
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?

Verizon 3G 2:14 PM 93%

“ Flights overhead now ”


Here is what I found:

Input interpretation

flights seen from current location

Result

	altitude	angle
Southwest Airlines flight 4504	35 000 feet	45° up
Southwest Airlines flight 987	34 700 feet	43° up
Delta Air	33 000 feet	42° up



Why knowledge bases?

Verizon 3G 2:14 PM 93%

“Flights overhead now”


Here is what I found:

Input interpretation

flights seen from current location

Result


	altitude	angle
Southwest Airlines flight 4504	35 000 feet	45° up
Southwest Airlines flight 987	34 700 feet	43° up
Delta Air	33 000 feet	42° up



Maps Play YouTube News Gmail Drive Calendar More

quentin tarantino movies

Grant Simmons 0 + Share



P
T
2

[Quentin Tarantino - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Quentin_Tarantino

His films include **Reservoir Dogs** (1992), **Pulp Fiction** (1994), Jackie Brown (1997), ... Tarantino's contribution to the Grindhouse project was titled **Death Proof**. ... Tarantino's 2009 film **Inglourious Basterds** is the story of a group of guerrilla U.S.
.co.uk/film/filmblog/2011/may/05/quentin-tarantino-django-unchained-script

[Quentin Tarantino filmography - Death Proof - Jackie Brown - New Beverly Cinema](#)

[Quentin Tarantino - IMDb](#)

www.imdb.com/name/nm0000233/

In January of 1992, **Reservoir Dogs** appeared at the Sundance Film Festival, by ... up Dogs success with **Pulp Fiction** which premiered at the Cannes film festival... ..

Quentin Tarantino Still of Quentin Tarantino in Django Unchained Still of ...

[Django Unchained - Biography - Death Proof - Kill Bill: Vol. 3](#)

[Quentin Tarantino movies, photos, movie reviews, filmography, and ...](#)
www.allmovie.com/artist/quentin-tarantino-p113658

Word-of-mouth on **Reservoir Dogs** began to build at the 1992 Sundance Film Festival, ... During 1993, Tarantino wrote and directed his next feature, **Pulp Fiction**, ... Grindhouse into two portions: the first half, **Death Proof**, directed by Tarantino, ...

[Quentin Tarantino Movies | Cracked.com](#)

www.cracked.com/funny-1906-quentin-tarantino-movies/

Quentin Tarantino used to recommend movies while working in a video rental store, making him the most successful "annoying a**hole on about 'good' ...

Quentin Tarantino



Quentin Jerome Tarantino is an American film director, screenwriter, producer, cinematographer and actor. [Wikipedia](#)

Born: March 27, 1963 (age 49), Knoxville

Upcoming movies: [Django Unchained](#), [Kill Bill: Vol. 3](#)

Awards: Palme d'Or, National Society of Film Critics Award for Best Director, [More](#)

Education: [Narbonne High School](#), Fleming Junior High School

Song	Year	Album
Interview	1997	Tarantino Connection
Pulp Fiction	1998	Interview Picture Disc
Jackie Brown	1998	Interview Picture Disc

Books

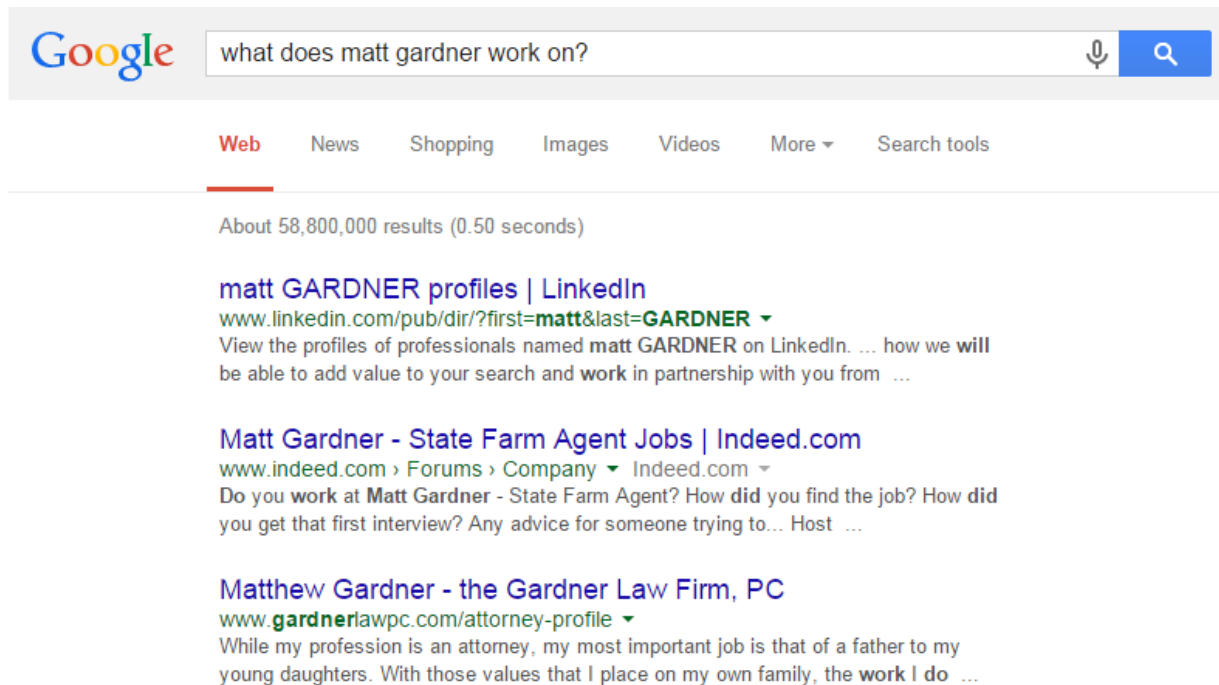


But ...

But ...



But ...



So we do inference

So we do inference

- Predict missing facts given what we know

So we do inference

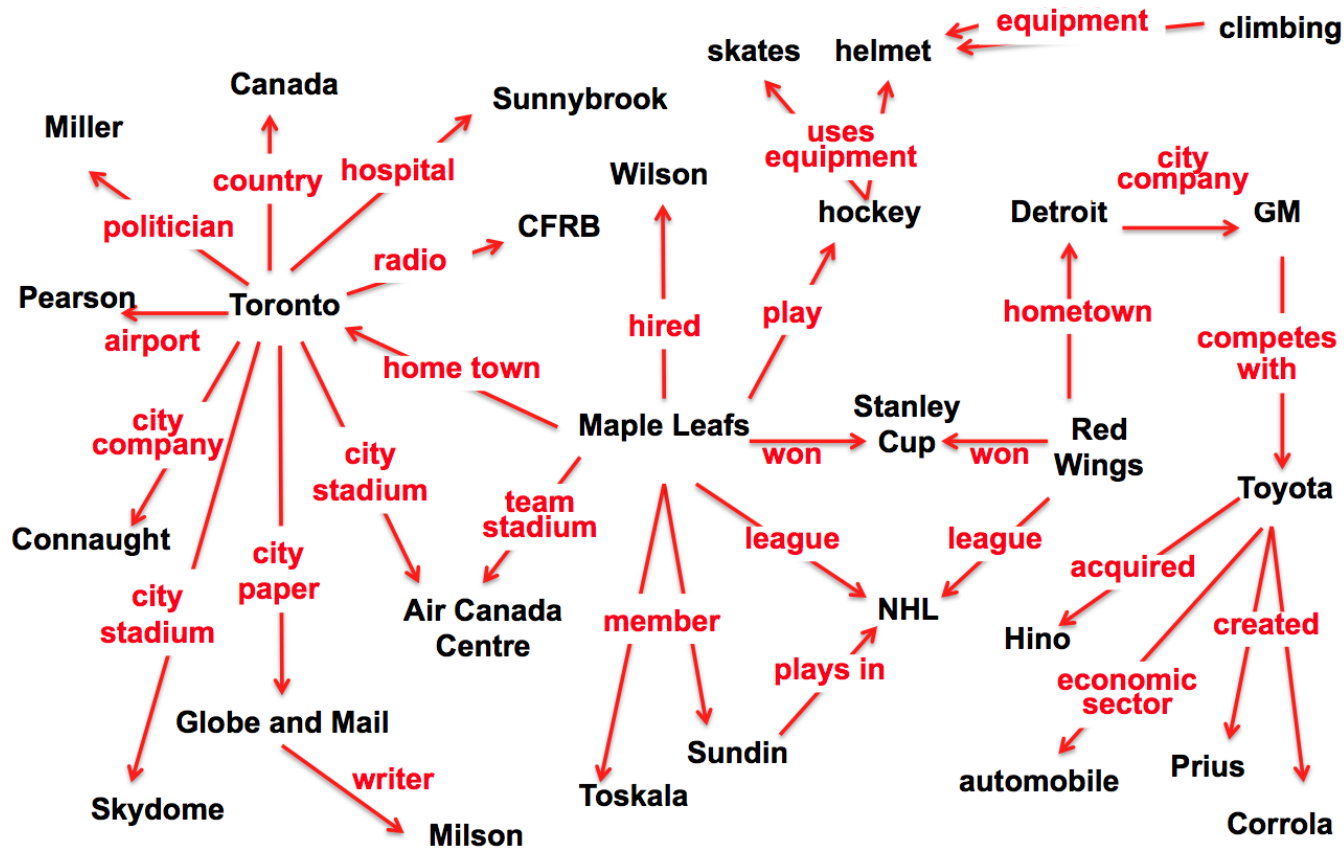
- Predict missing facts given what we know
- Lots of ways to do this, today we'll talk about random walk inference, or the Path Ranking Algorithm (PRA)

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]

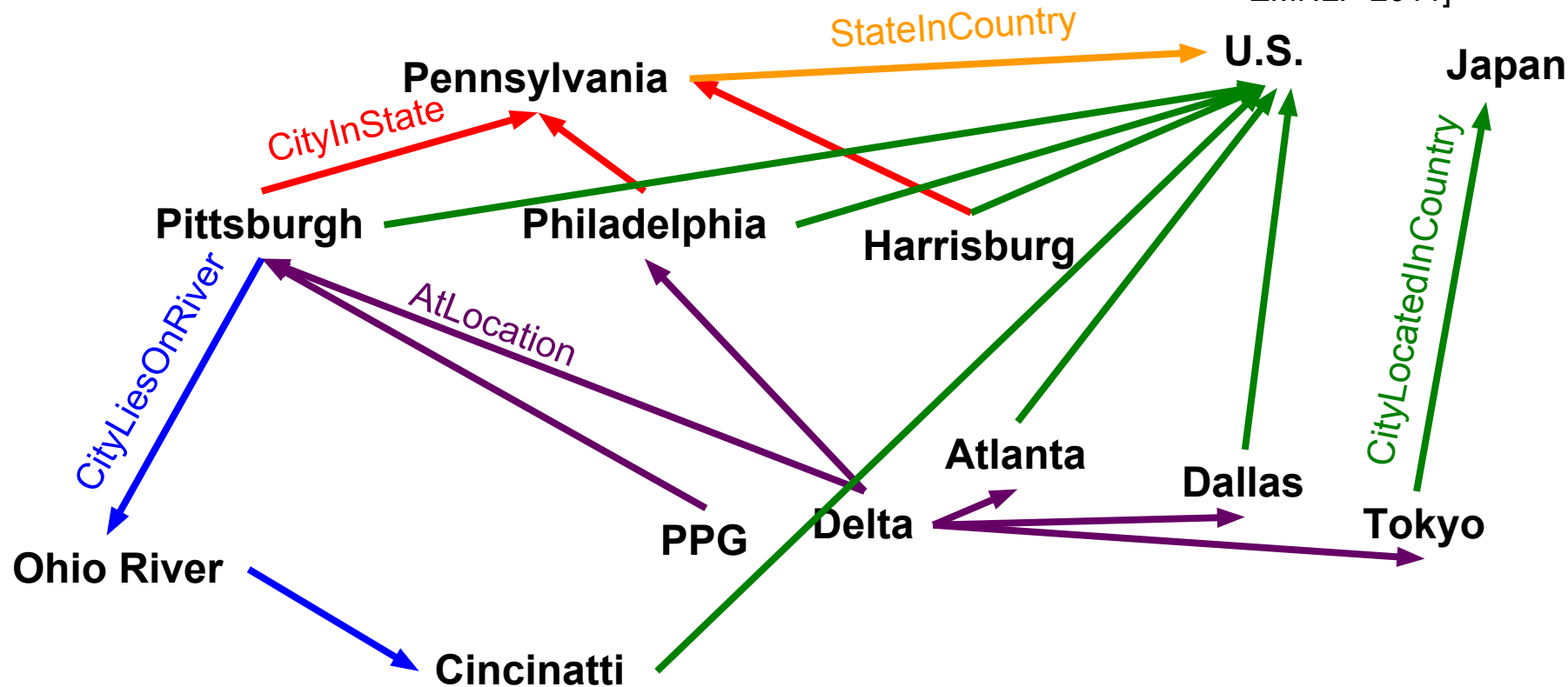


If: x_1 — **competes with** (x_1, x_2) — x_2 — **economic sector** (x_2, x_3) — x_3

Then: **economic sector** (x_1, x_3)

CityLocatedInCountry - Selecting path features

[Lao et al,
EMNLP 2011]

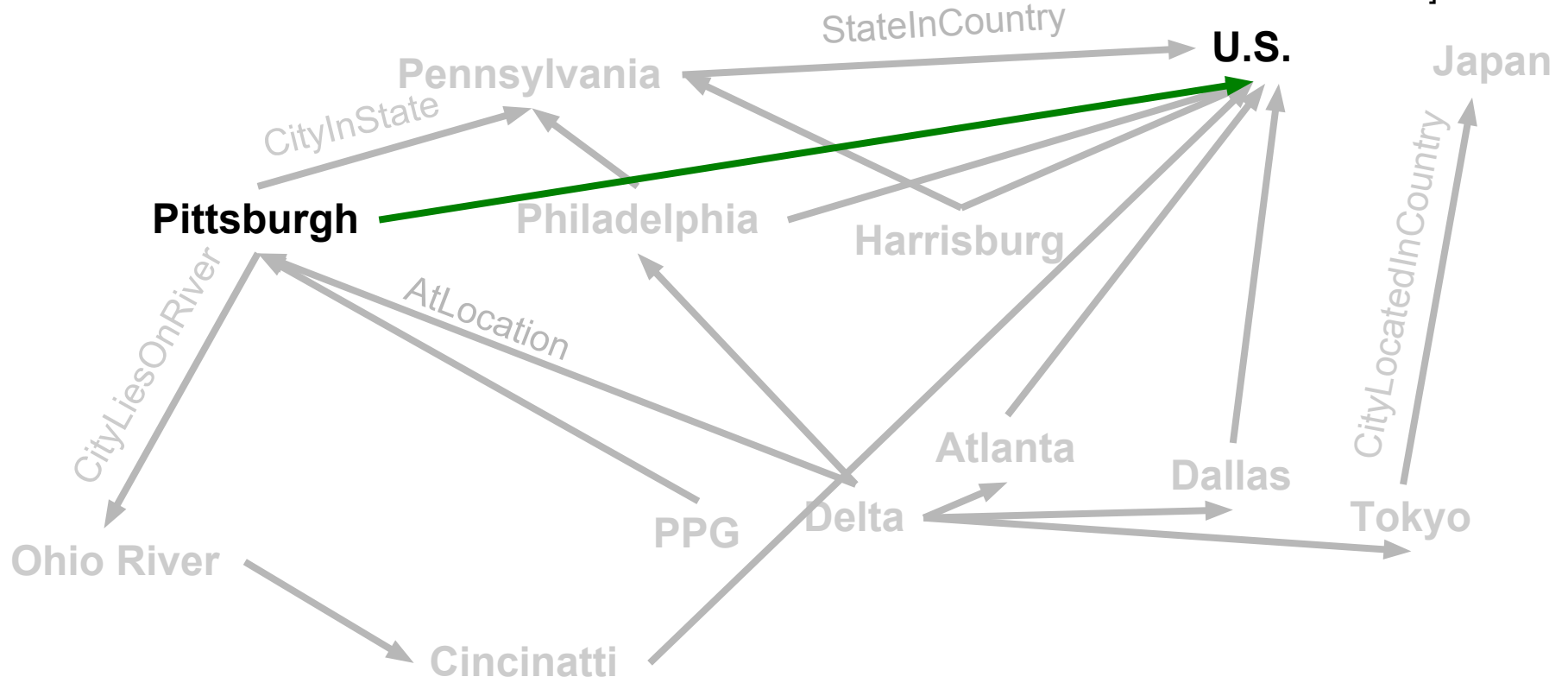


Path type

Count

CityLocatedInCountry - Selecting path features

[Lao et al,
EMNLP 2011]

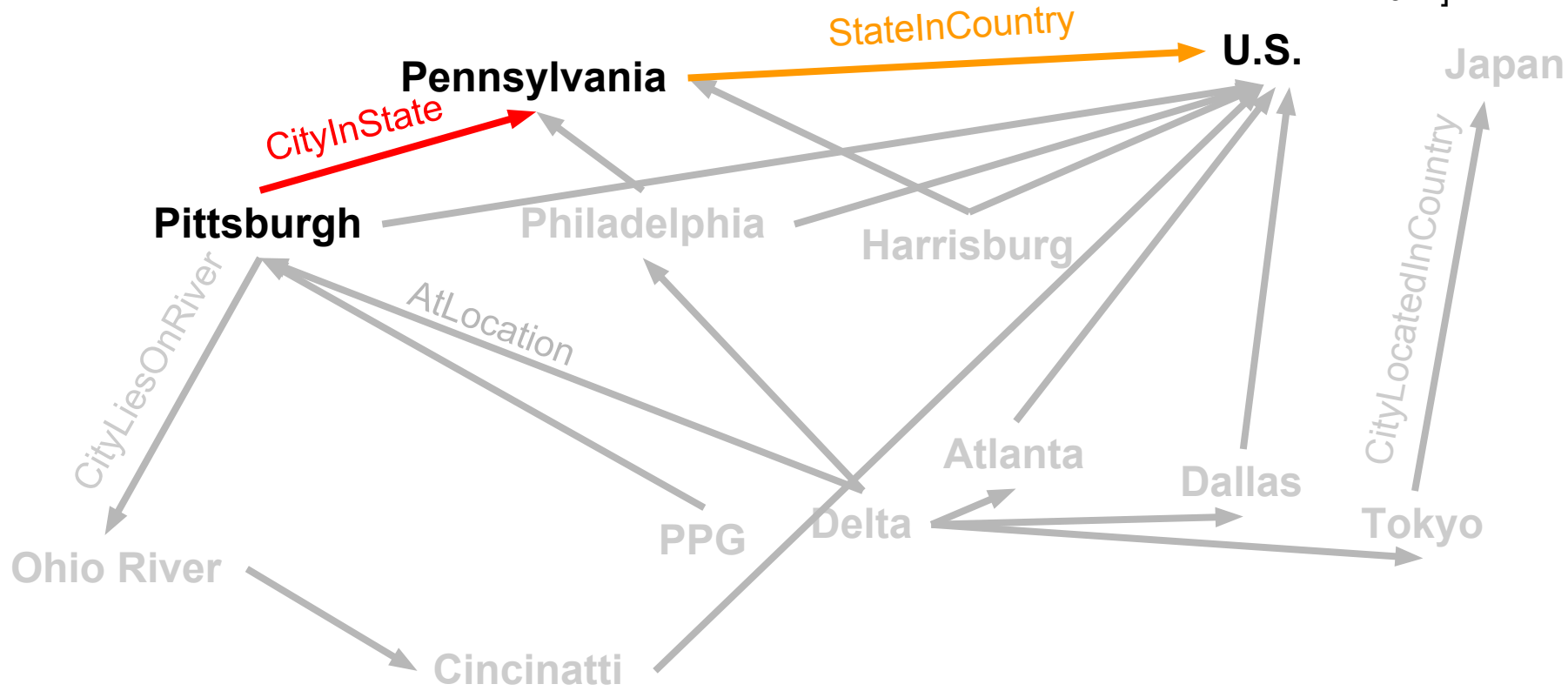


Path type

Count

CityLocatedInCountry - Selecting path features

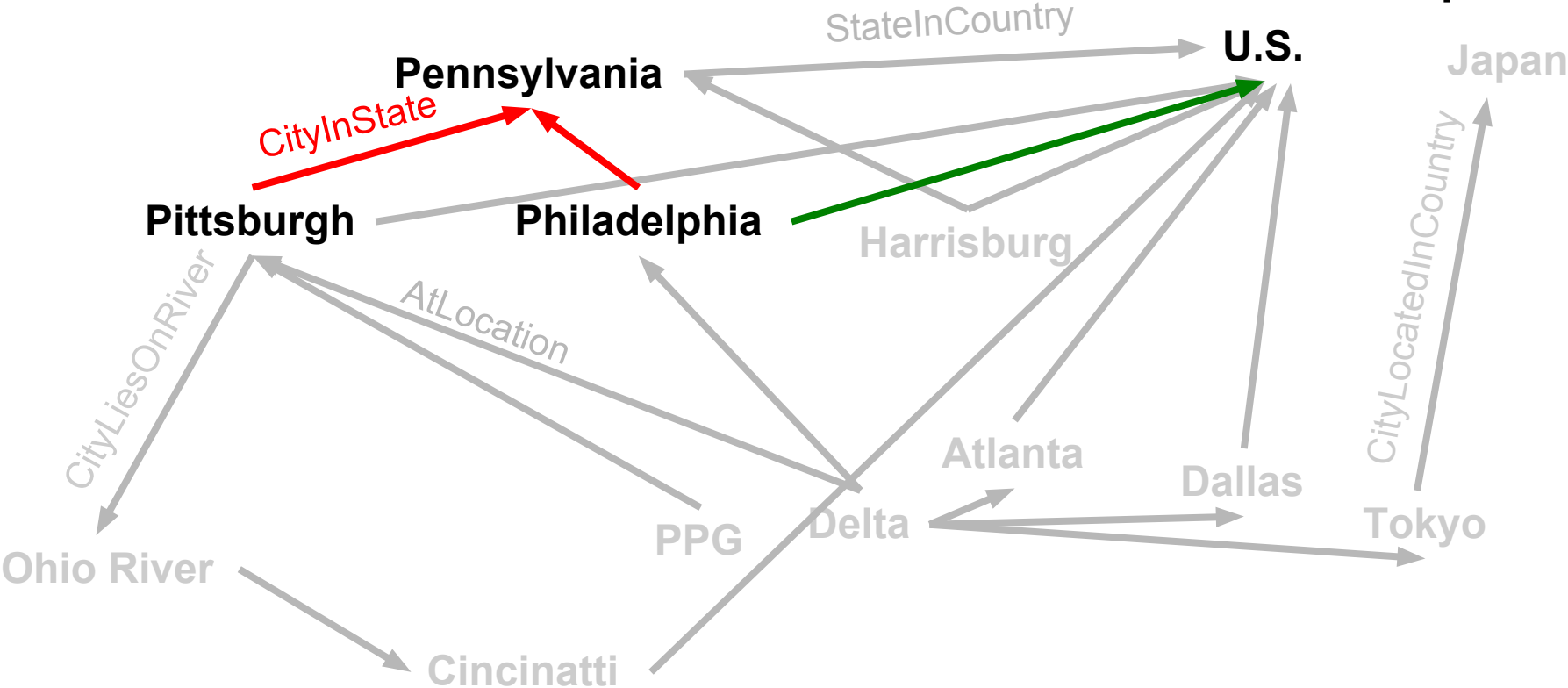
[Lao et al,
EMNLP 2011]



Path type	Count
CityInState, StateInCountry	1

CityLocatedInCountry - Selecting path features

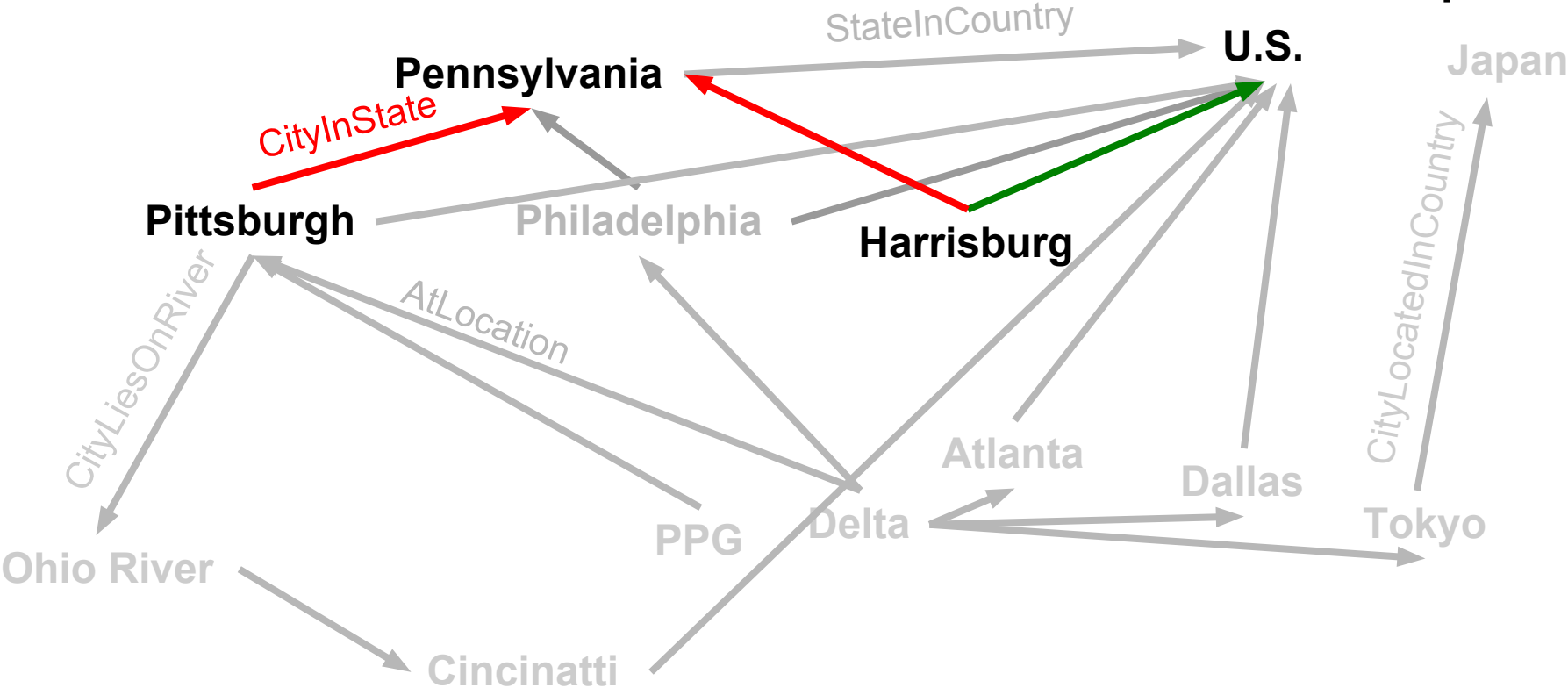
[Lao et al,
EMNLP 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState ⁻¹ , CityLocatedInCountry	1

CityLocatedInCountry - Selecting path features

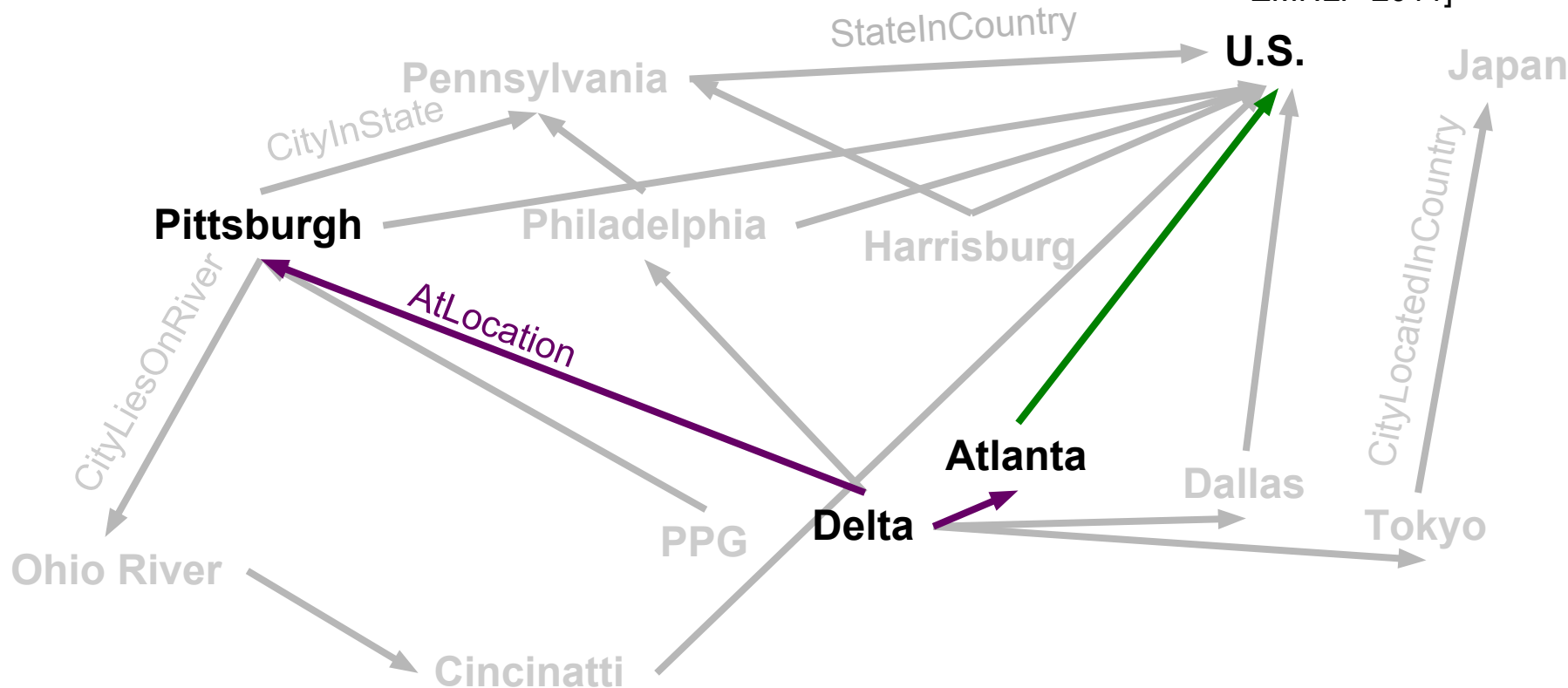
[Lao et al,
EMNLP 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState ⁻¹ , CityLocatedInCountry	2

CityLocatedInCountry - Selecting path features

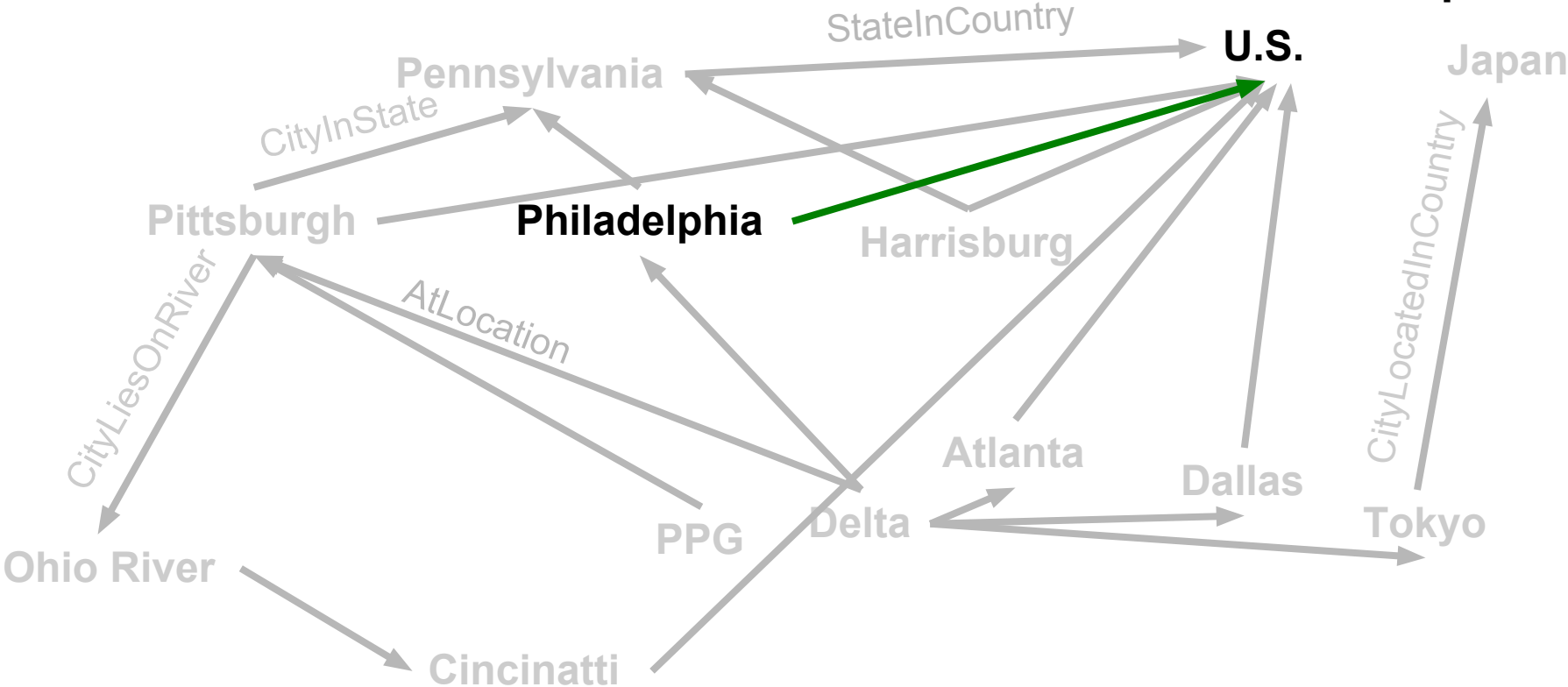
[Lao et al,
EMNLP 2011]



Path type	Count
CityInState, StateInCountry	1
CityInState, CityInState ⁻¹ , CityLocatedInCountry	2
AtLocation ⁻¹ , AtLocation, CityLocatedInCountry	1

CityLocatedInCountry - Selecting path features

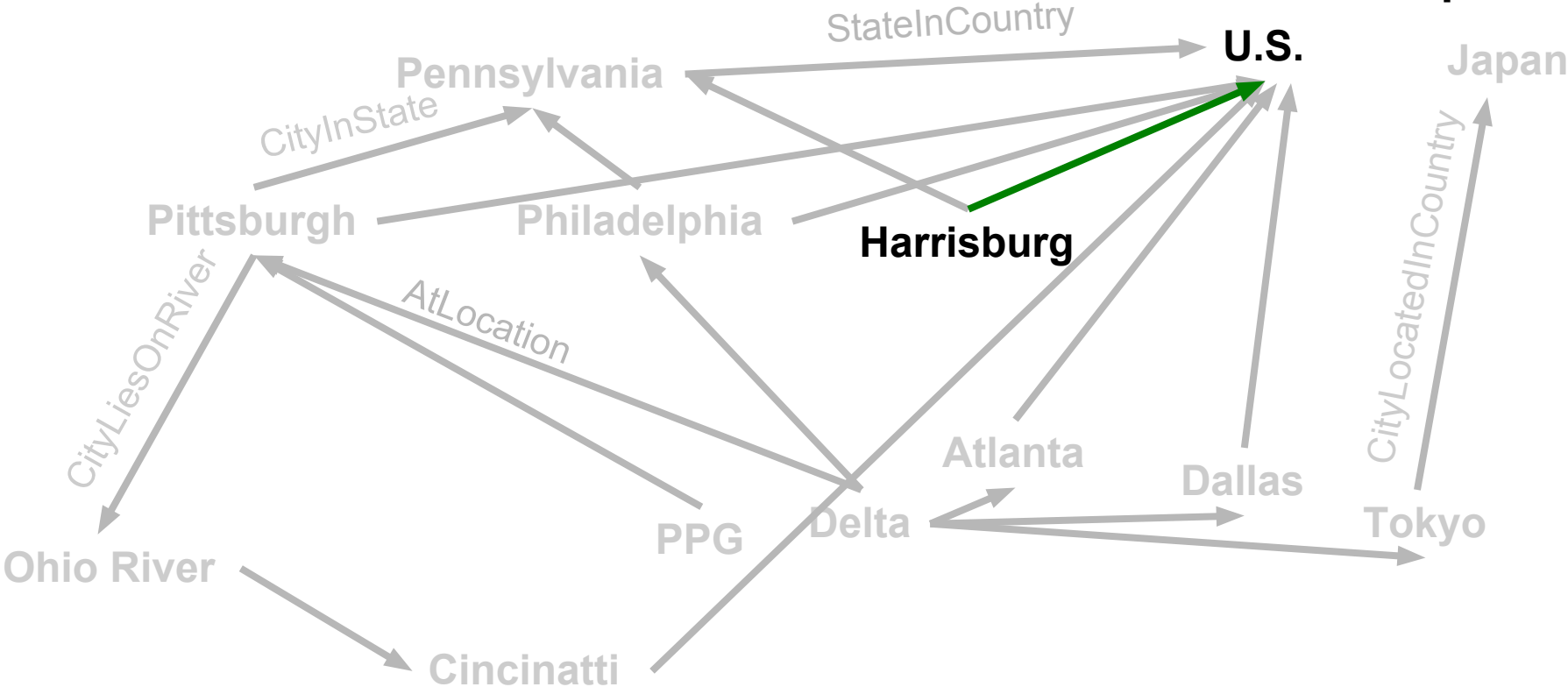
[Lao et al,
EMNLP 2011]



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

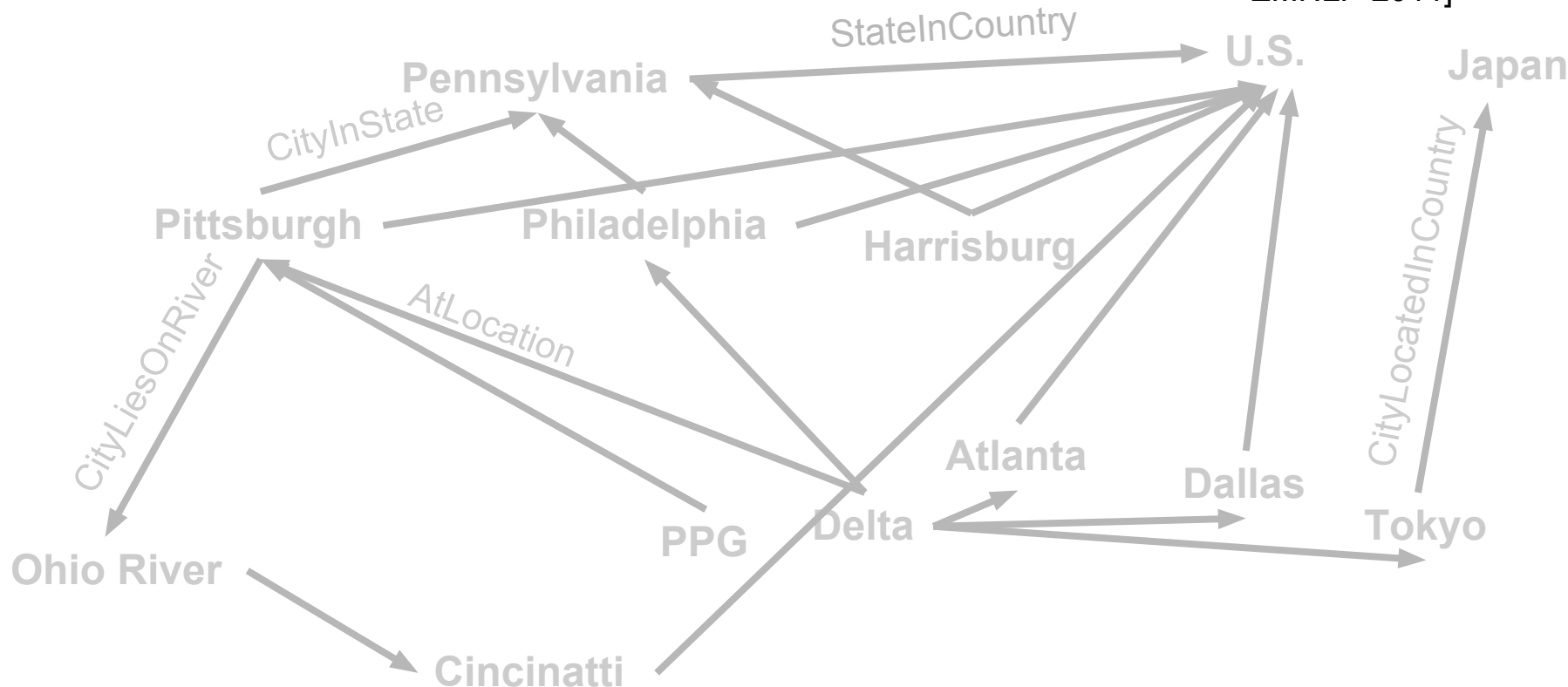
[Lao et al,
EMNLP 2011]



Path type	Count
CityInState, StateInCountry	3,892
CityInState, CityInState ⁻¹ , CityLocatedInCountry	234
AtLocation ⁻¹ , AtLocation, CityLocatedInCountry	1,543
CityLiesOnRiver, 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, CityLiesOnRiver ⁻¹ , CityLocatedInCountry	298
...	...

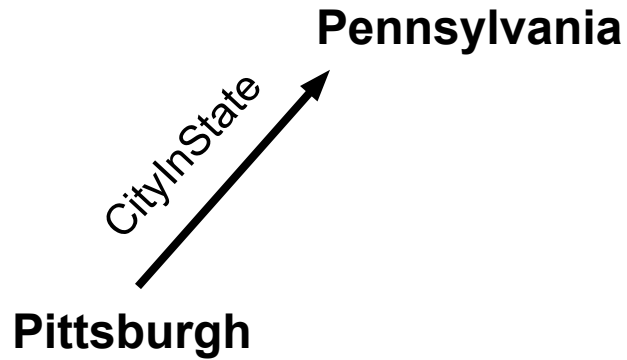
Select the most frequent path types, and keep them as features in the model

Pittsburgh

Feature = Typed Path

CityInState, CityInState^{-1} , CityLocatedInCountry

Feature Value



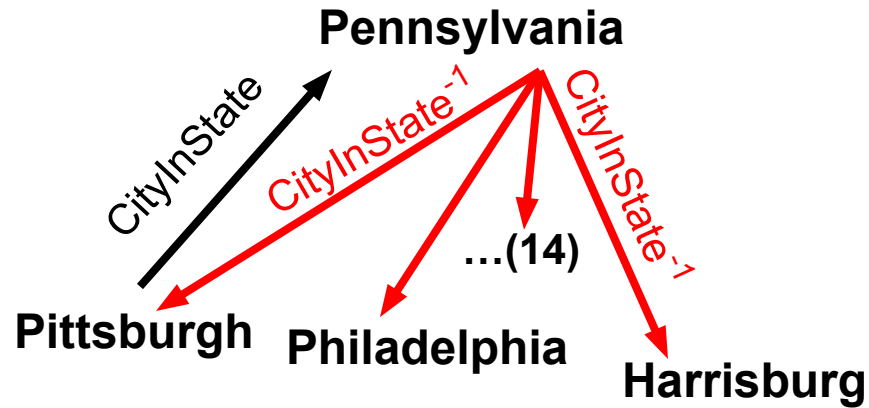
Feature = Typed Path

CityInState, **CityInState⁻¹**, CityLocatedInCountry

Feature Value

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



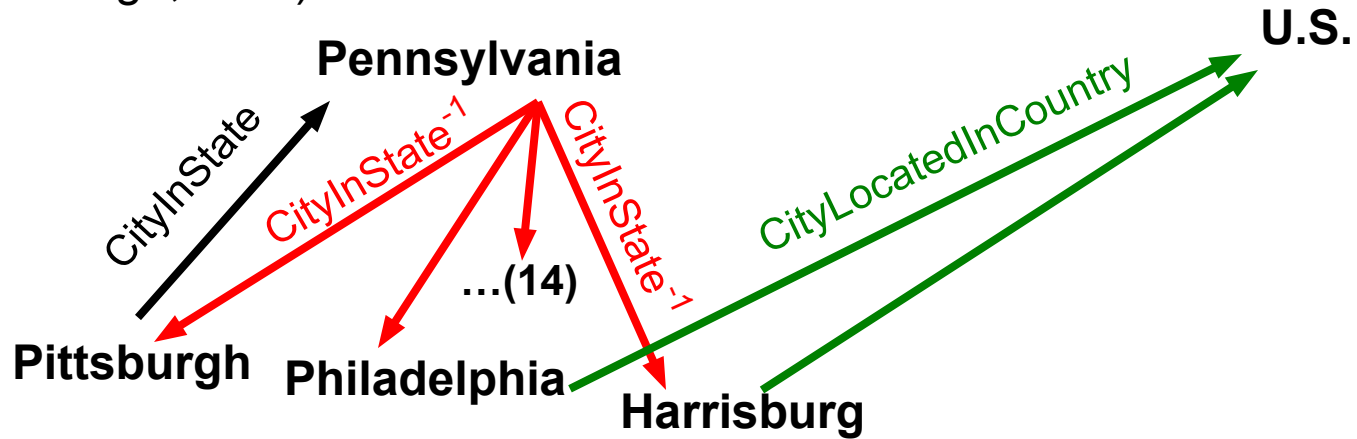
Feature = Typed Path

CityInState, **CityInState⁻¹**, CityLocatedInCountry

Feature Value

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



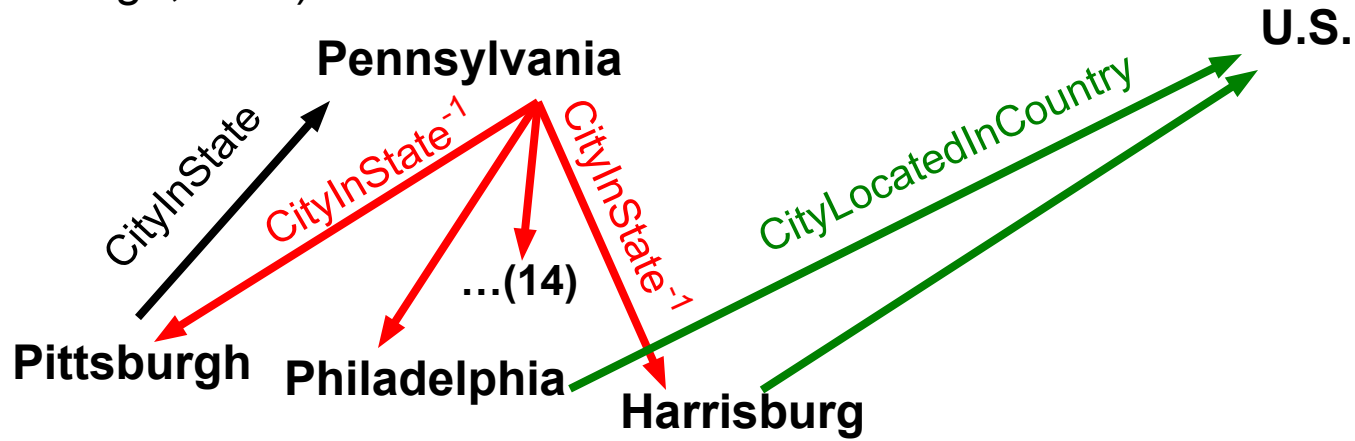
Feature = Typed Path

CityInState, **CityInState⁻¹**, **CityLocatedInCountry**

Feature Value

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



$\text{Pr}(\text{U.S.} \mid \text{Pittsburgh, TypedPath})$

Feature = Typed Path

CityInState, **CityInState⁻¹**, CityLocatedInCountry

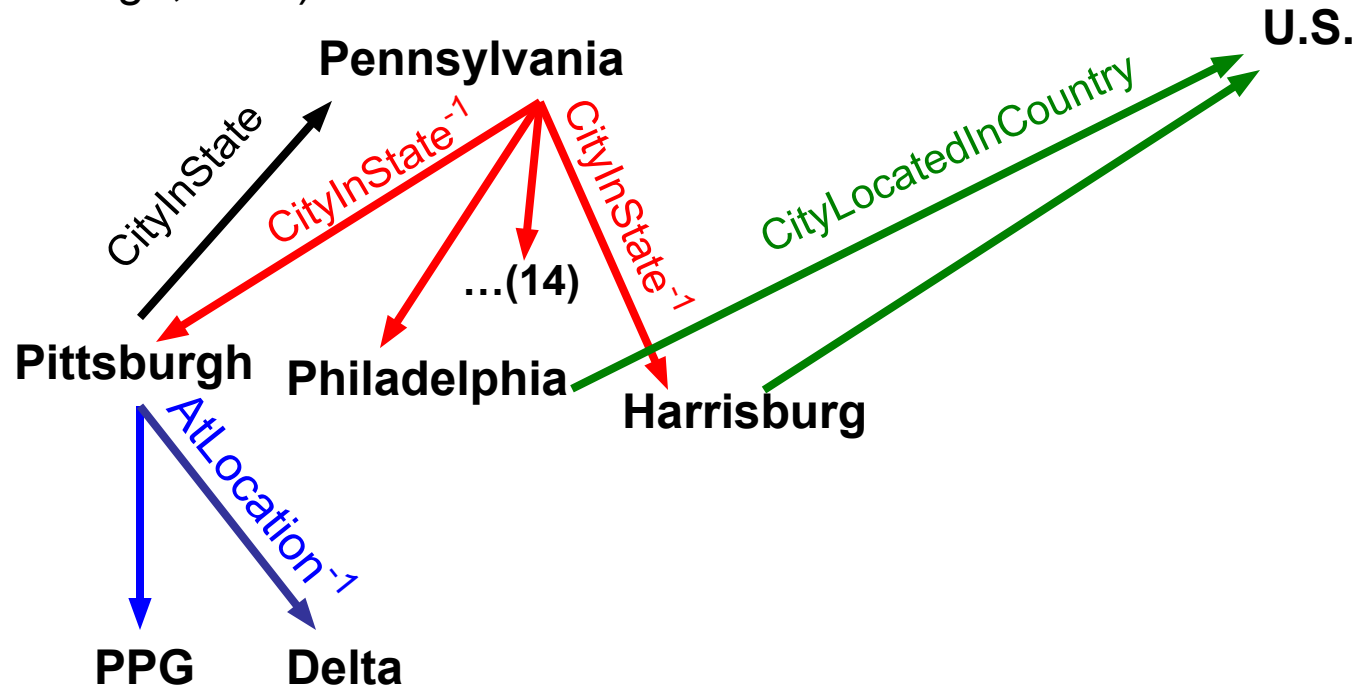
Feature Value

1.0



CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



Feature = Typed Path

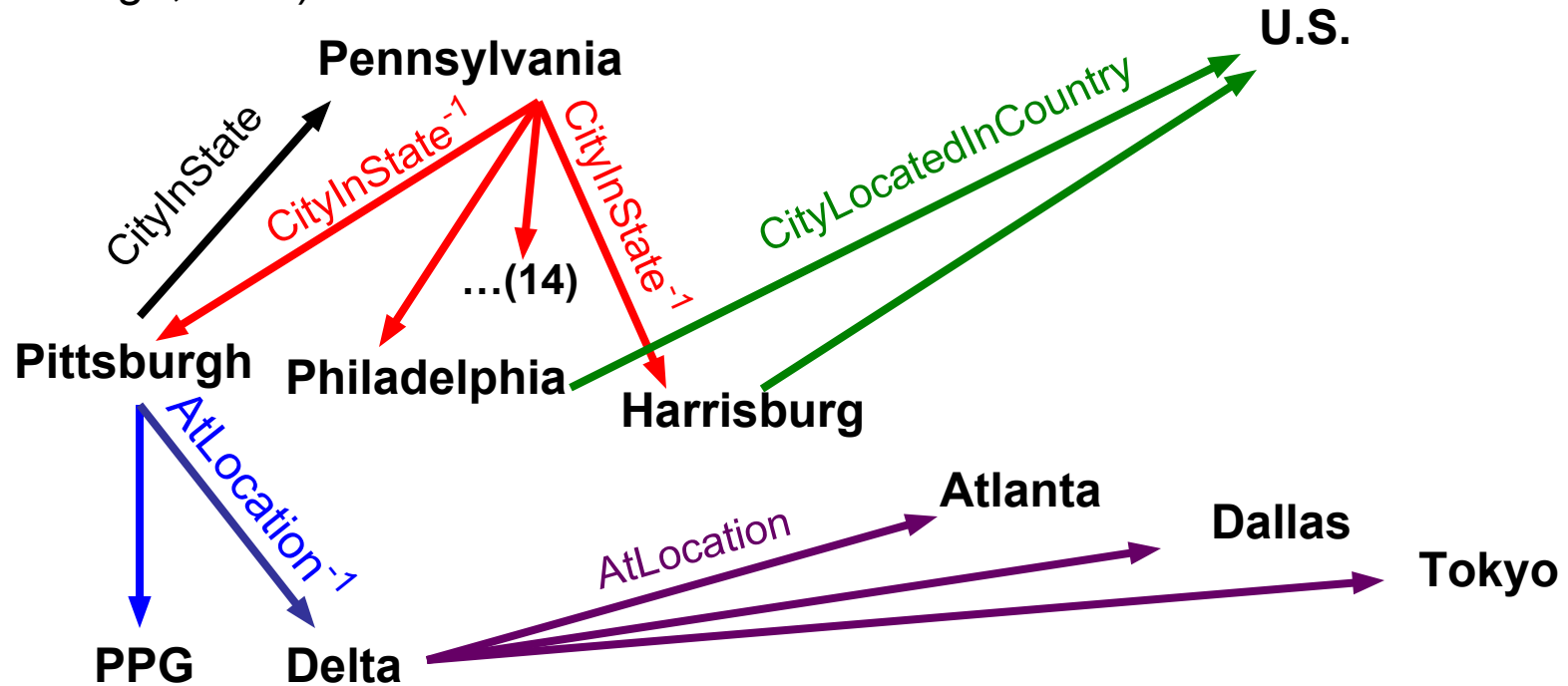
CityInState, **CityInState⁻¹**, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value

1.0

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



Feature = Typed Path

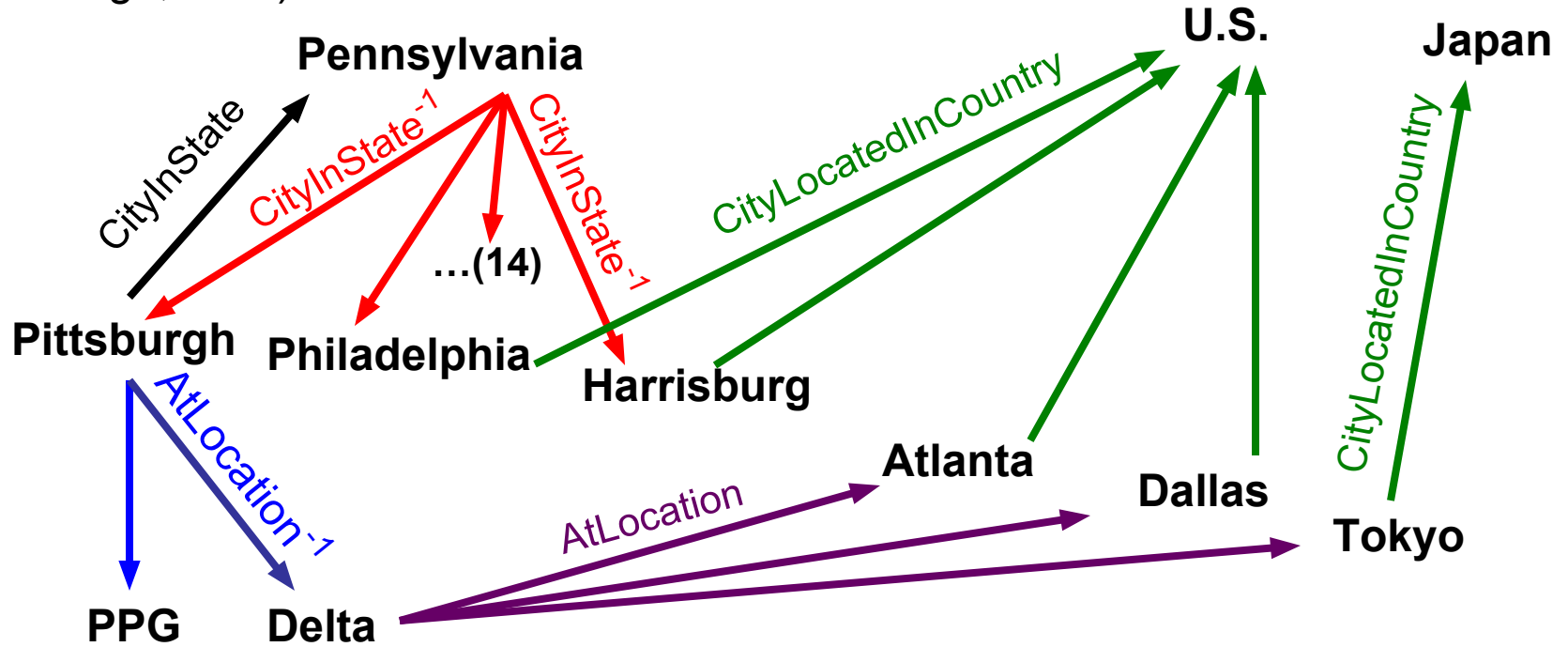
CityInState, **CityInState⁻¹**, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry

Feature Value

1.0

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al, EMNLP 2011]



Feature = Typed Path

CityInState, CityInState⁻¹, CityLocatedInCountry

AtLocation⁻¹, AtLocation, CityLocatedInCountry

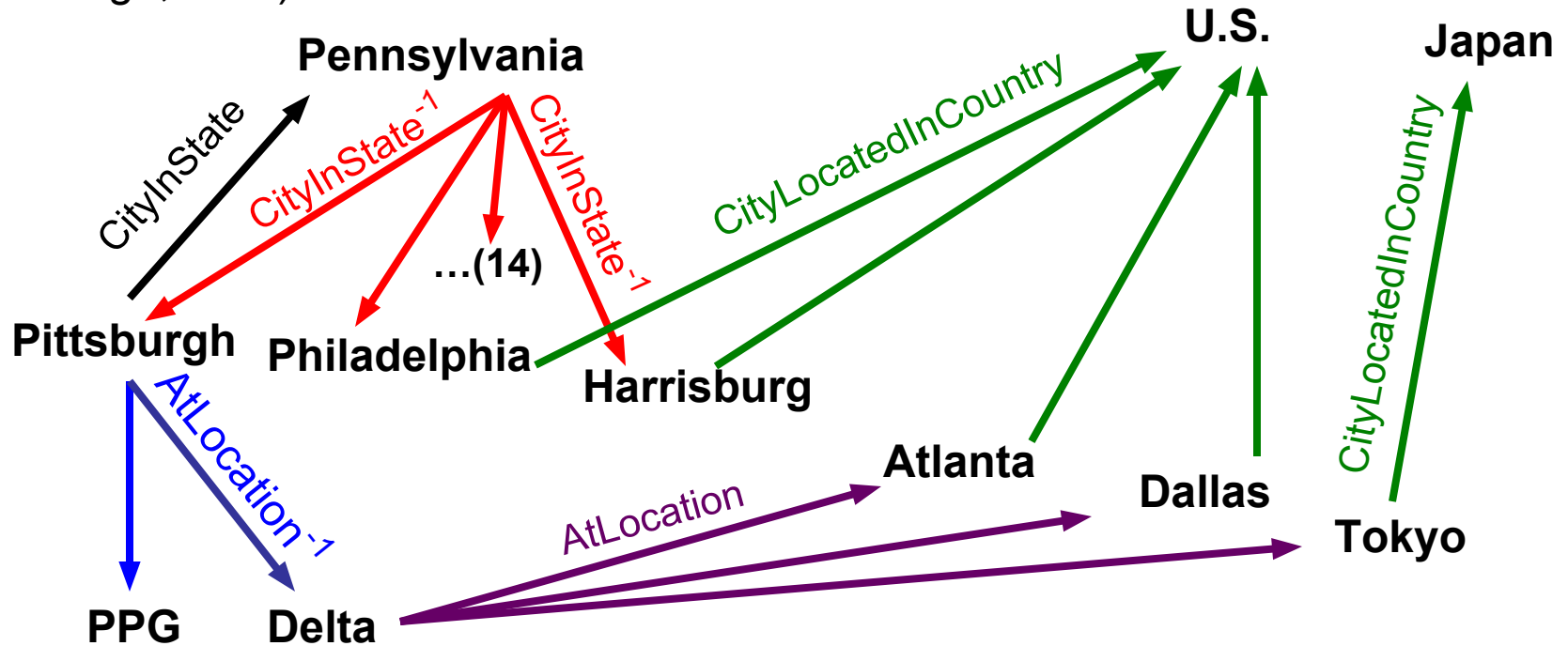
Feature Value

1.0

0.6

CityLocatedInCountry - Computing feature values for (Pittsburgh, USA)

[Lao et al,
EMNLP 2011]



Feature = Typed Path

CityInState, **CityInState⁻¹**, CityLocatedInCountry

AtLocation⁻¹, AtLocation, CityLocatedInCountry

...

Feature Value

1.0

0.6

...

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

PRA Feature Matrix

CityLocatedInCountry

(Pittsburgh, USA)
(Pittsburgh, Japan)
...
(Tokyo, USA)
(Tokyo, Japan)
...

CityInState, CityInState^{-1} , $\text{CityLocatedInCountry}$ AtLocation^{-1} , AtLocation , $\text{CityLocatedInCountry}$					
...	1.0	0.6	0.0	0.4	...
...	0.0	0.2	0.2	0.1	...
	...				
	...				
	...				
	...				

PRA Feature Matrix

CityLocatedInCountry

(Pittsburgh, USA)
(Pittsburgh, Japan)
...
(Tokyo, USA)
(Tokyo, Japan)
...

CityInState, CityInState^{-1} , $\text{CityLocatedInCountry}$ AtLocation^{-1} , AtLocation , $\text{CityLocatedInCountry}$					
(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

(Pittsburgh, USA)
(Pittsburgh, Japan)
...
(Tokyo, USA)
(Tokyo, Japan)
...

CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry
... ..

1.0	0.6	0.0	0.4	...
0.0	0.2	0.2	0.1	...
...				
...				
...				
...				

Large data sets?

$$O(n^2)$$

PRA Feature Matrix

CityLocatedInCountry

(Pittsburgh, USA)
(Pittsburgh, Japan)
...
(Tokyo, USA)
(Tokyo, Japan)
...

$O(n^2)$

CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry
... ..

$O(r^l)$

1.0	0.6	0.0	0.4	...
0.0	0.2	0.2	0.1	...
...				
...				
...				
...				

Large data sets?

PRA Feature Matrix

CityLocatedInCountry

(Pittsburgh, USA)
(Pittsburgh, Japan)
...
(Tokyo, USA)
(Tokyo, Japan)
...

$$O(n^2)$$

CityInState, CityInState⁻¹, CityLocatedInCountry
AtLocation⁻¹, AtLocation, CityLocatedInCountry
... ..

$$O(r^l)$$

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

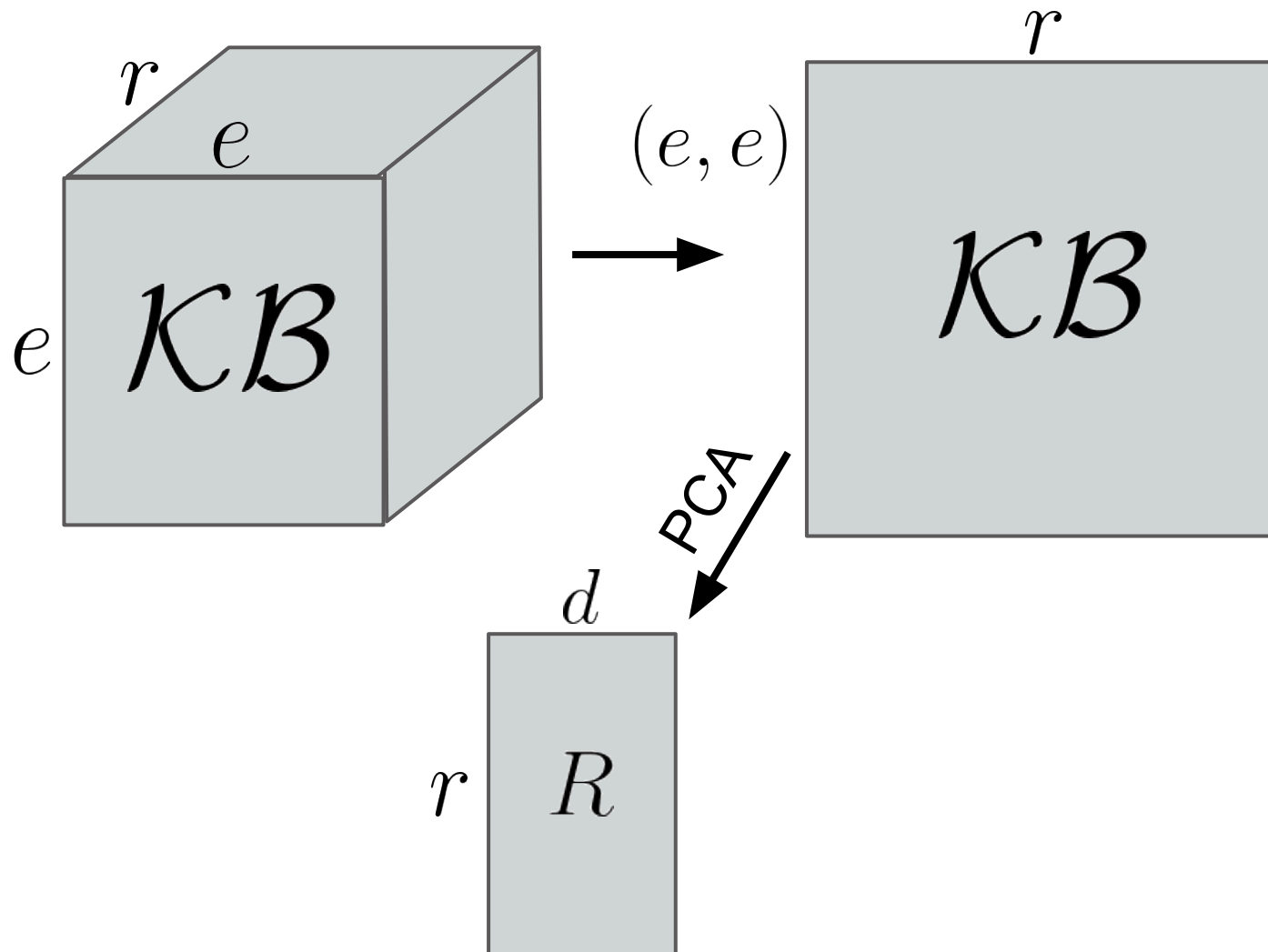
Inference over KB plus text

- 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”

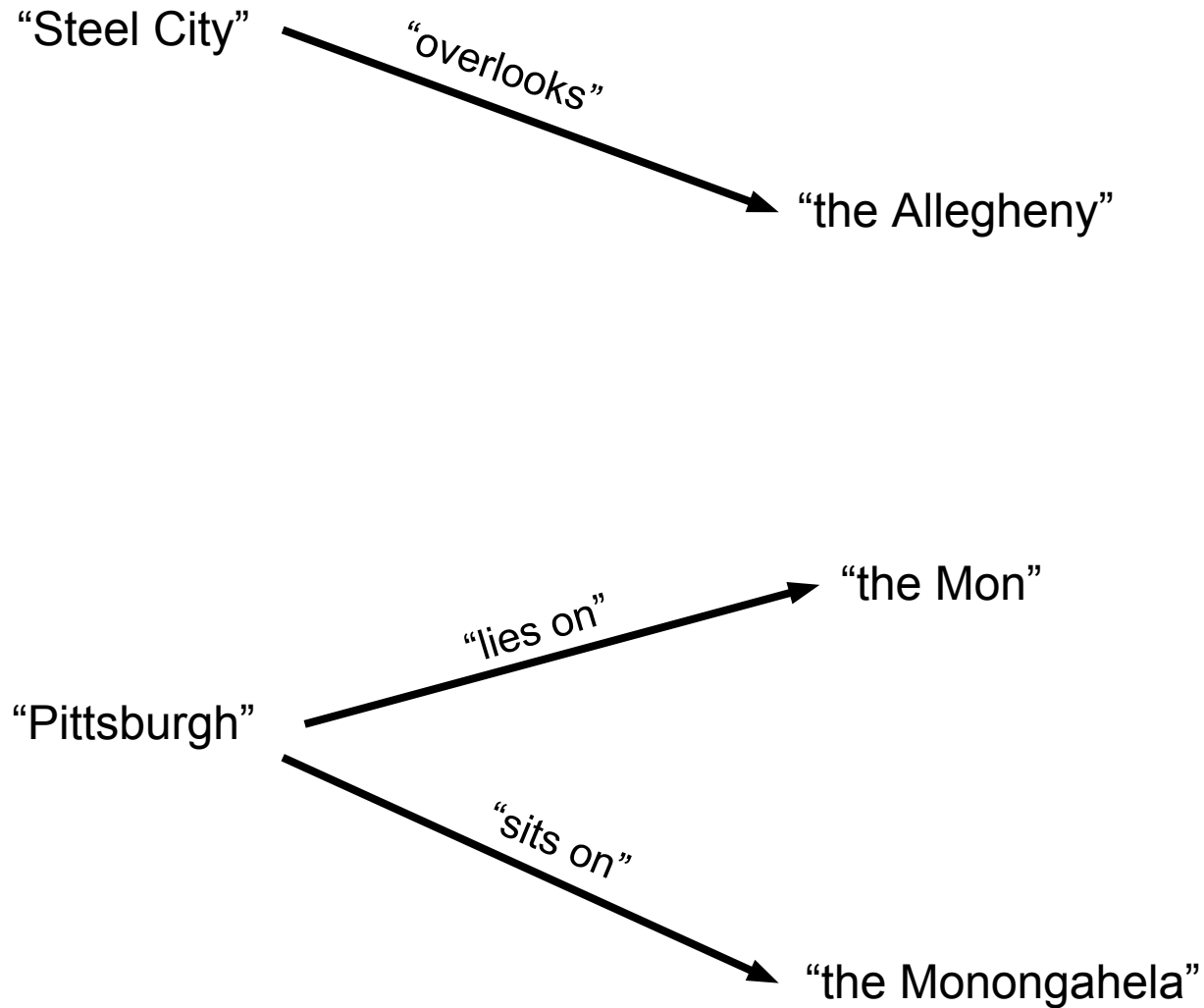
“Pittsburgh”

“the Monongahela”

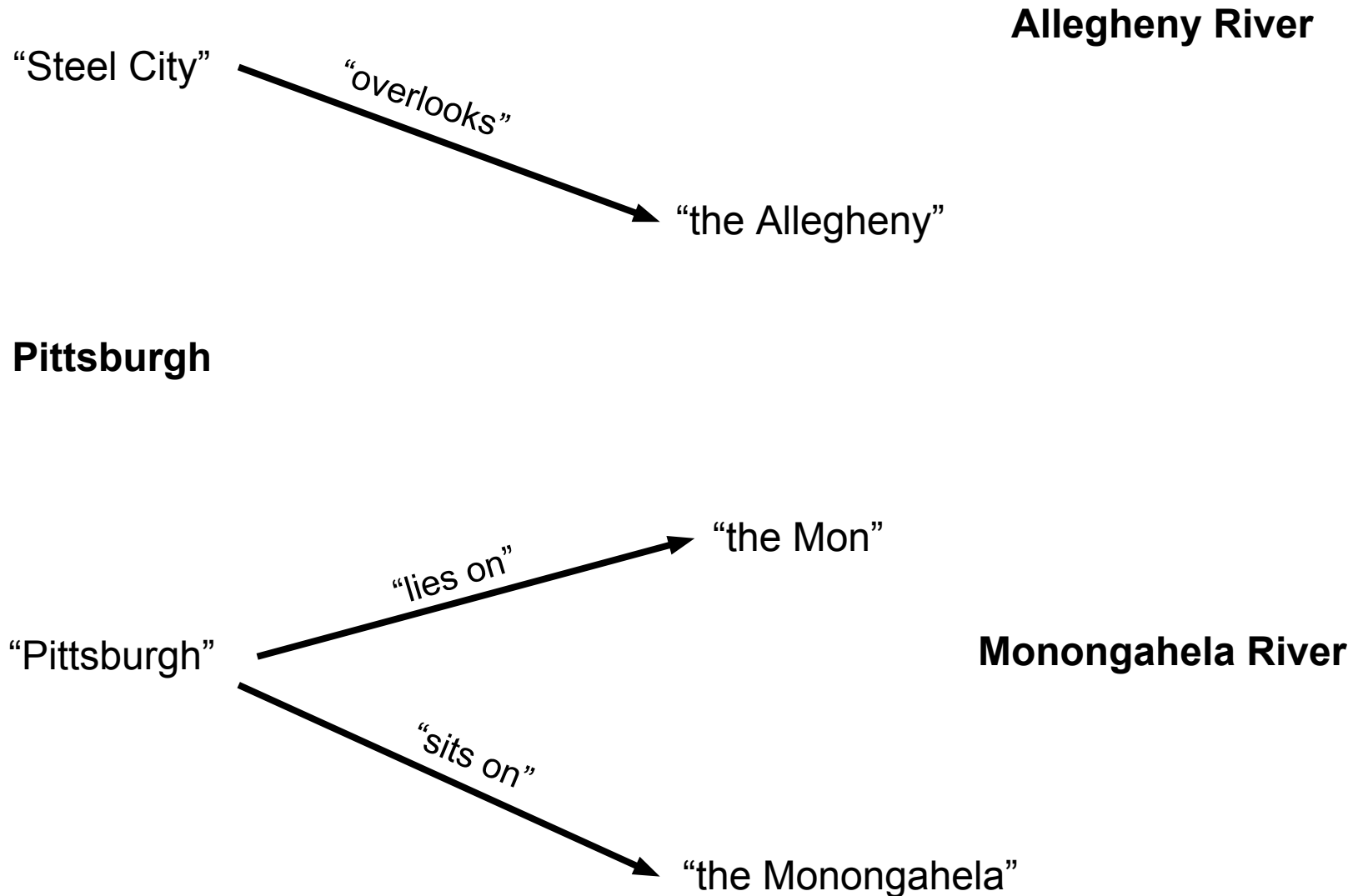
“Steel City overlooks the Allegheny”

“Pittsburgh lies on the Mon”

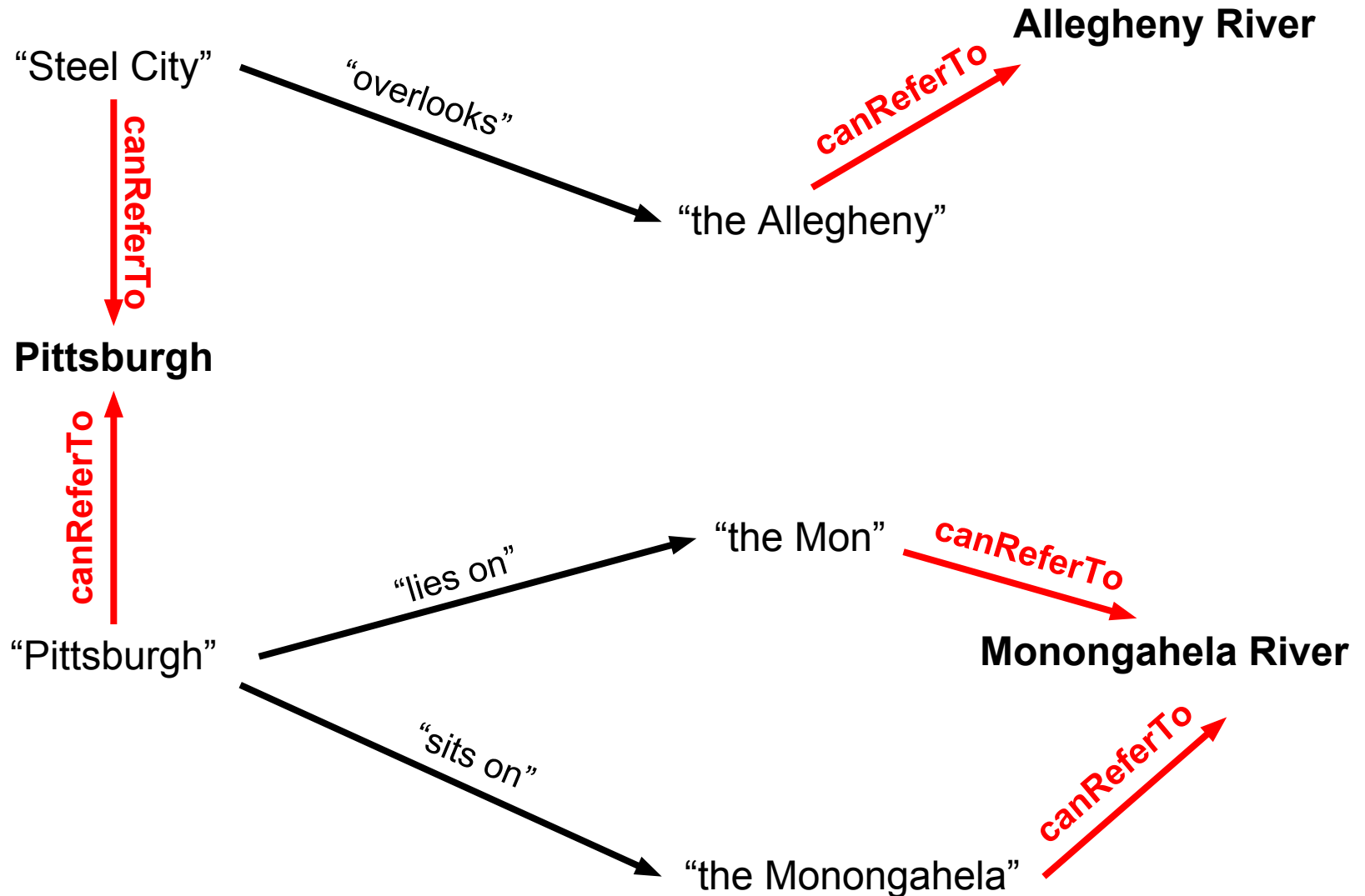
“Pittsburgh sits on the Monongahela”



“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

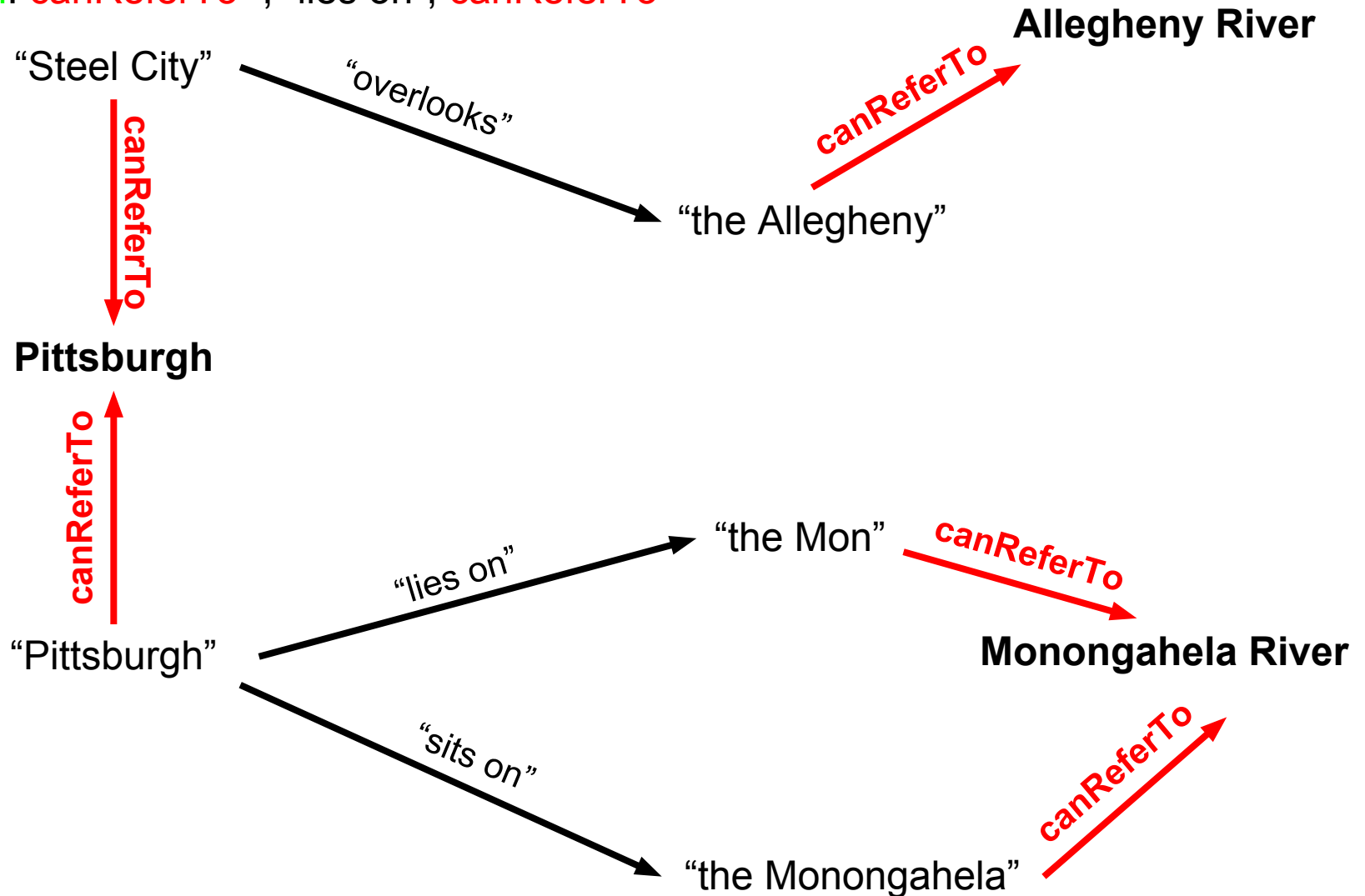
Relation: cityLiesOnRiver

Path: canReferTo⁻¹, “lies on”, canReferTo

KB + Text

Relation: **cityLiesOnRiver**

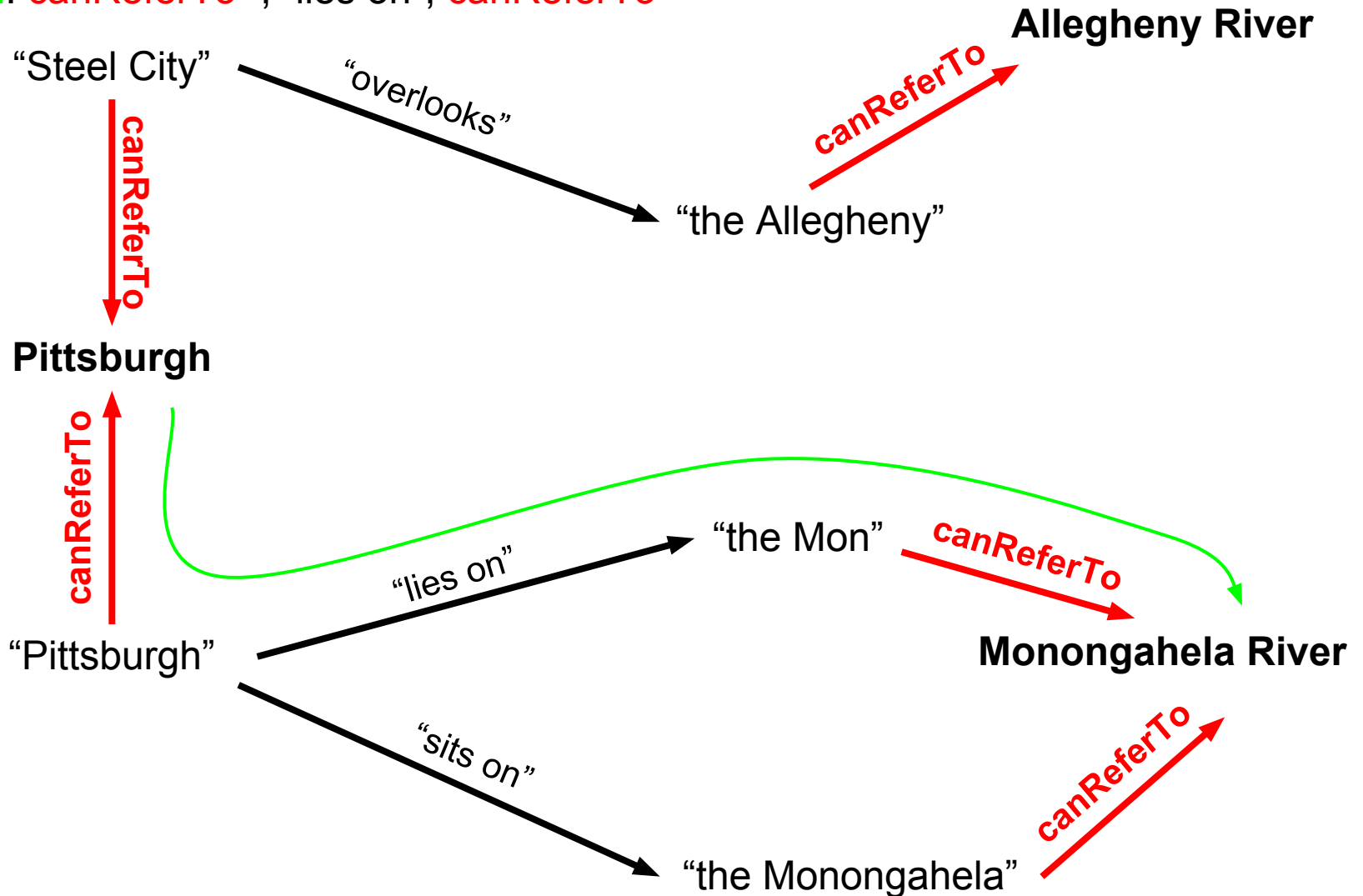
Path: **canReferTo**⁻¹, “lies on”, **canReferTo**



KB + Text

Relation: cityLiesOnRiver

Path: canReferTo⁻¹, "lies on", canReferTo



KB + Text

Relation: **cityLiesOnRiver**

Path: **canReferTo**⁻¹, “lies on”

“Steel City”

canReferTo

Pittsburgh

canReferTo

“Pittsburgh”

“overlooks”

“lies on”

“sits on”

“the Mon”

“the Monongahela”

canReferTo

Monongahela River

canReferTo

- 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

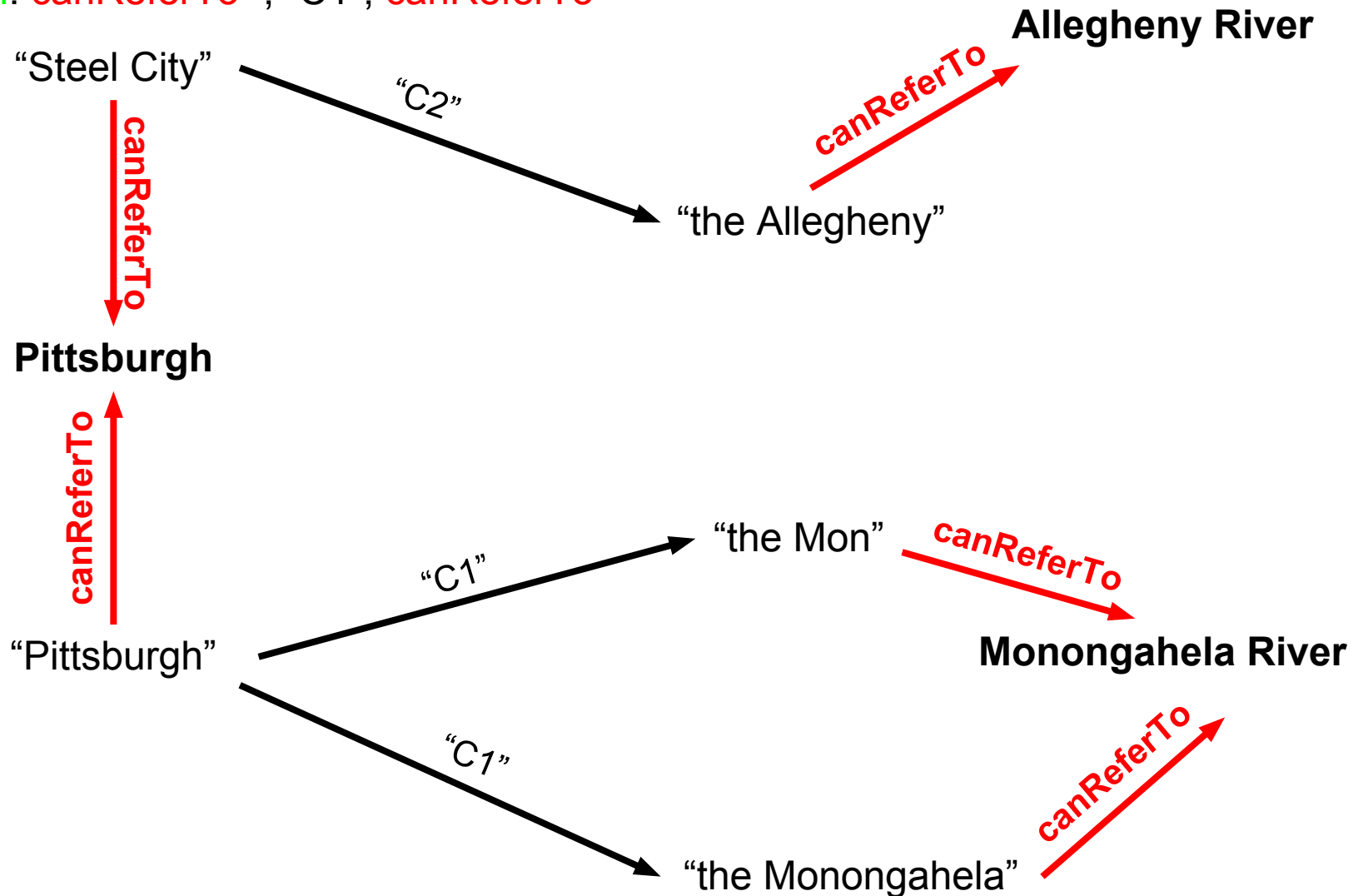
Relation: cityLiesOnRiver

Path: canReferTo⁻¹, "C1", canReferTo

KB + Clustered Text

Relation: cityLiesOnRiver

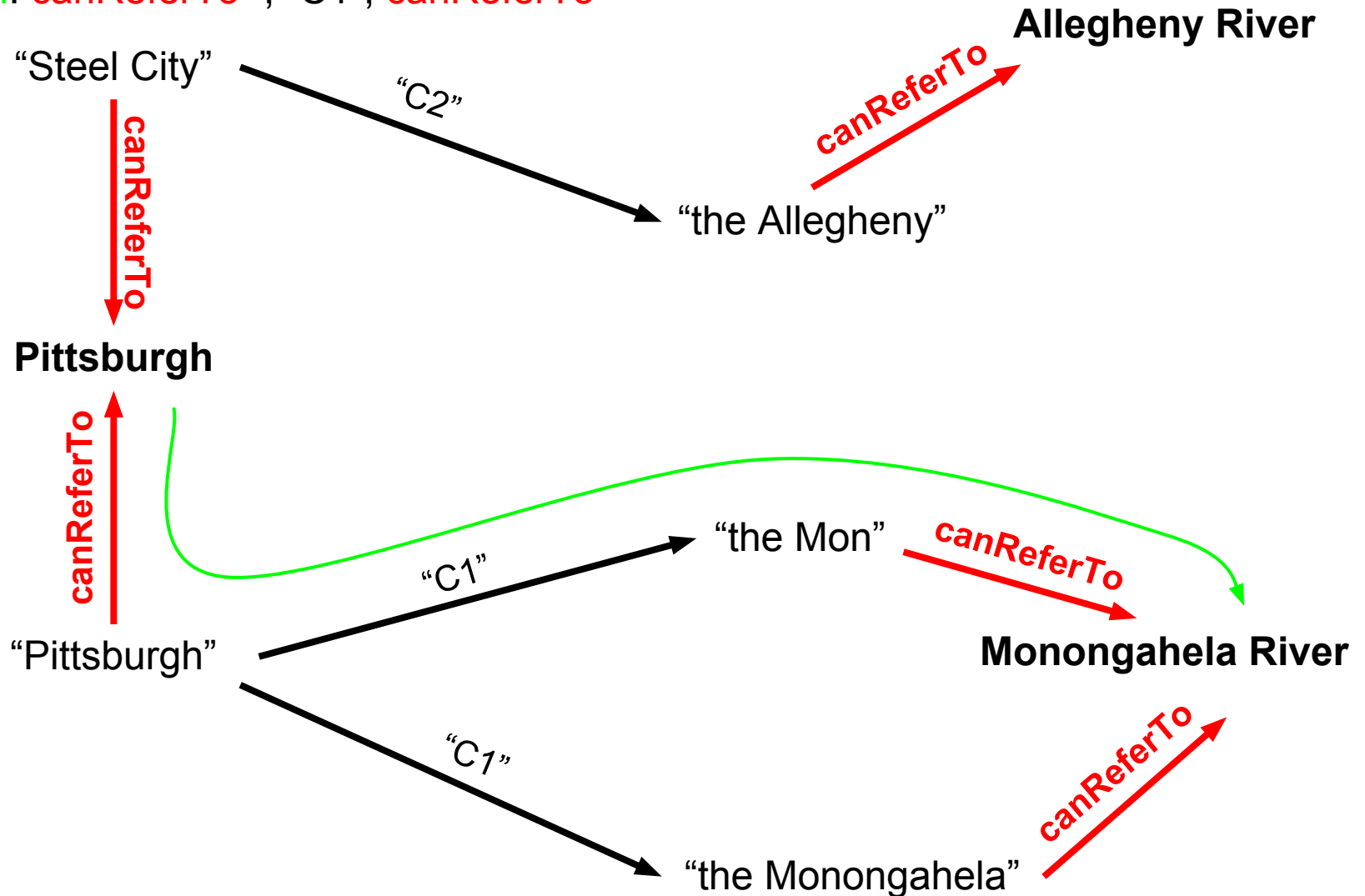
Path: canReferTo⁻¹, "C1", canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

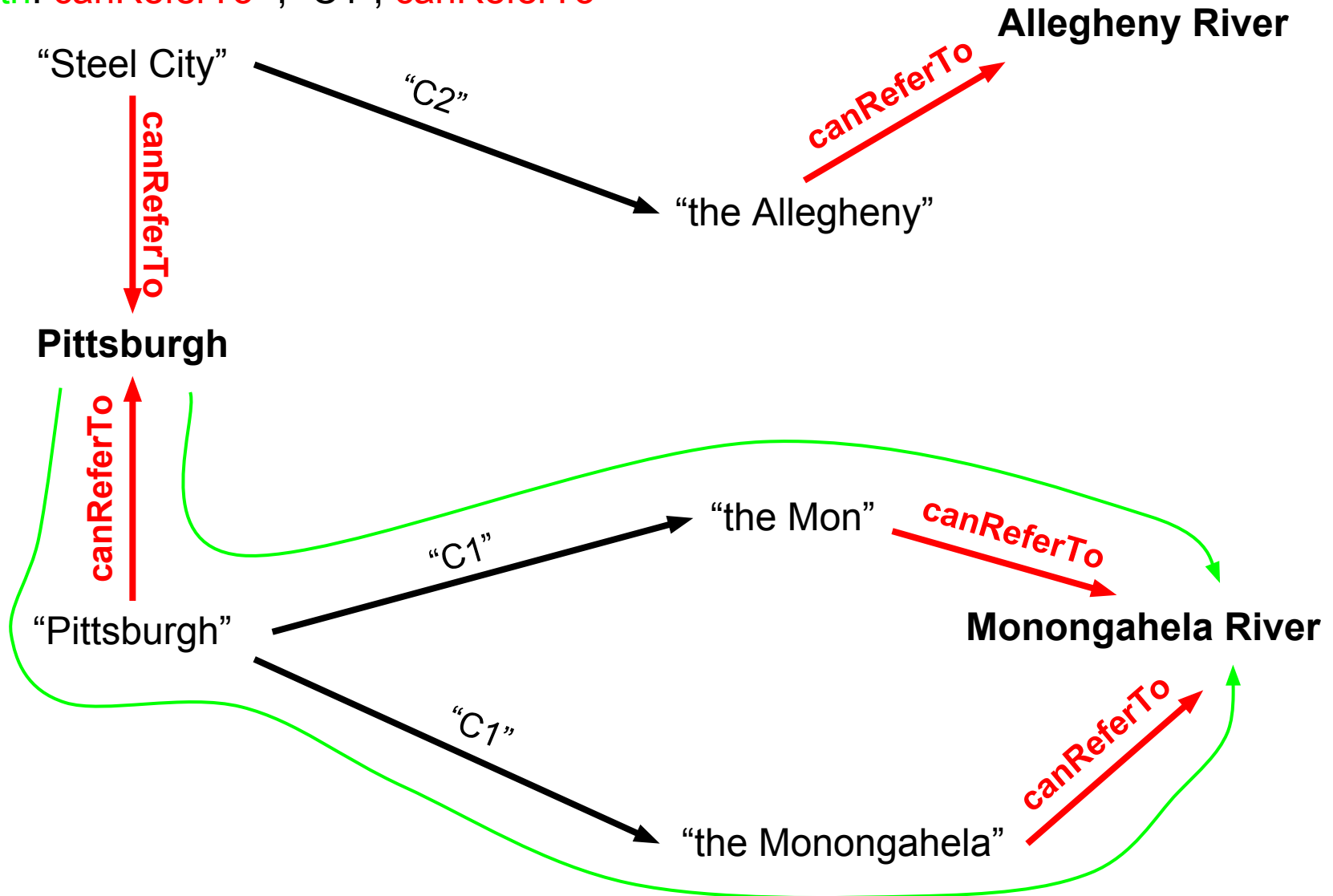
Path: canReferTo⁻¹, "C1", canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

Path: canReferTo⁻¹, "C1", canReferTo



KB + Clustered Text

Relation: cityLiesOnRiver

Path: canReferTo⁻¹, "C1", canReferTo

"Steel City"

canReferTo

Pittsburgh

canReferTo

"Pittsburgh"

"C1"

"the Mon"

canReferTo

Monongahela River

"C1"

"the Monongahela"

canReferTo

- 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

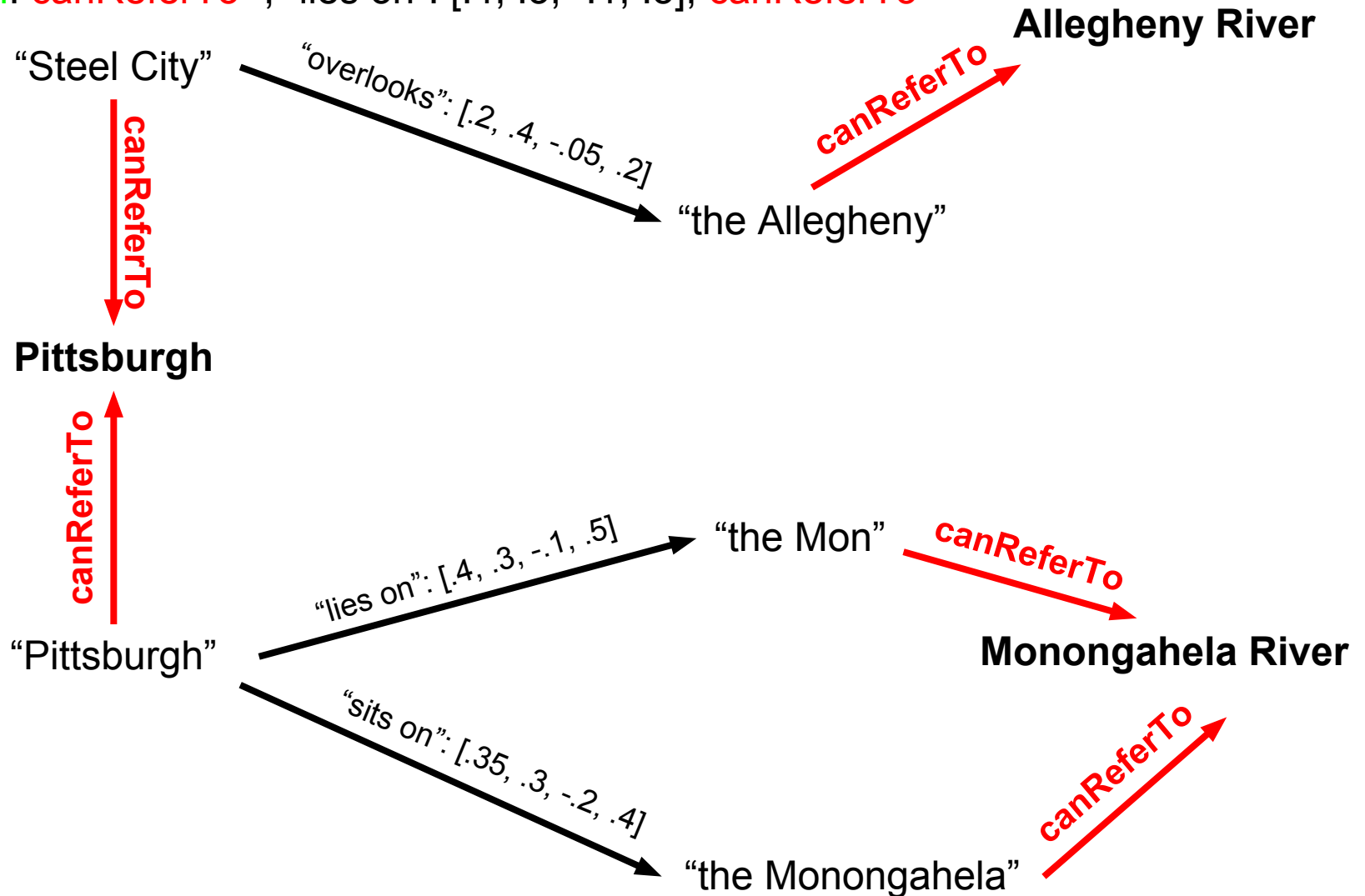
Relation: cityLiesOnRiver

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

KB + Text Vectors

Relation: **cityLiesOnRiver**

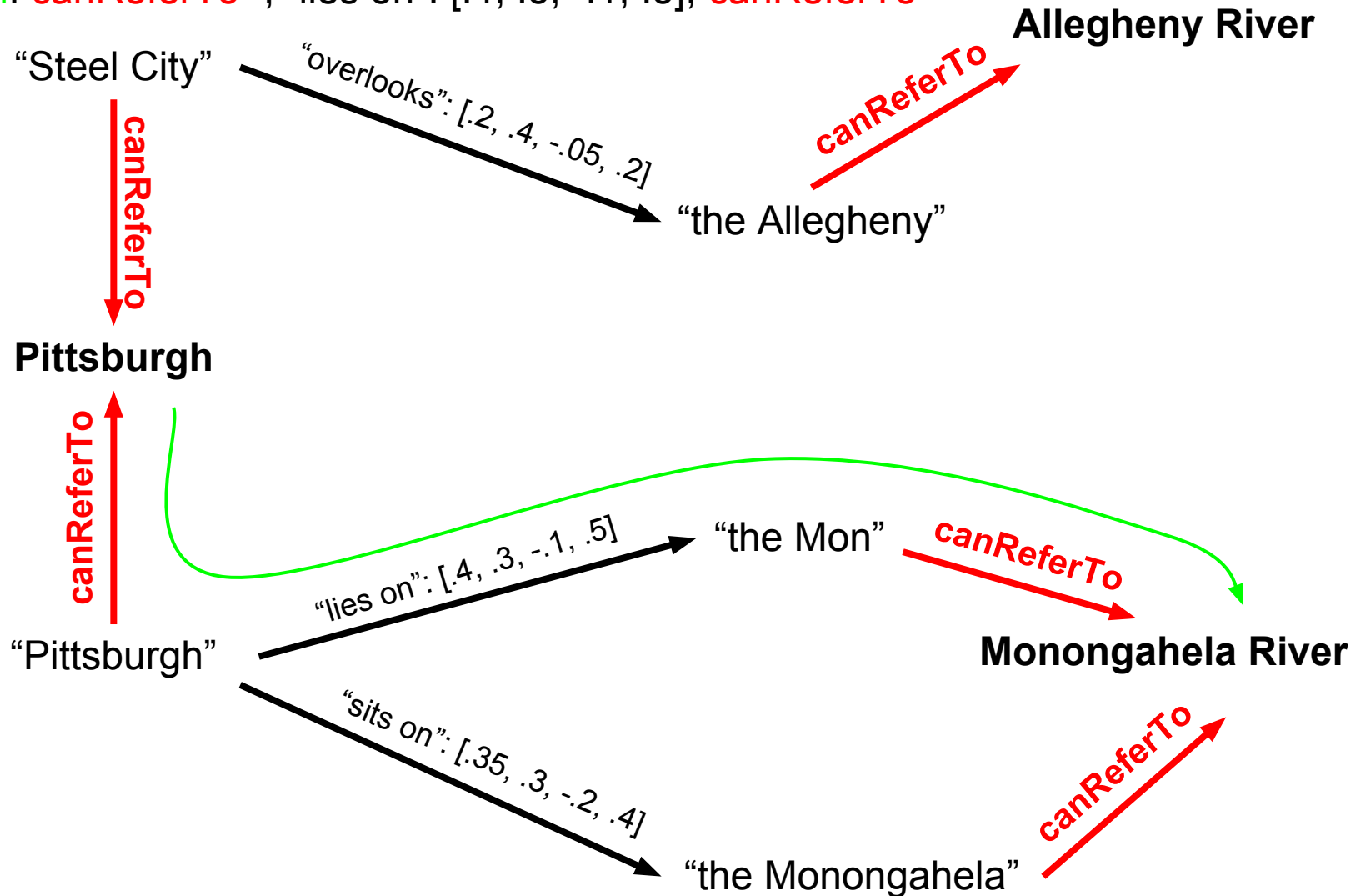
Path: **canReferTo**⁻¹, “lies on”: [.4, .3, -.1, .5], **canReferTo**



KB + Text Vectors

Relation: **cityLiesOnRiver**

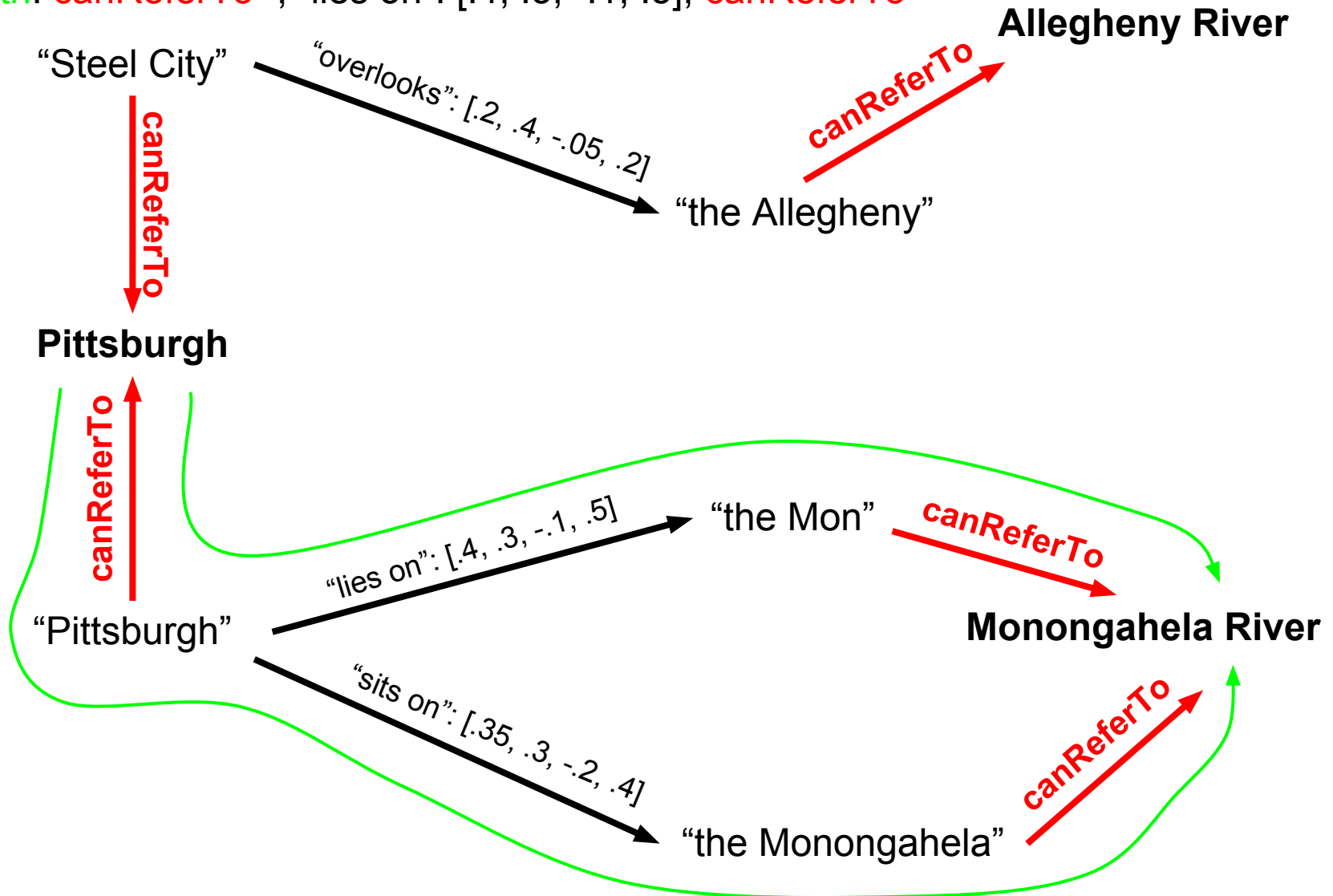
Path: **canReferTo**⁻¹, “lies on”: [.4, .3, -.1, .5], **canReferTo**



KB + Text Vectors

Relation: **cityLiesOnRiver**

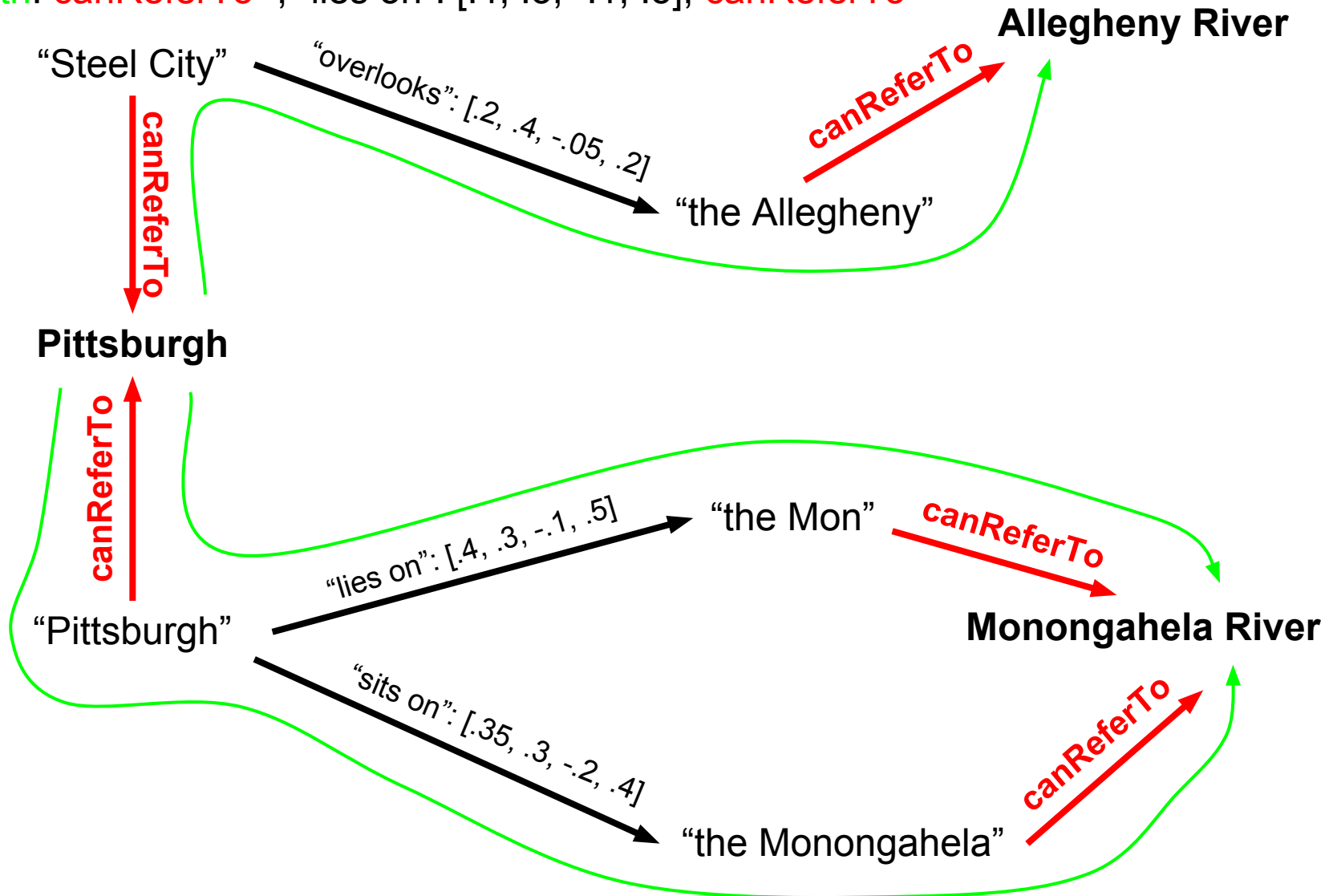
Path: **canReferTo**⁻¹, “lies on”: [.4, .3, -.1, .5], **canReferTo**



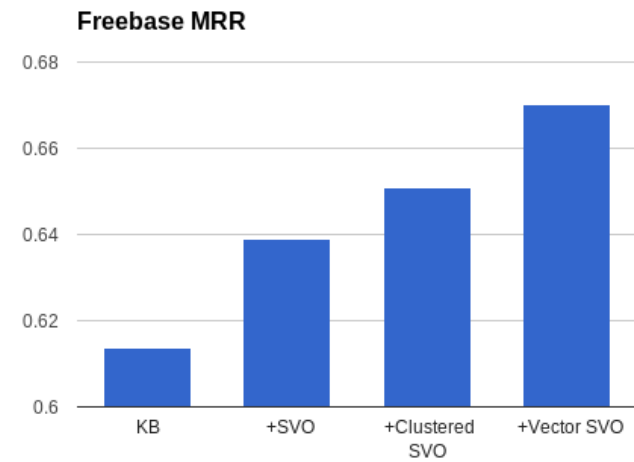
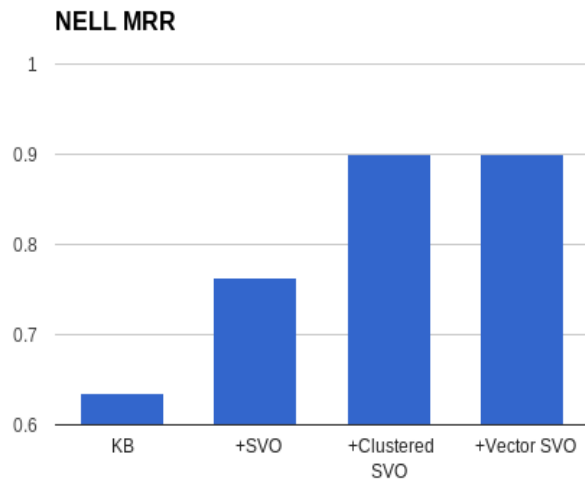
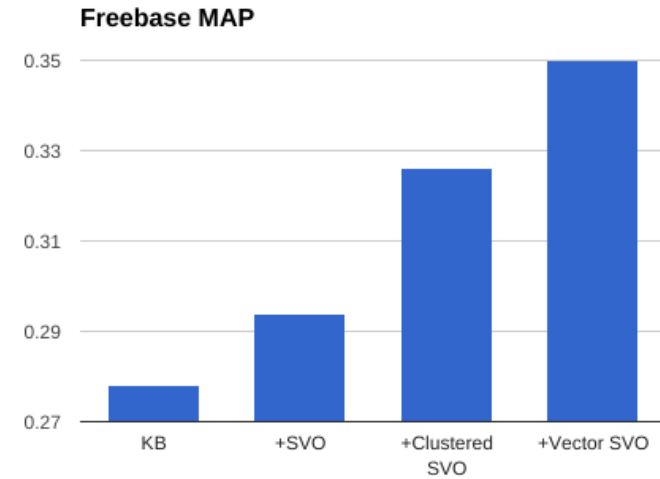
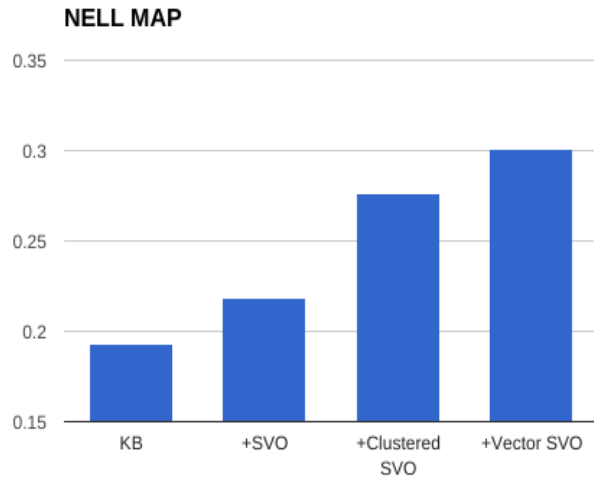
KB + Text Vectors

Relation: **cityLiesOnRiver**

Path: **canReferTo**⁻¹, “lies on”: [.4, .3, -.1, .5], **canReferTo**



Empirical Results



Questions?
