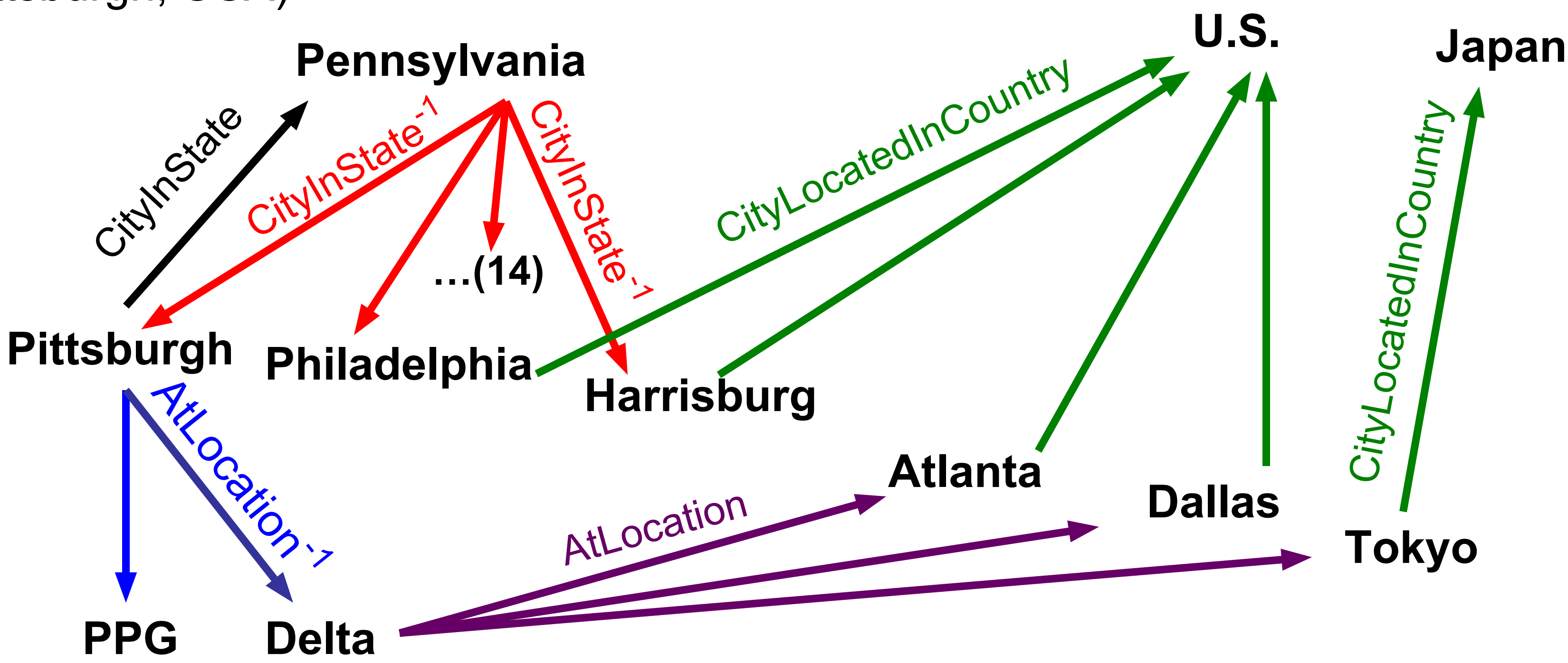
[Lao et al, *EMNLP* 2011]



Feature = Typed Path	Feature Value
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0
AtLocation^{-1}, AtLocation, CityLocatedInCountry	0.6

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, *EMNLP* 2011]



Feature = Typed Path	**Feature Value**
CityInState, CityInState^{-1}, CityLocatedInCountry	1.0
AtLocation^{-1}, AtLocation, CityLocatedInCountry	0.6
…	…

This is a row in a feature matrix! Use standard logistic regression

PRA Feature Matrix

CityLocatedInCountry	CityInState, CityInState^{-1}, CityLocatedInCountry	AtLocation^{-1}, AtLocation, CityLocatedInCountry	…	…	
(Pittsburgh, USA)	1.0	0.6	0.0	0.4	…
(Pittsburgh, Japan)	0.0	0.2	0.2	0.1	…
…	…				
(Tokyo, USA)	…				
(Tokyo, Japan)	…				
…	…				

PRA Feature Matrix

CityLocatedInCountry	CityInState, CityInState^{-1}, CityLocatedInCountry	AtLocation^{-1}, AtLocation, CityLocatedInCountry	…	…	
(Pittsburgh, USA)	1.0	0.6	0.0	0.4	…
(Pittsburgh, Japan)	0.0	0.2	0.2	0.1	…
…	…				
(Tokyo, USA)	…				
(Tokyo, Japan)	…				
…	…				

Large data sets?

PRA Feature Matrix

CityLocatedInCountry	CityInState, $CityInState^{-1}$, CityLocatedInCountry	$AtLocation^{-1}$, AtLocation, CityLocatedInCountry	…	…	
(Pittsburgh, USA)	1.0	0.6	0.0	0.4	…
(Pittsburgh, Japan)	0.0	0.2	0.2	0.1	…
…	…				
(Tokyo, USA)	…				
(Tokyo, Japan)	…				
…	…				

$$O(n^2)$$

Large data sets?

PRA Feature Matrix

CityLocatedInCountry

	CityInState, CityInState^{-1}, CityLocatedInCountry	AtLocation^{-1}, AtLocation, CityLocatedInCountry	…	…	
(Pittsburgh, USA)	1.0	0.6	0.0	0.4	…
(Pittsburgh, Japan)	0.0	0.2	0.2	0.1	…
…	…				
(Tokyo, USA)	…				
(Tokyo, Japan)	…				
…	…				

$$O(r^l)$$

$$O(n^2)$$

Large data sets?

PRA Feature Matrix

CityLocatedInCountry

	CityInState, CityInState^{-1}, CityLocatedInCountry	AtLocation^{-1}, AtLocation, CityLocatedInCountry	…	…	
(Pittsburgh, USA)	1.0	0.6	0.0	0.4	…
(Pittsburgh, Japan)	0.0	0.2	0.2	0.1	…
…	…				
(Tokyo, USA)	…				
(Tokyo, Japan)	…				
…	…				

$$O(r^l)$$

$$O(n^2)$$

Large data sets?

$$\propto \left(\frac{e}{n}\right)^l$$

Implementation

Implementation

- Small graph: in memory

Implementation

- Small graph: in memory
- Larger graph: GraphChi

Implementation

- Small graph: in memory
- Larger graph: GraphChi
  - Vertex-centric computation. Go through the graph sequentially, processing each walk at each vertex and sending it to the next stop.

Implementation

- Small graph: in memory
- Larger graph: GraphChi
  - Vertex-centric computation. Go through the graph sequentially, processing each walk at each vertex and sending it to the next stop.
- On a cluster: have a graph server

So...

So...

- KB inference is all well and good, but…

So...

- KB inference is all well and good, but…

- What about text?

Inference over KB plus text

Inference over KB plus text

- Augment NELL or Freebase with information automatically extracted from text

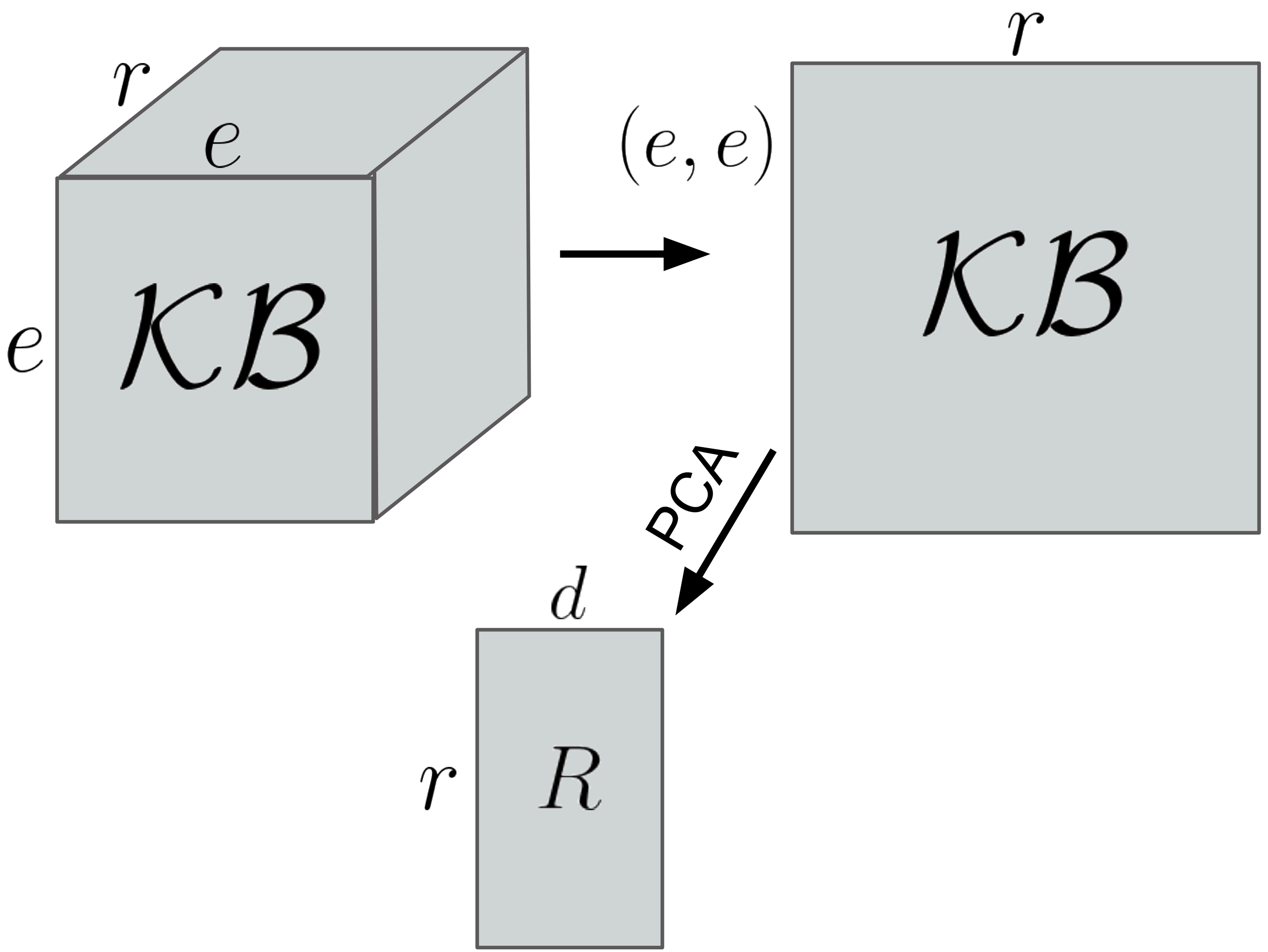
Inference over KB plus text

- Augment NELL or Freebase with information automatically extracted from text
- From the combined graph, predict new NELL (or Freebase) relations

Inference over KB plus text

- Augment NELL or Freebase with information automatically extracted from text
- From the combined graph, predict new NELL (or Freebase) relations
- Basically “aggregate” relation extraction

Simple relation embeddings



“Steel City overlooks the Allegheny”
“Pittsburgh lies on the Mon”
“Pittsburgh sits on the Monongahela”

"Steel City overlooks the Allegheny"
"Pittsburgh lies on the Mon"
"Pittsburgh sits on the Monongahela"

"Steel City"

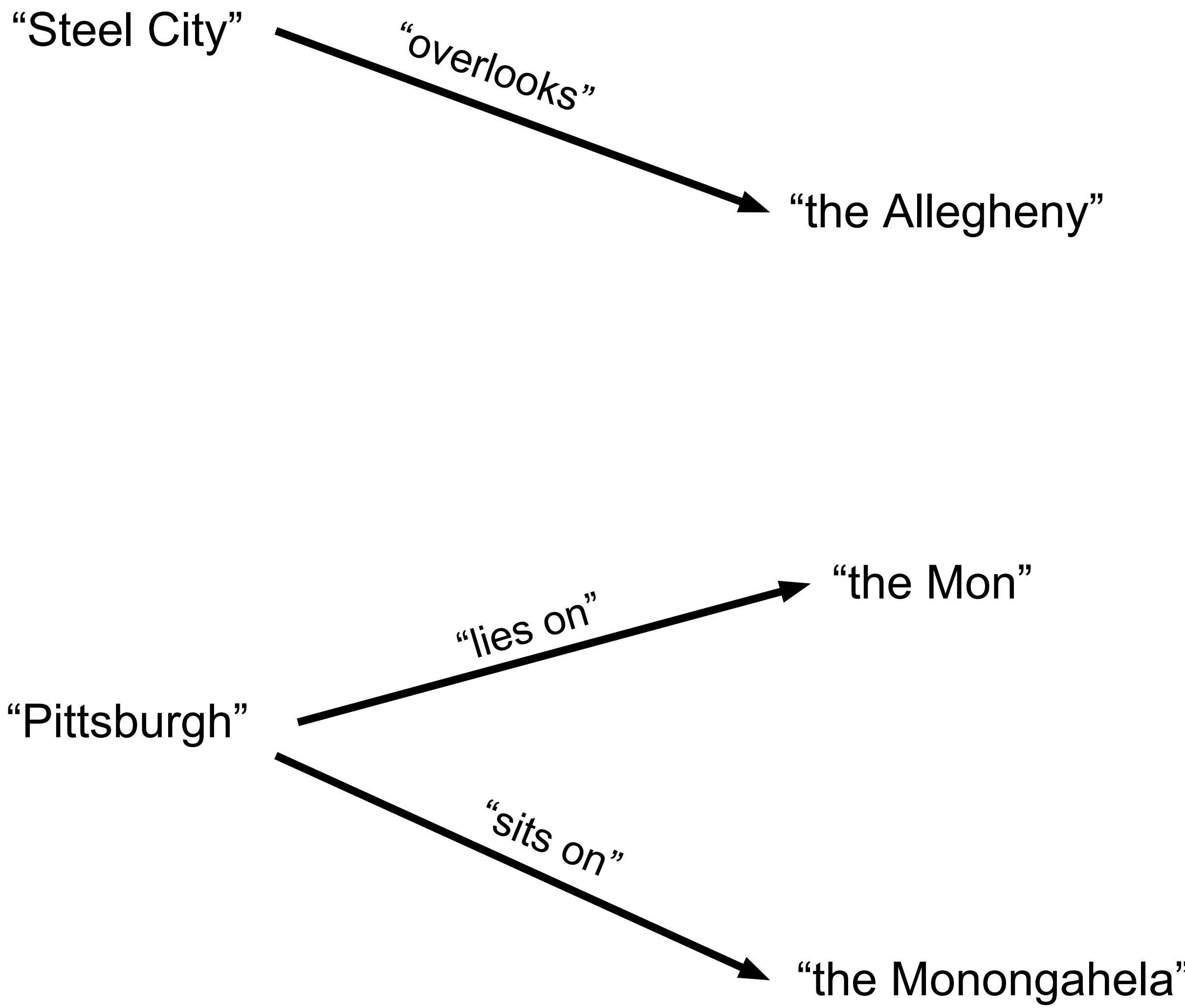
"the Allegheny"

"the Mon"

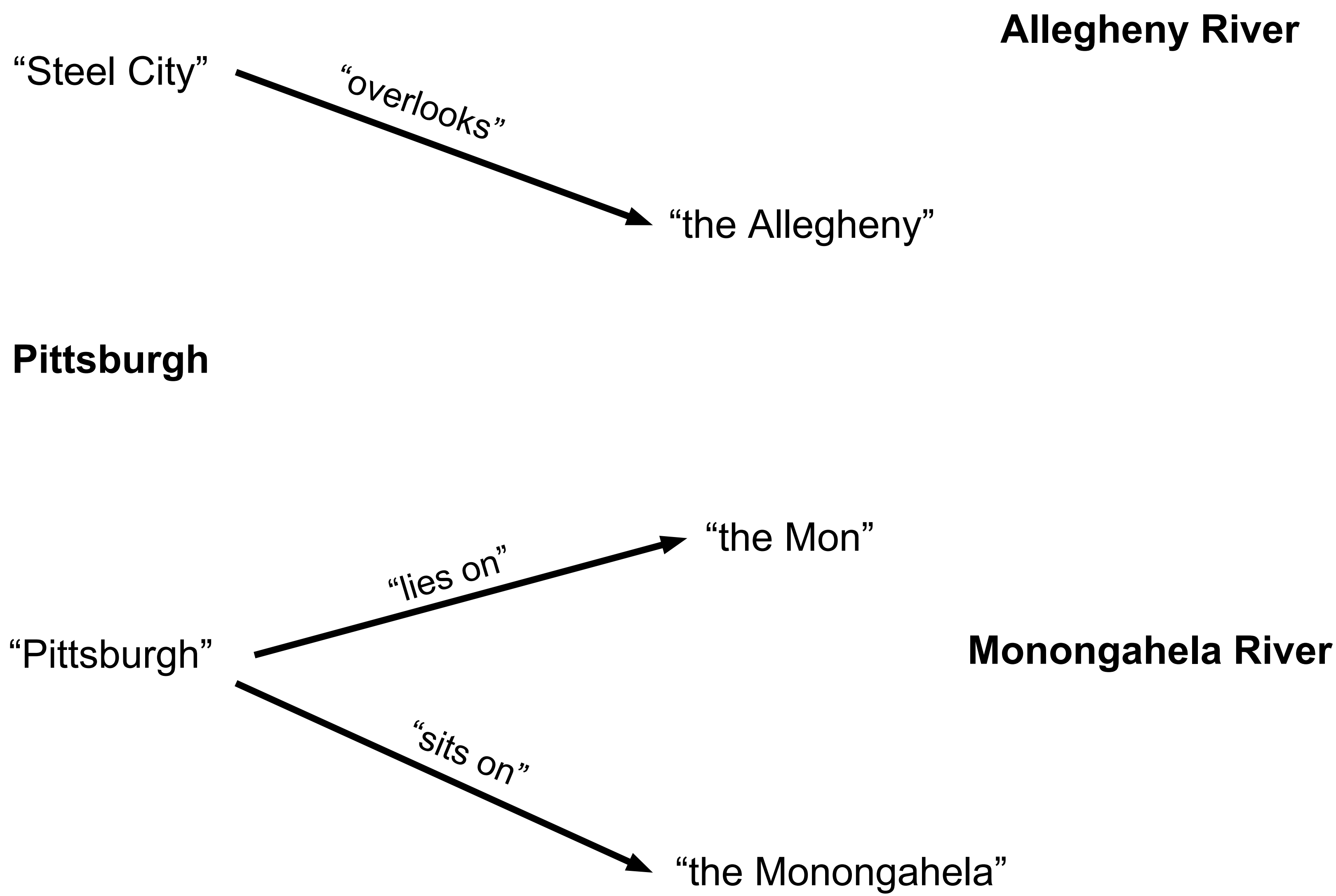
"Pittsburgh"

"the Monongahela"

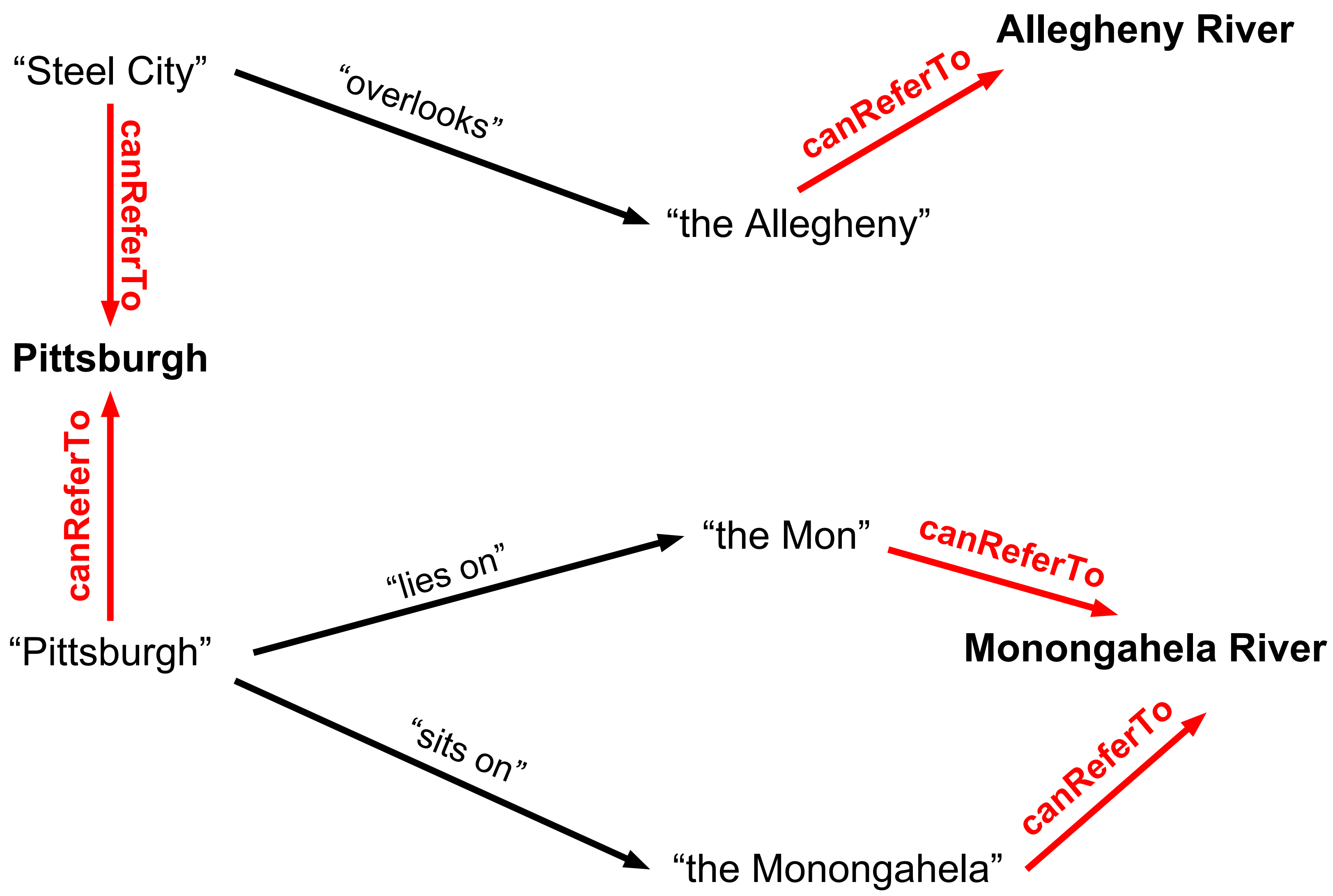
“Steel City overlooks the Allegheny”
“Pittsburgh lies on the Mon”
“Pittsburgh sits on the Monongahela”



“Steel City overlooks the Allegheny”
“Pittsburgh lies on the Mon”
“Pittsburgh sits on the Monongahela”



"Steel City overlooks the Allegheny"
"Pittsburgh lies on the Mon"
"Pittsburgh sits on the Monongahela"



KB + Text

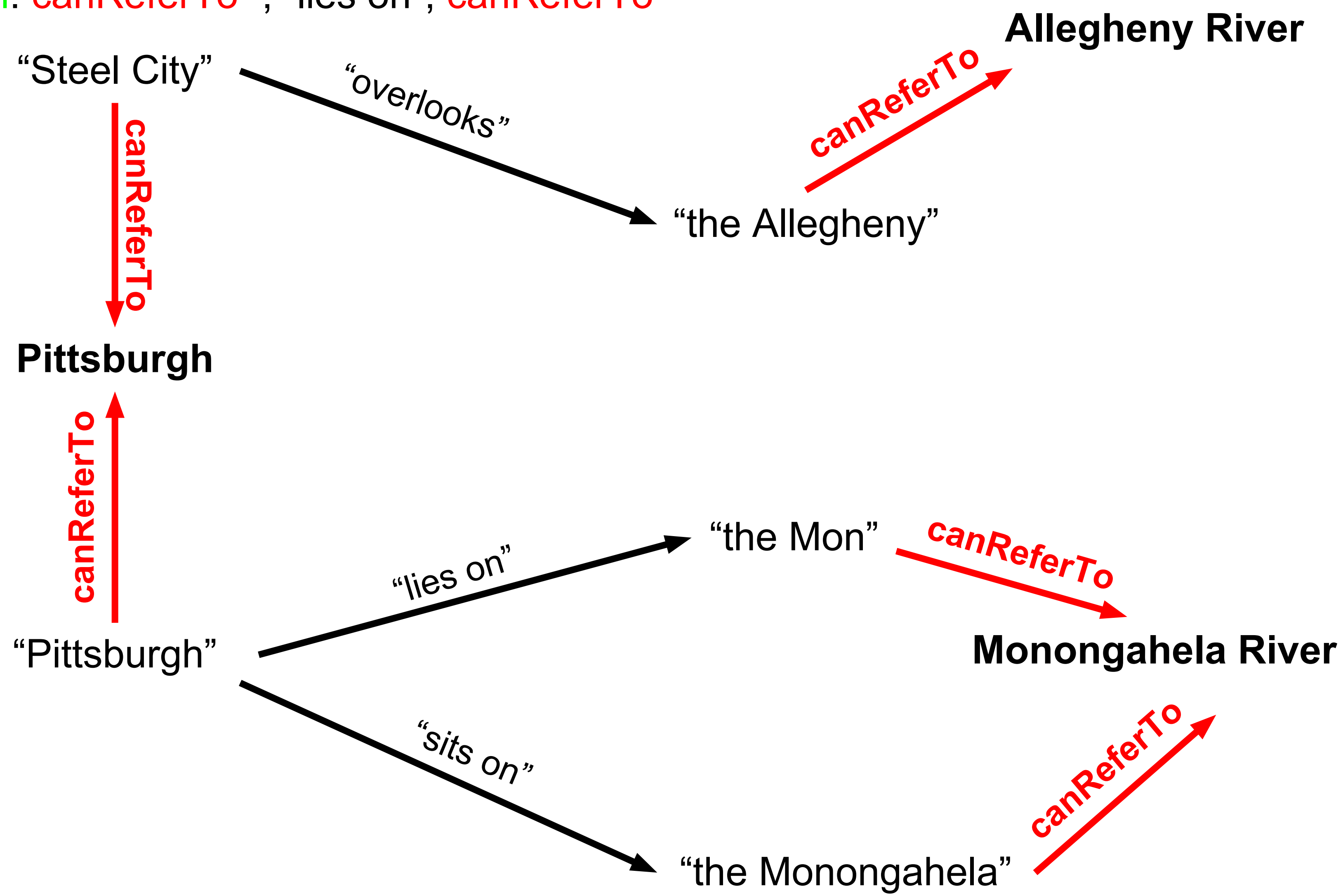
Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “lies on”, canReferTo

KB + Text

Relation: cityLiesOnRiver

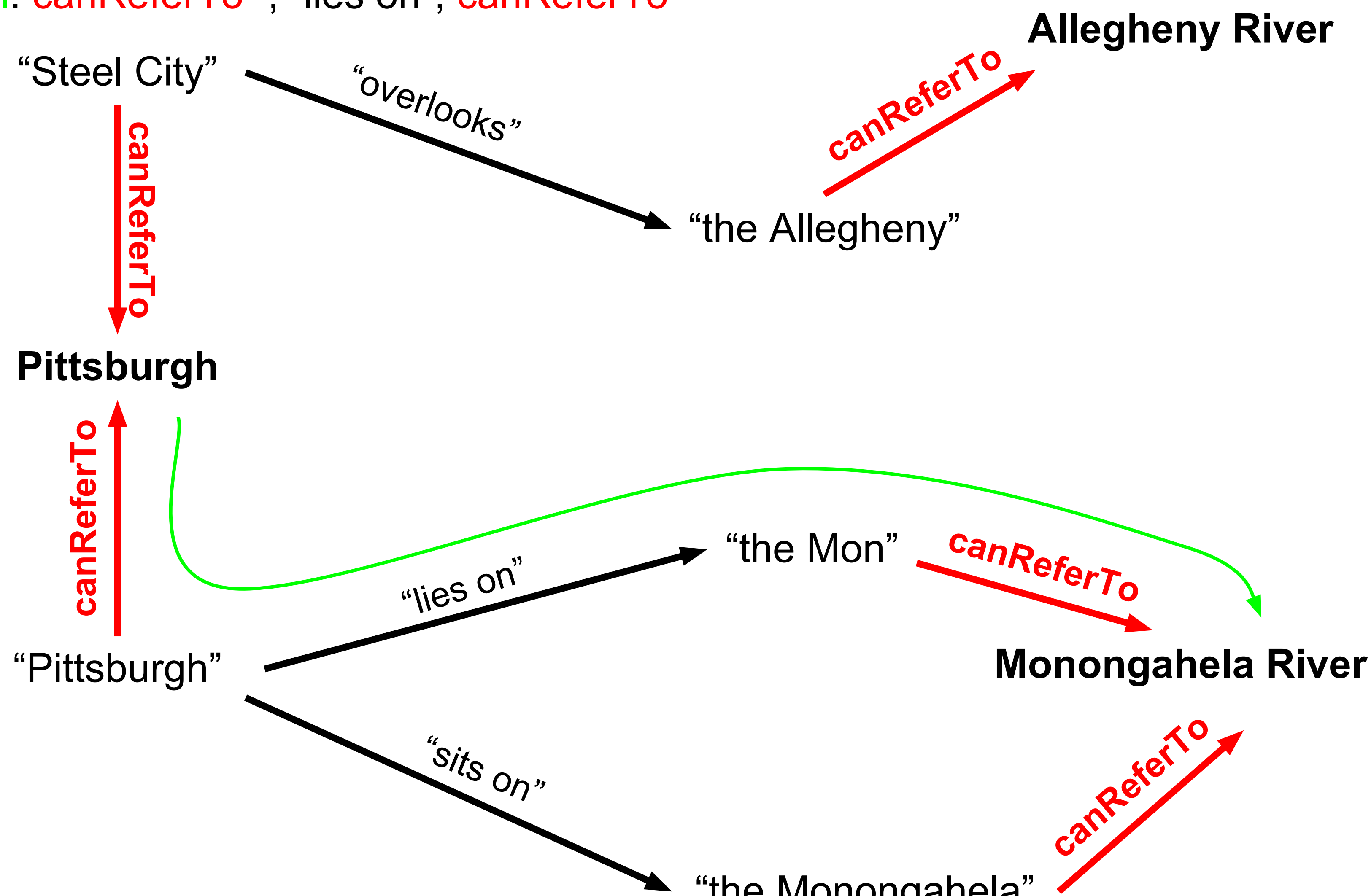
Path: canReferTo^{-1}, “lies on”, canReferTo



KB + Text

Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “lies on”, canReferTo



KB + Text

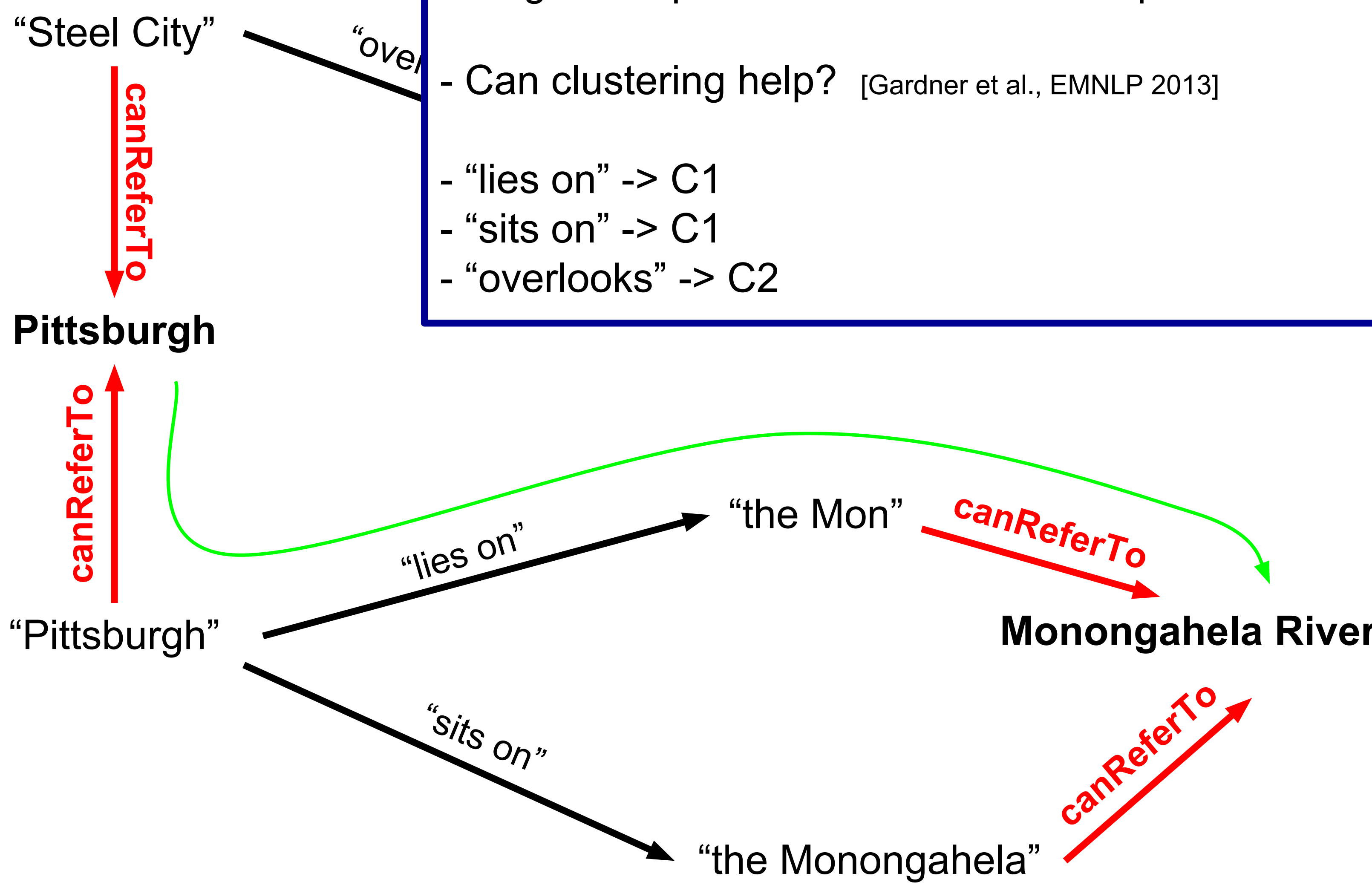
Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “lies on”,

- Large data problem: verb forms are sparse!

- Can clustering help? [Gardner et al., EMNLP 2013]

- “lies on” -> C1
- “sits on” -> C1
- “overlooks” -> C2



KB + Clustered Text

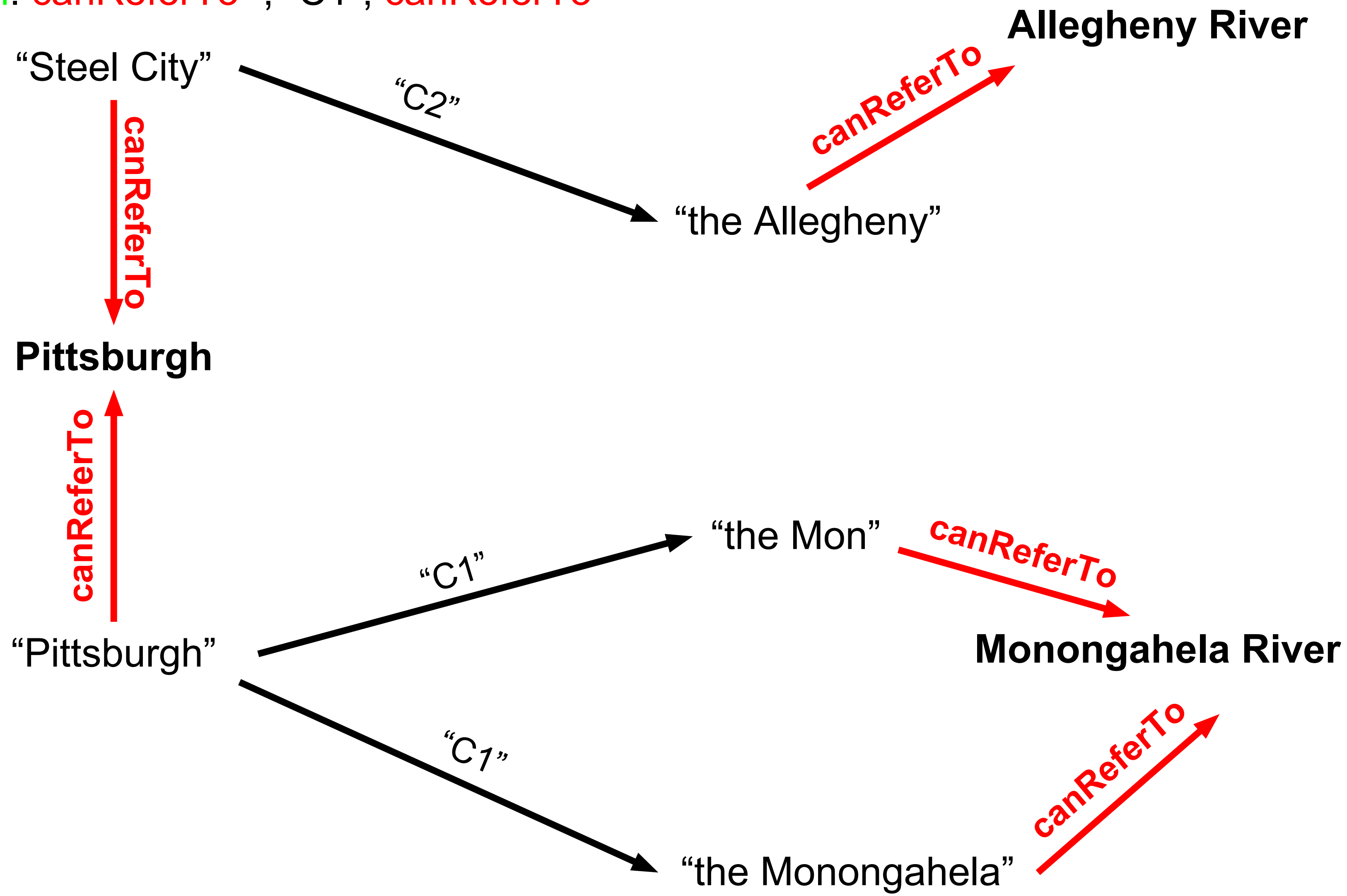
Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “C1”, canReferTo

KB + Clustered Text

Relation: cityLiesOnRiver

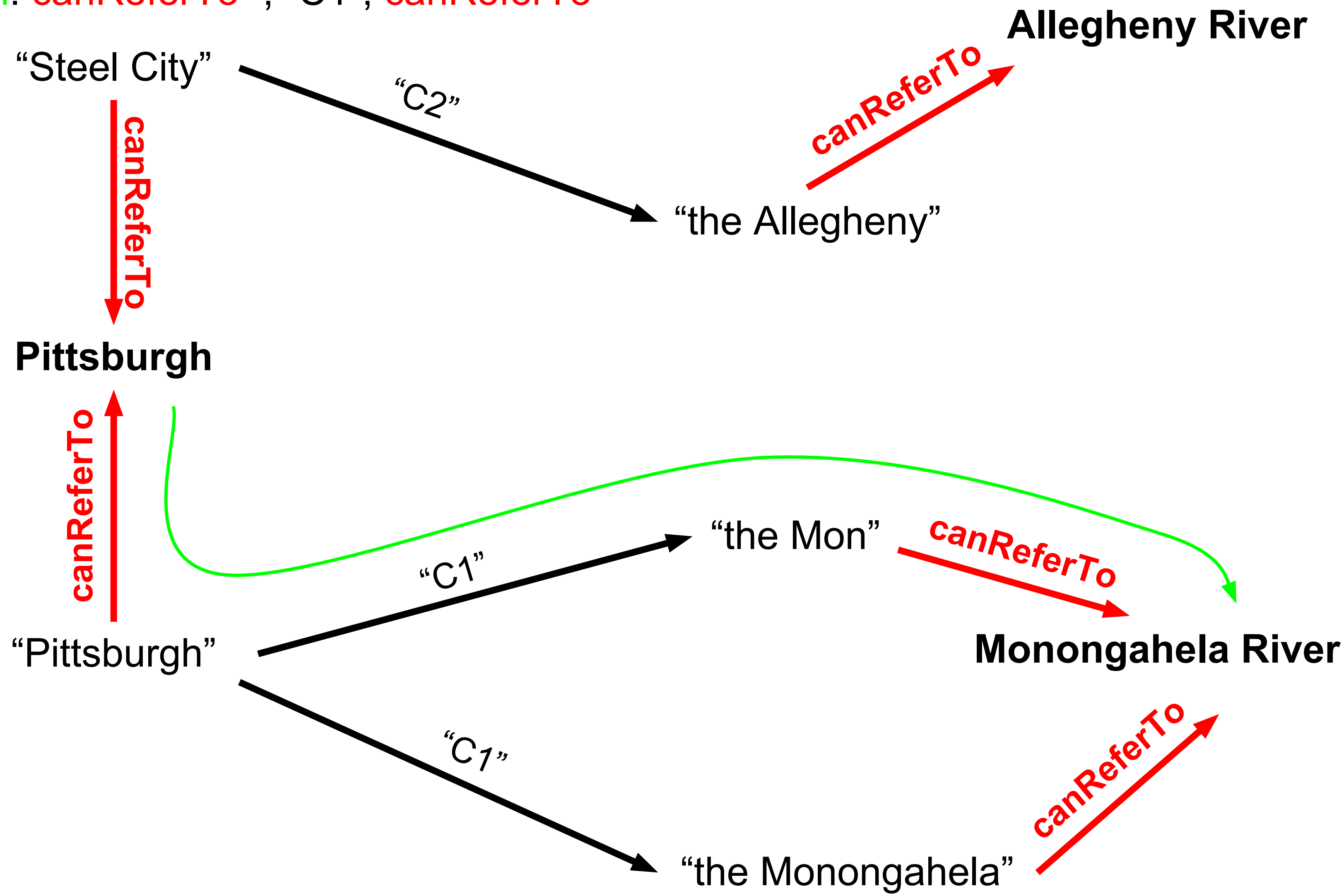
Path: canReferTo^{-1}, “C1”, canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

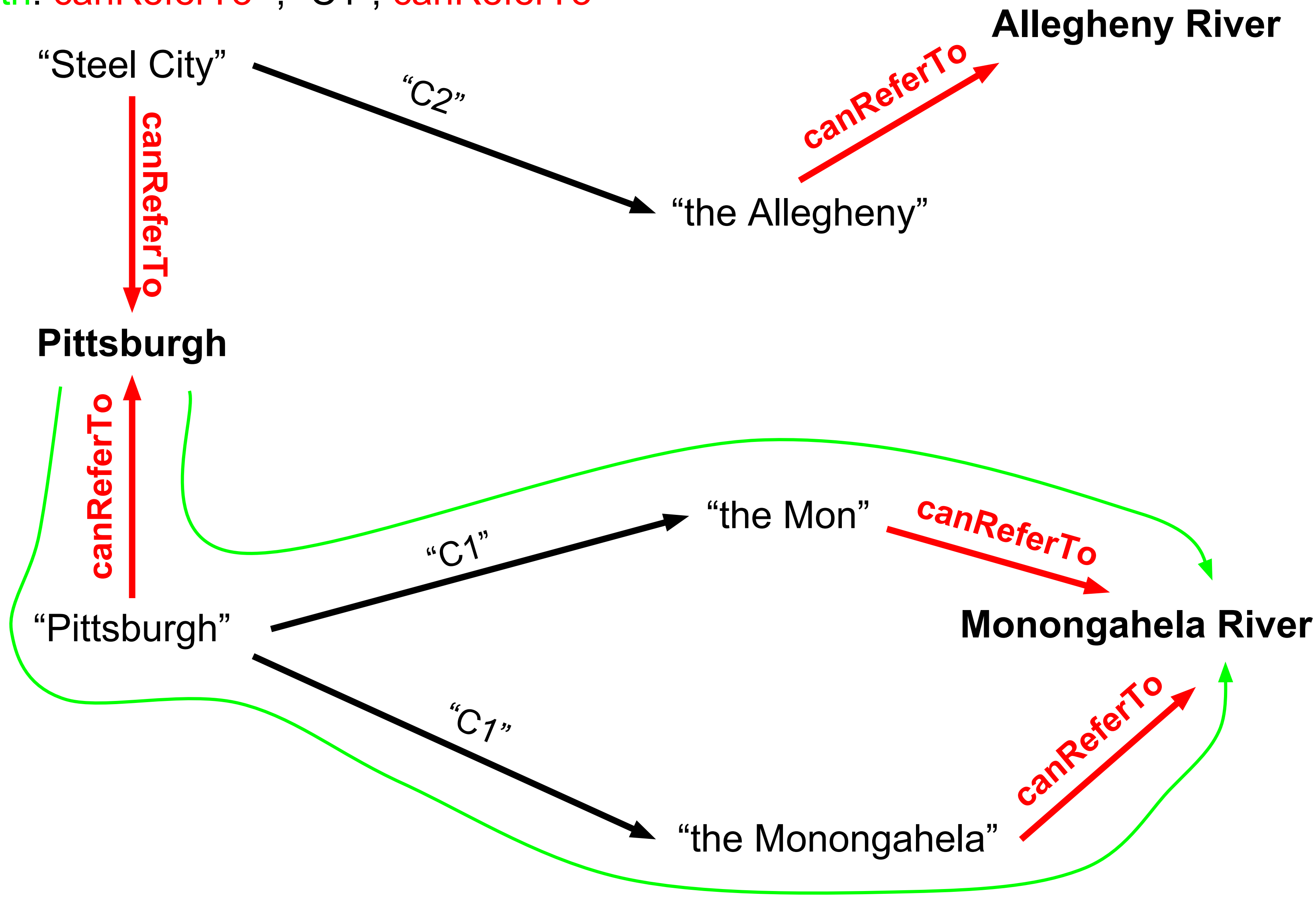
Path: canReferTo^{-1}, “C1”, canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

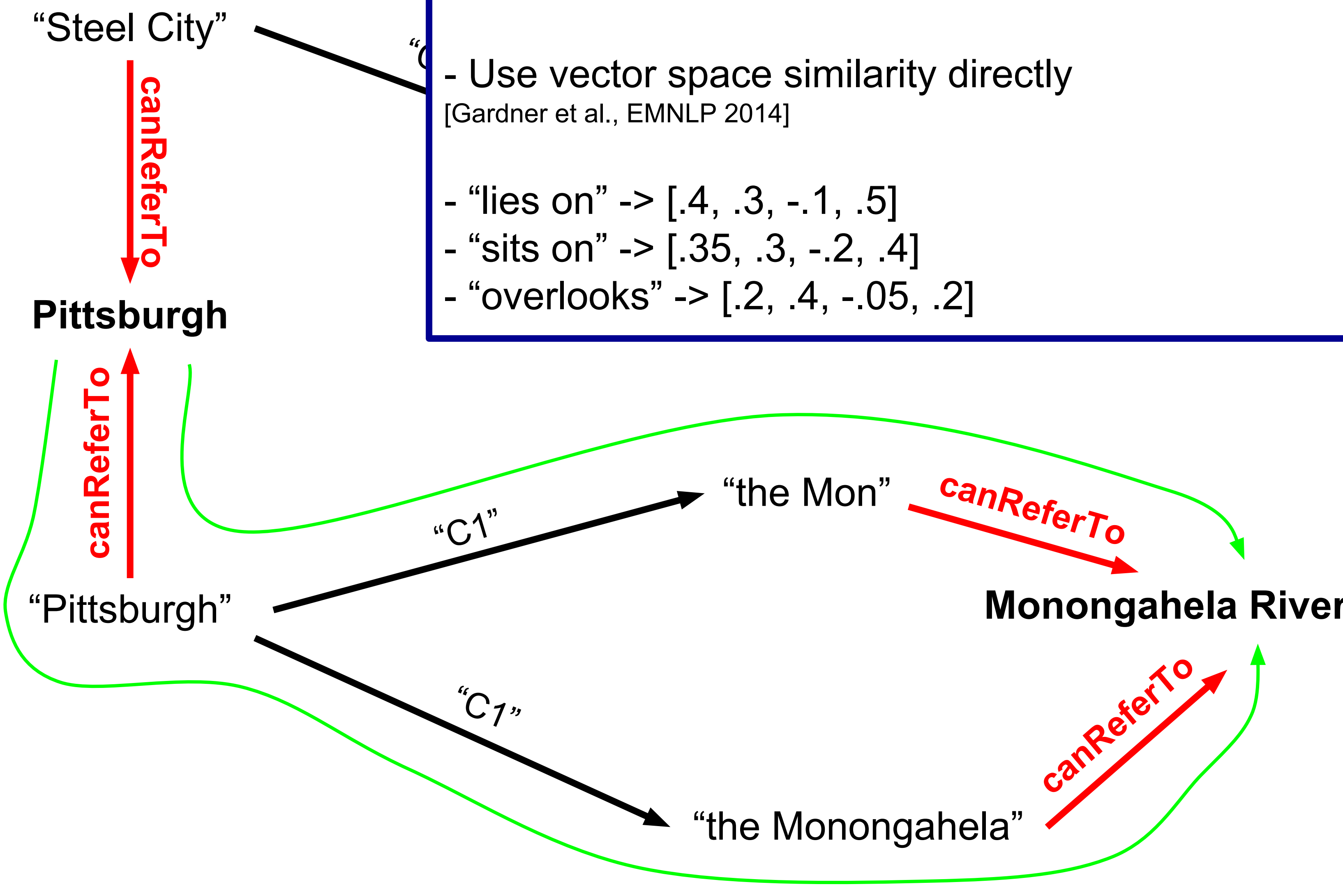
Path: canReferTo^{-1}, “C1”, canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “C1”, ca



- Much better, but still can be sparse

- Use vector space similarity directly
[Gardner et al., EMNLP 2014]

- “lies on” -> [.4, .3, -.1, .5]
- “sits on” -> [.35, .3, -.2, .4]
- “overlooks” -> [.2, .4, -.05, .2]

KB + Text Vectors

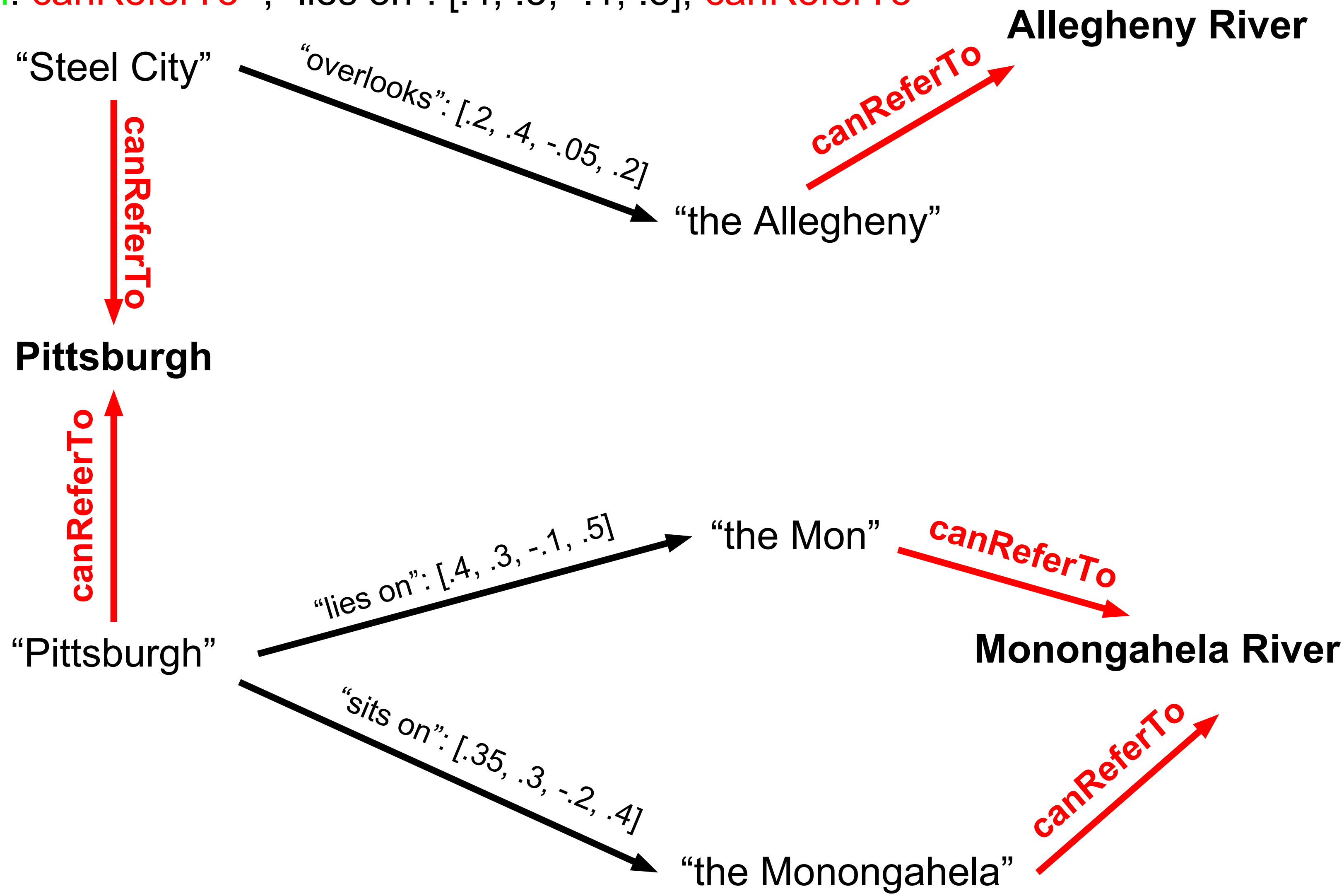
Relation: cityLiesOnRiver

Path: canReferTo^{-1}, “lies on”: [.4, .3, -.1, .5], canReferTo

KB + Text Vectors

Relation: cityLiesOnRiver

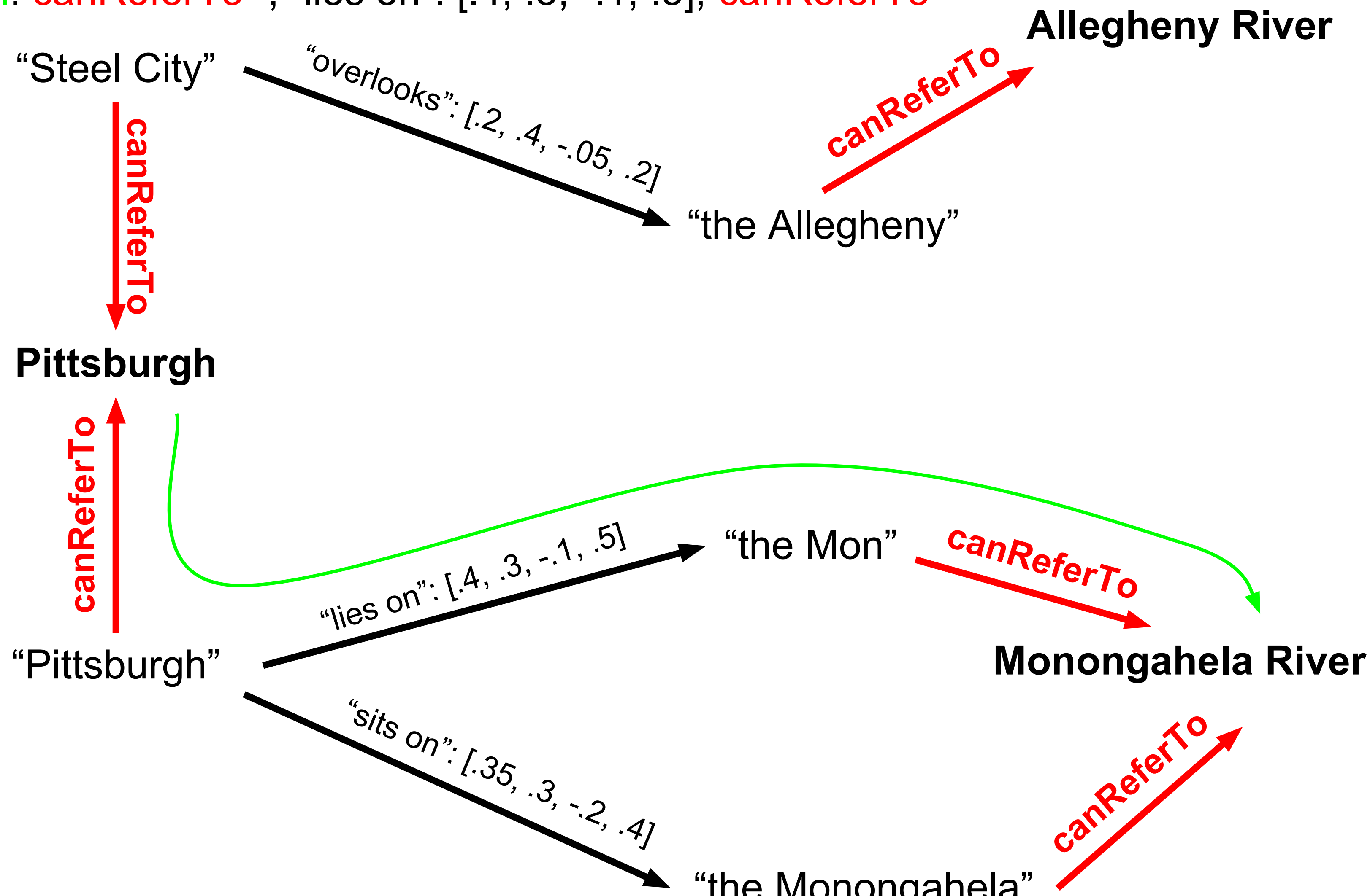
Path: canReferTo^{-1}, "lies on": [.4, .3, -.1, .5], canReferTo



KB + Text Vectors

Relation: cityLiesOnRiver

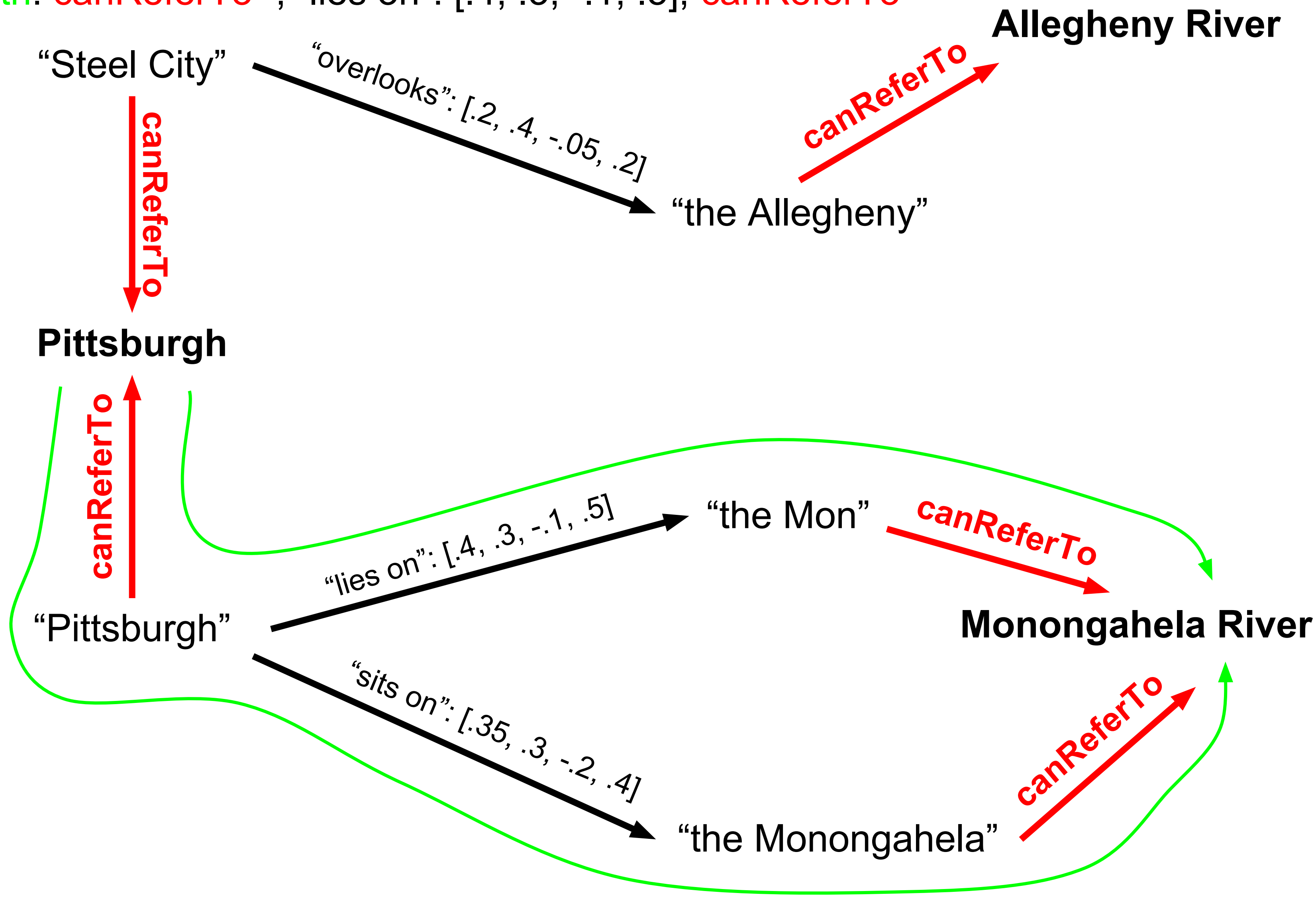
Path: canReferTo^{-1}, “lies on”: [.4, .3, -.1, .5], canReferTo



KB + Text Vectors

Relation: cityLiesOnRiver

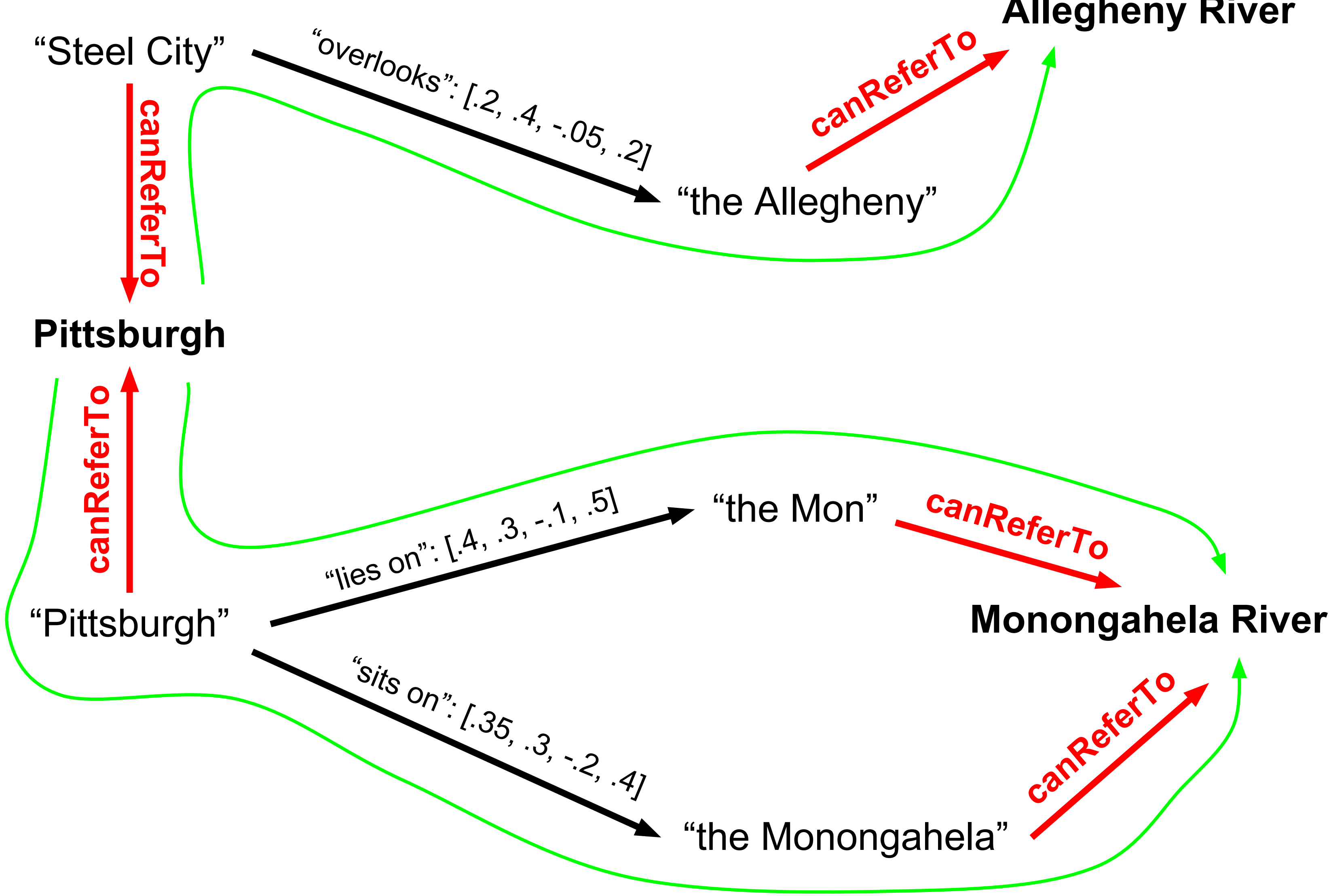
Path: canReferTo^{-1}, “lies on”: [.4, .3, -.1, .5], canReferTo



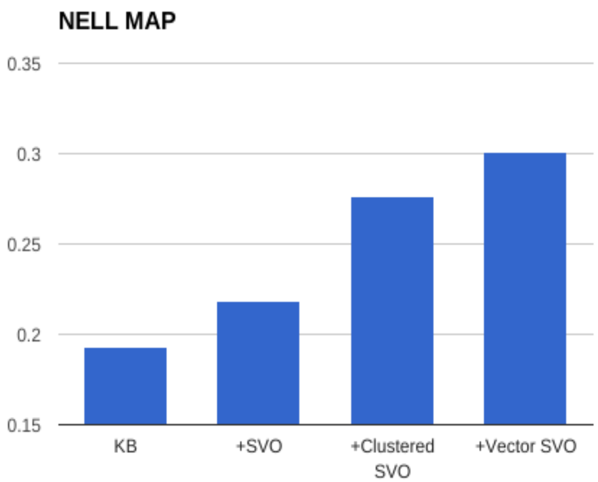
KB + Text Vectors

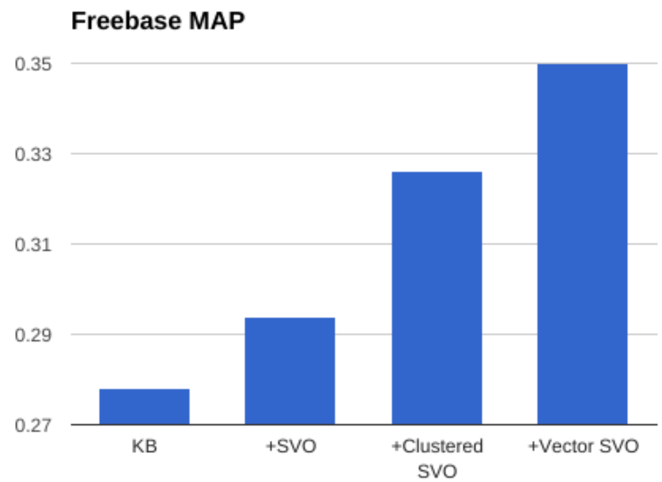
Relation: cityLiesOnRiver

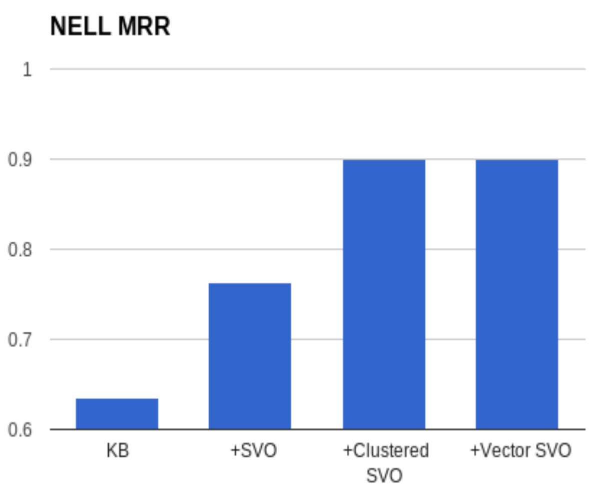
Path: canReferTo^{-1}, “lies on”: [.4, .3, -.1, .5], canReferTo

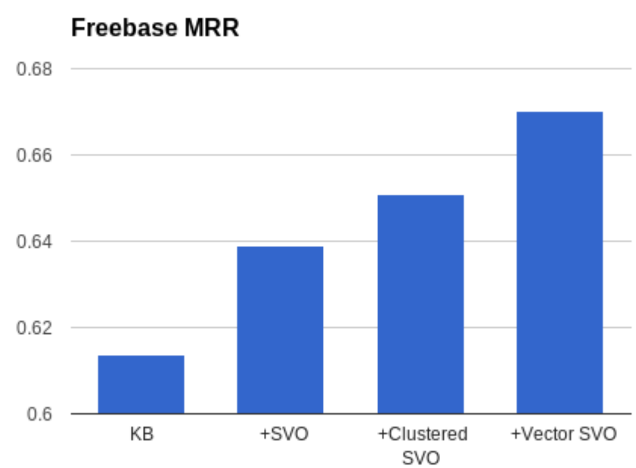


Empirical Results









Questions?