Mining Concepts from Code

using Community Detection in co-occurrence Graphs

Toni Mattis

Software Architecture Group (Robert Hirschfeld)

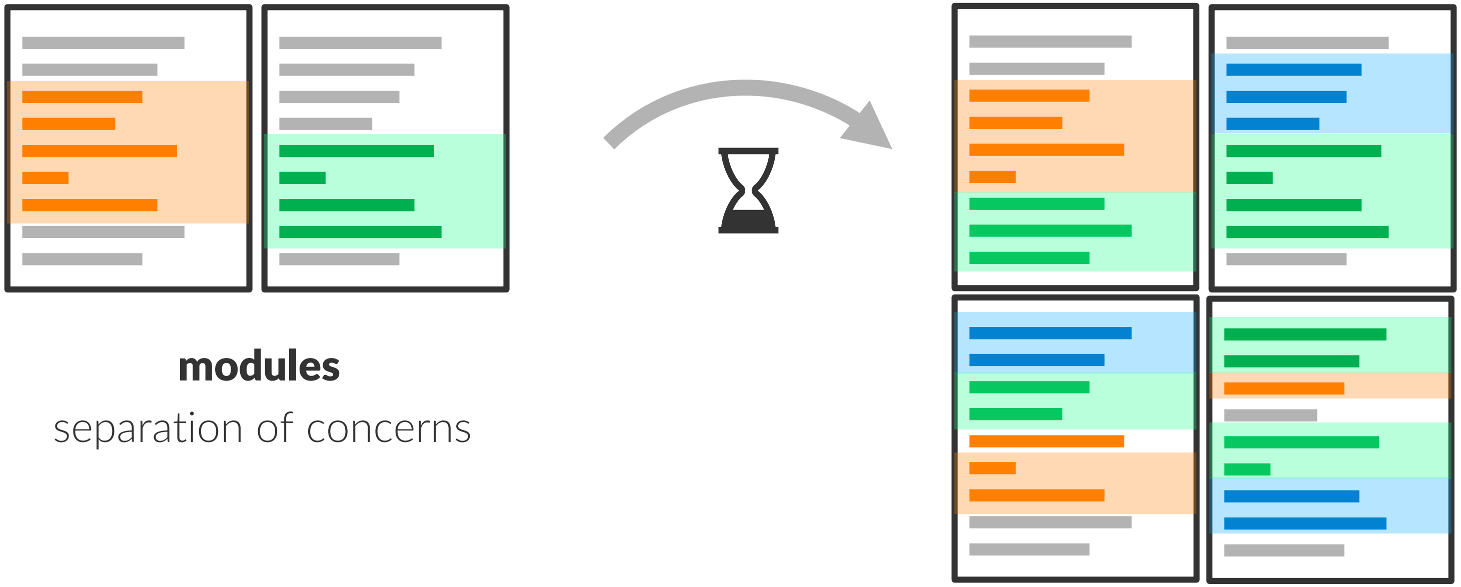
Hasso Plattner Institute, University of Potsdam, Germany

ACM SRC at ‹Programming› 11 Apr. 2018, Nice, France

Problem: Architectural Drift

With growing code bases…

- Concepts tend to scatter and entangle
- Programmers need more time to locate concepts

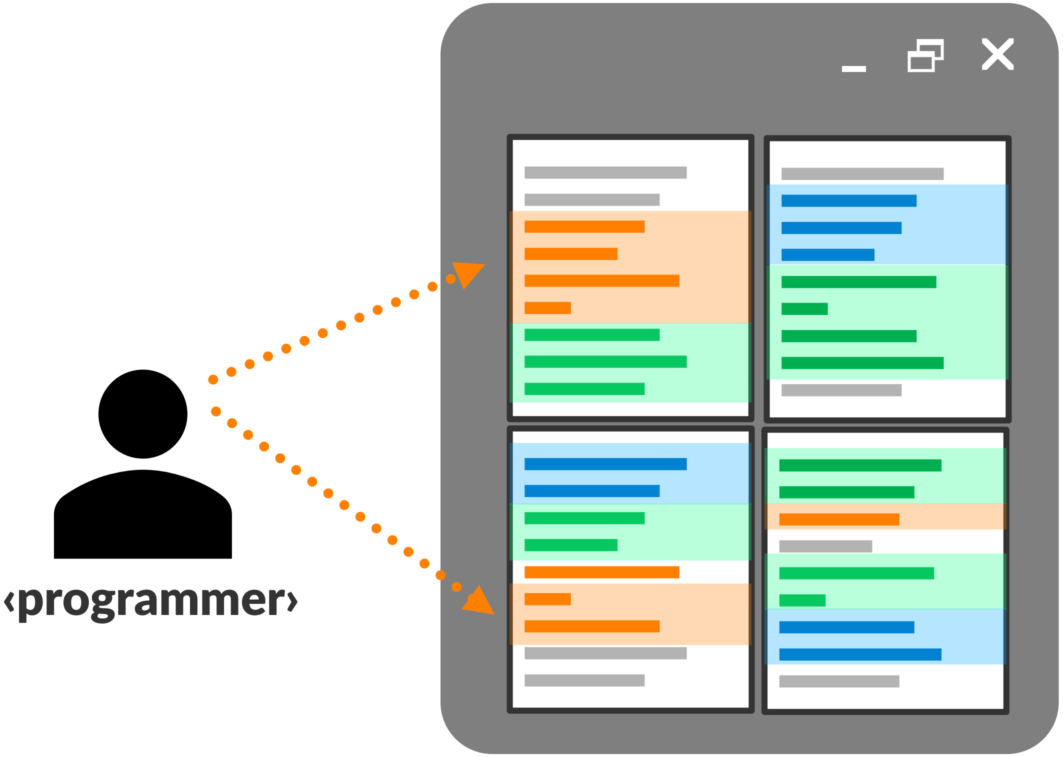


modules

separation of concerns

scattering / tangling

Goal: Concept Recovery and Location



Name-based Concept Model

concept locations

which concept a name belongs to

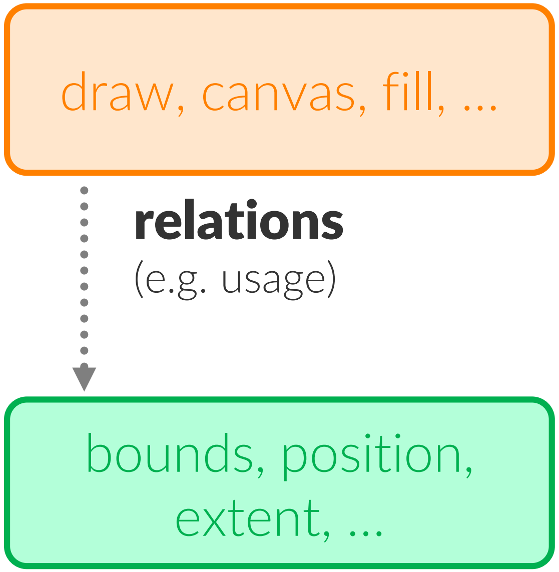
```
Canvas » draw: anObject
  ^ anObject drawOn: self


Morph » drawOn: aCanvas
  aCanvas fillRectangle: self bounds.


Morph » bounds: newBounds
  self position: newBounds topLeft;
  extent: newBounds extent.
```

concepts

prevalent names



draw, canvas, fill, ...

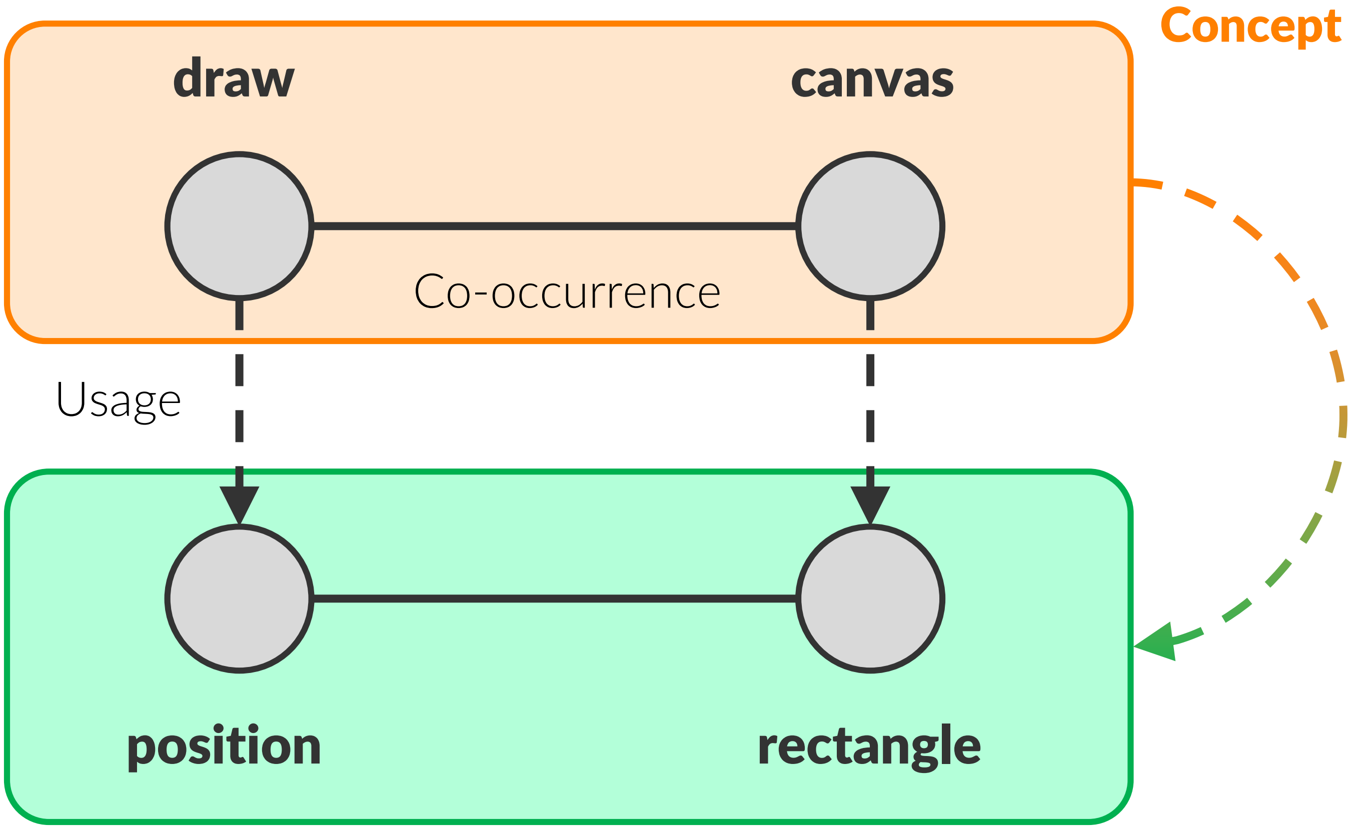
relations

(e.g. usage)

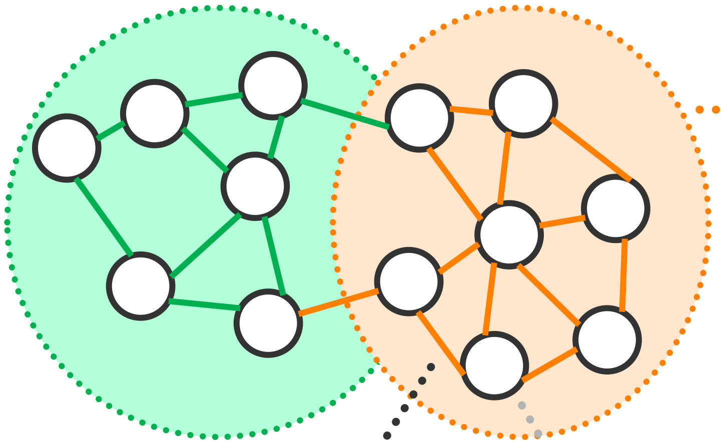
bounds, position, extent, ...

Graph-based Semantic Models

Nodes are **names**. **Edges** indicate they **co-occurr** in close proximity.



Random Graph Model + Topic Model

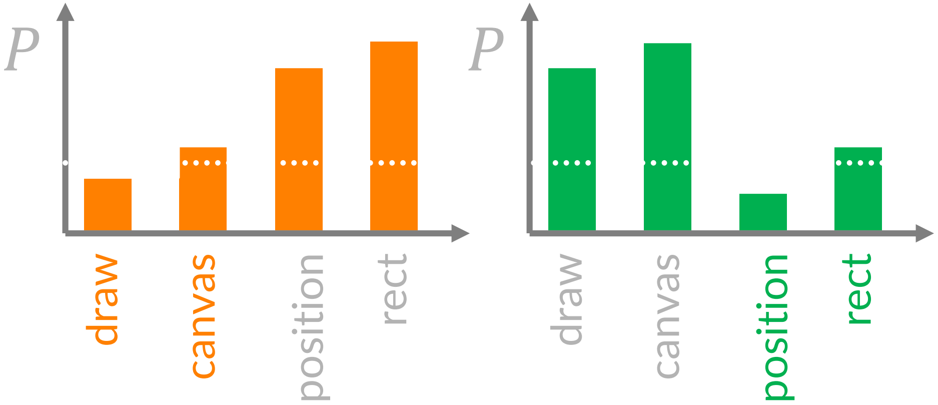


Concept c

distribution over names $P(n|c)$

global frequency $P(c)$

Name n

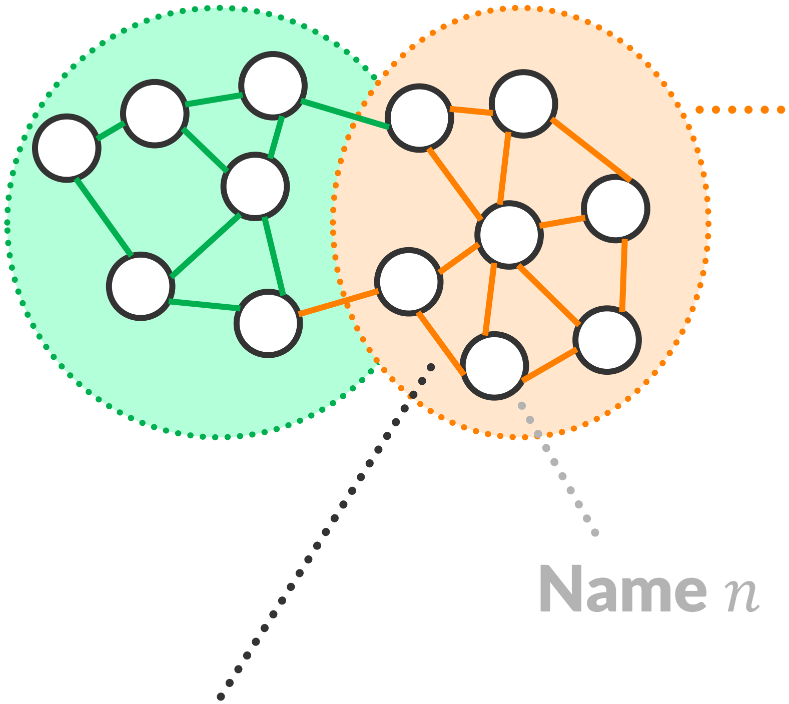


Edge e

$$P(e=(n_1,n_2)\mid c) \propto P(n_1|c)P(n_2|c)$$

$$P(G=(V,E)|\ \ldots) \propto \prod_{e\in E}\sum_{c} P(e|c) =$$

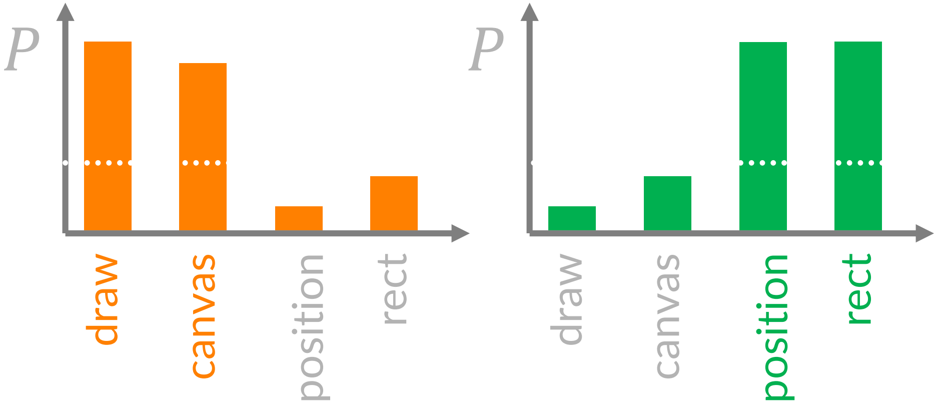
Random Graph Model + Topic Model



Concept c

distribution over names $P(n|c)$

global frequency $P(c)$



Name n

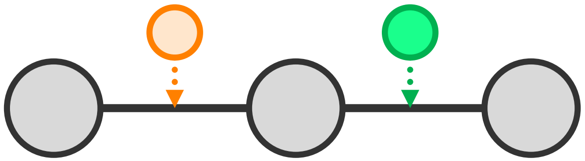
Edge e

$$P(e = (n_1, n_2) \mid c) \propto P(n_1|c)P(n_2|c)$$

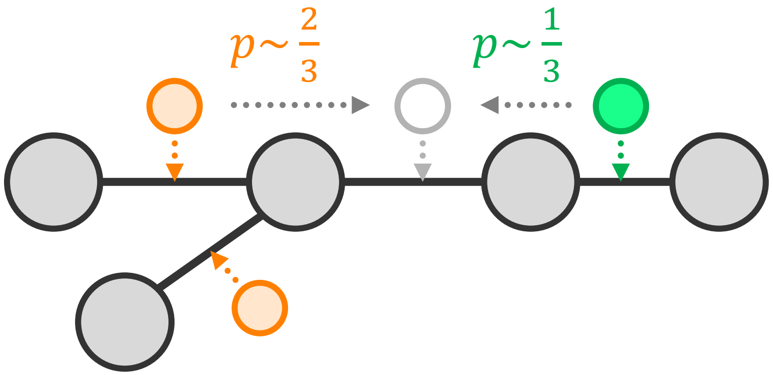
$$P(G = (V, E)|\ \dots) \propto \prod_{e \in E} \sum_{c} P(e|c) =$$

Determining $P(n|c)$ via Gibbs Sampling

Random assignment
of latent variables $\boldsymbol{c}$ to edges



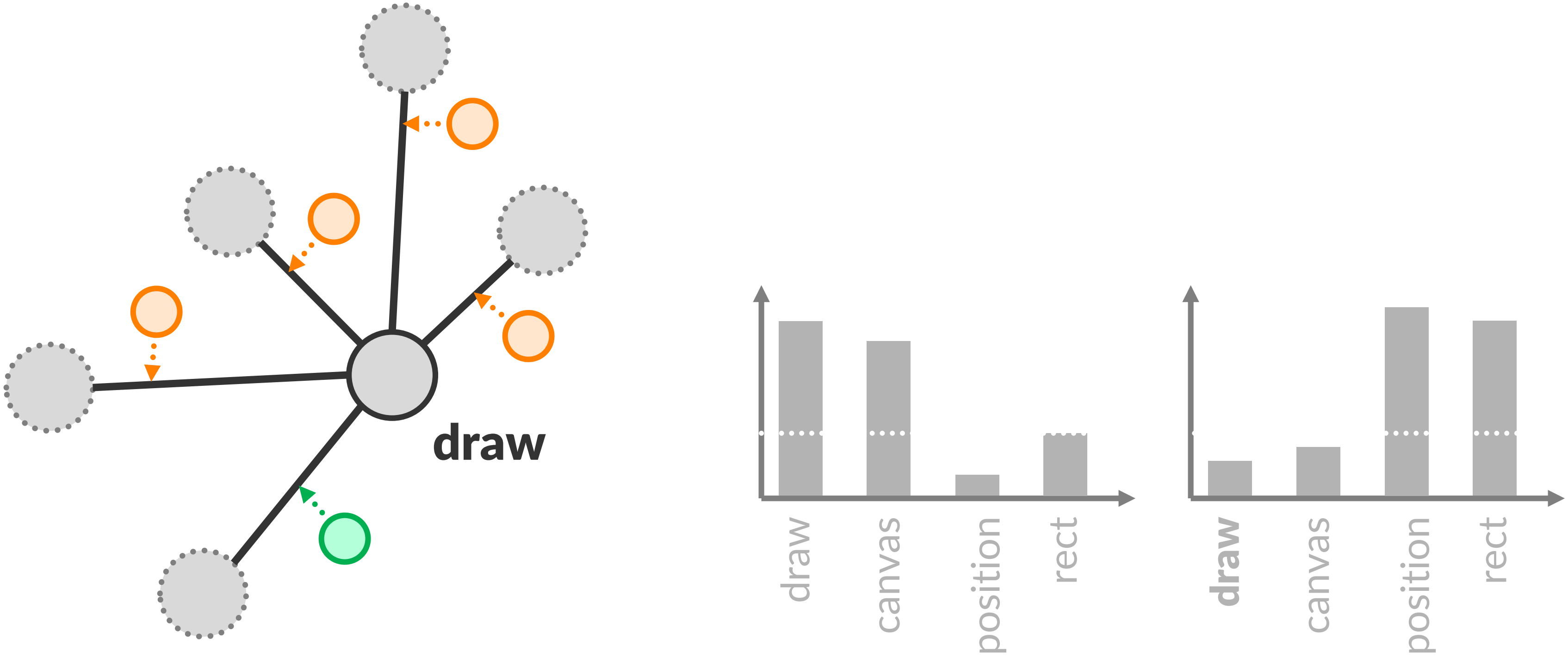
Iterative Re-sapling



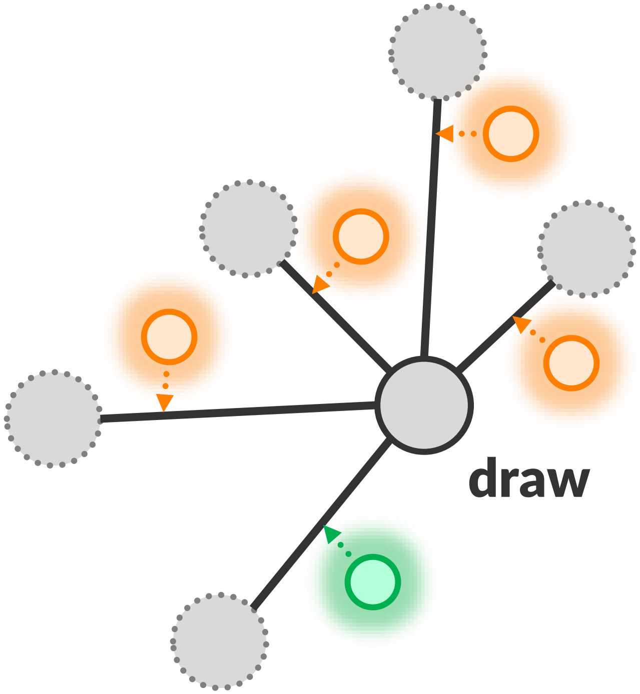
1. Decide on maximum number of concepts
2. Uniformly assign a concept to each edge
3. Re-assign each edge until near convergence

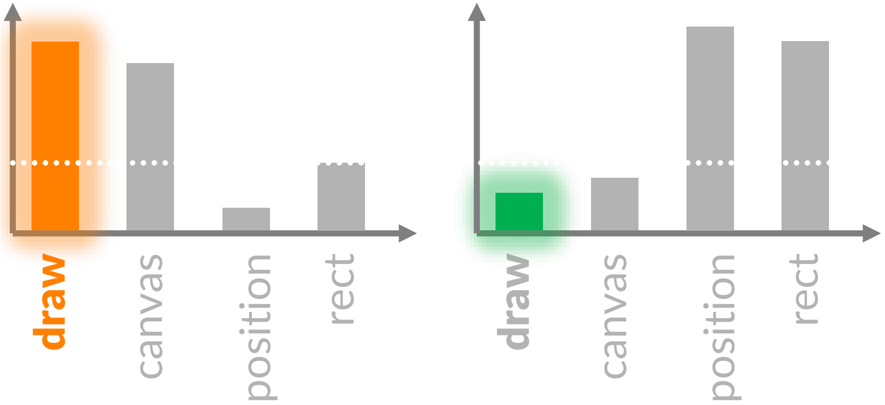
(clustering edges instead of nodes)

Random Graph Model + Topic Model



Random Graph Model + Topic Model



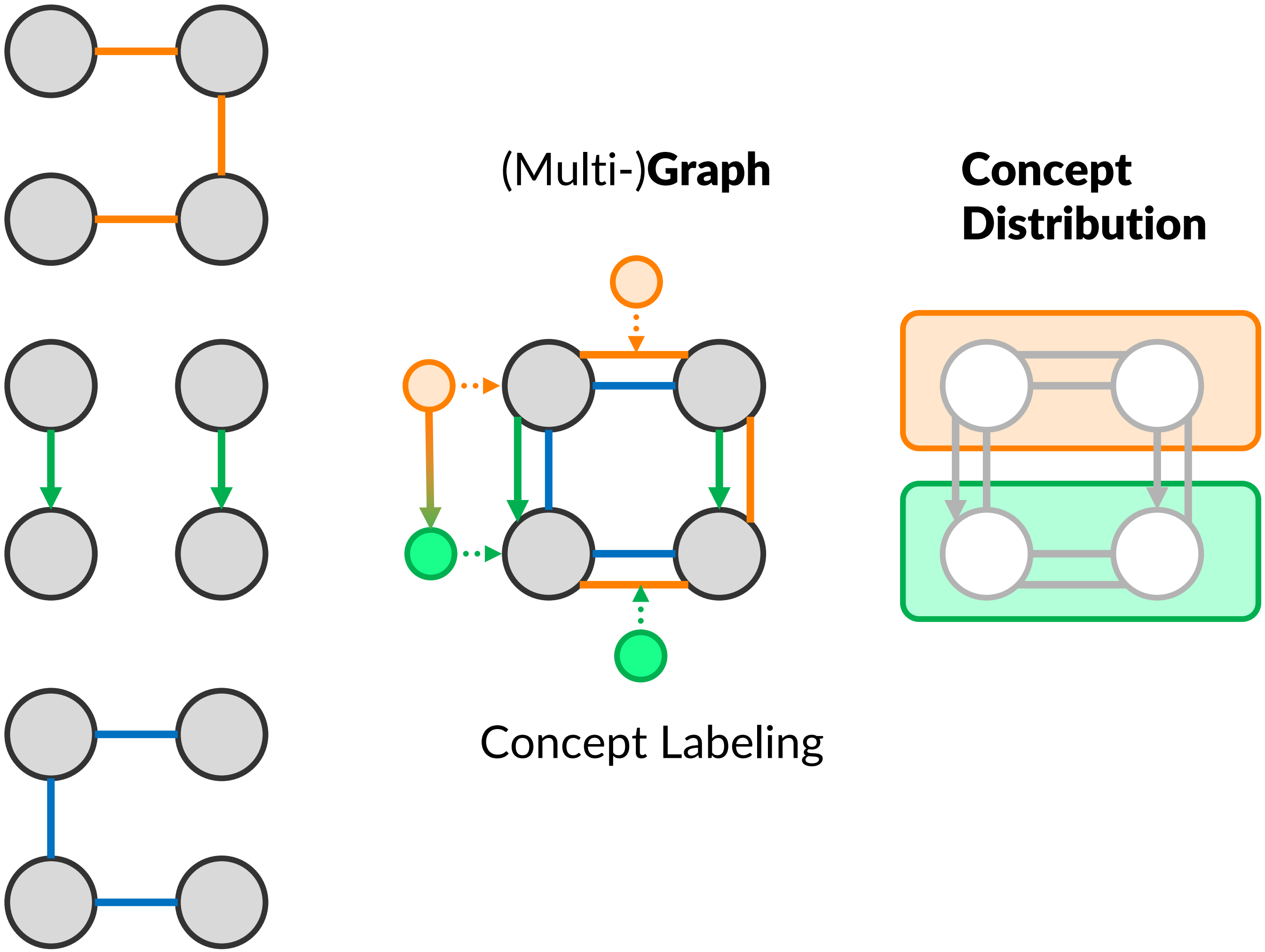


Multi-view Concepts

Co-located Names

Run-time Call Data

Git Commit (Diff)



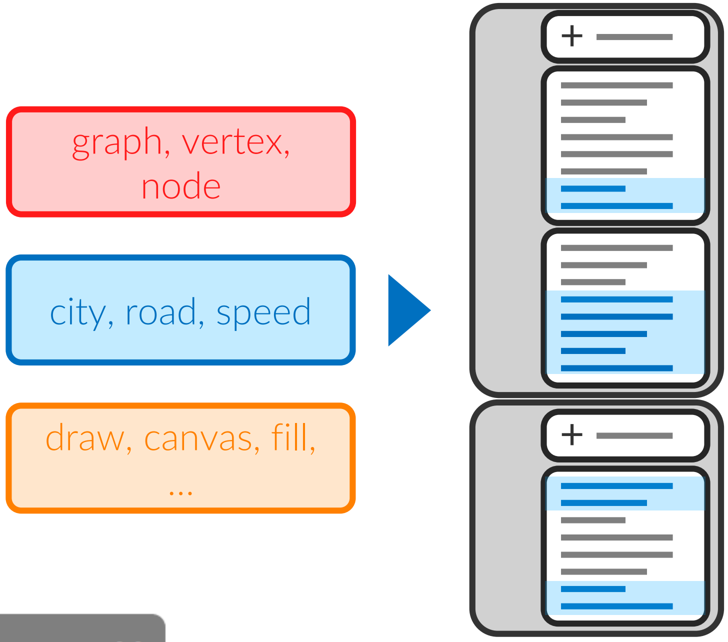
(Multi-)**Graph**

Concept Labeling

Concept Distribution

Future Work: Concept-aware Tooling

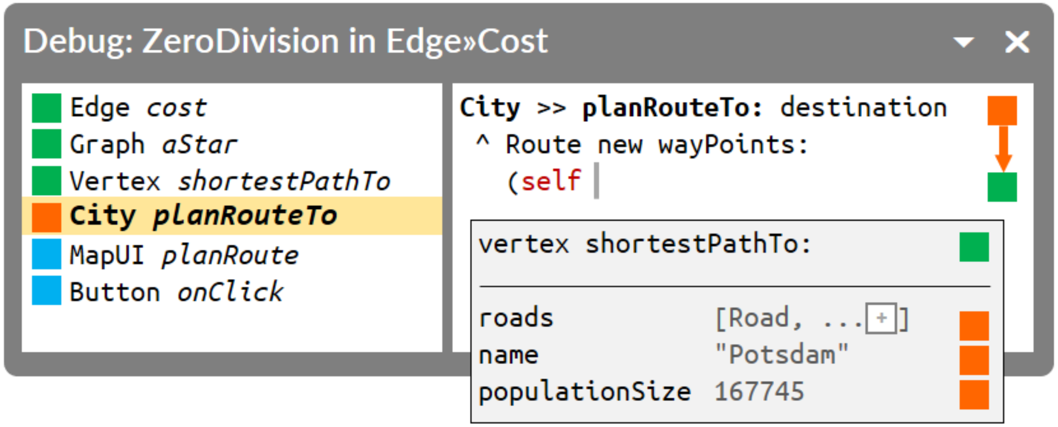
- **Highlight** concepts
- **Improve relevance** of information displayed during
  - search
  - code completion
  - debugging



graph, vertex, node

city, road, speed

draw, canvas, fill, …



Debug: ZeroDivision in Edge»Cost

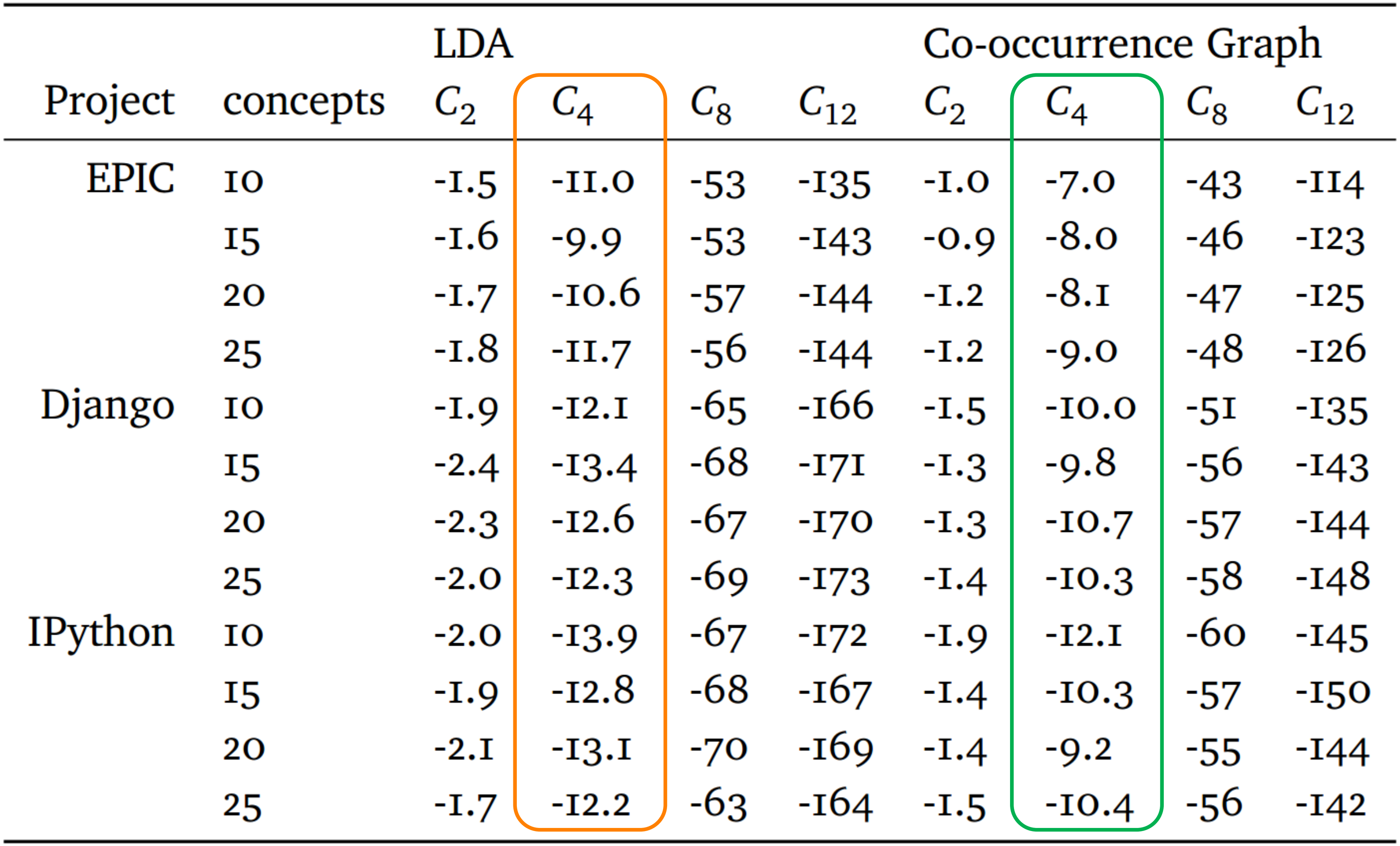
```
🟩 Edge cost
🟩 Graph aStar
🟩 Vertex shortestPathTo
🟧 City planRouteTo
🟦 MapUI planRoute
🟦 Button onClick
```

```
City >> planRouteTo: destination
  ^ Route new wayPoints:
    (self |
```

```
vertex shortestPathTo:

roads            [Road, ...[+]]
name             "Potsdam"
populationSize   167745
```

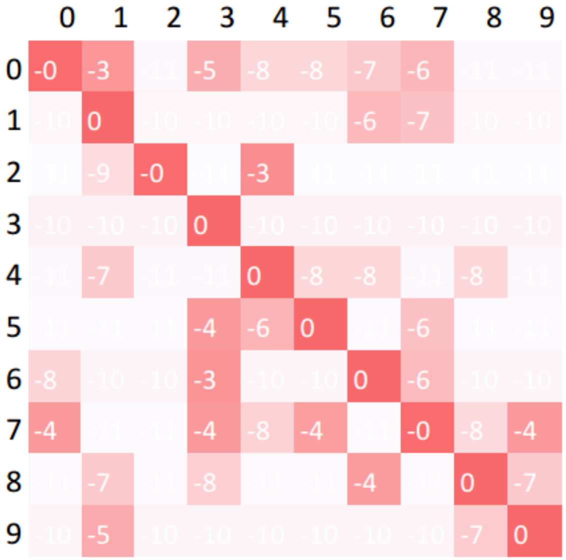
Concept Coherence (Mimno et al.)



		LDA				Co-occurrence Graph			
Project	concepts	C_2	C_4	C_8	C_{12}	C_2	C_4	C_8	C_{12}
EPIC	10	-1.5	-11.0	-53	-135	-1.0	-7.0	-43	-114
	15	-1.6	-9.9	-53	-143	-0.9	-8.0	-46	-123
	20	-1.7	-10.6	-57	-144	-1.2	-8.1	-47	-125
	25	-1.8	-11.7	-56	-144	-1.2	-9.0	-48	-126
Django	10	-1.9	-12.1	-65	-166	-1.5	-10.0	-51	-135
	15	-2.4	-13.4	-68	-171	-1.3	-9.8	-56	-143
	20	-2.3	-12.6	-67	-170	-1.3	-10.7	-57	-144
	25	-2.0	-12.3	-69	-173	-1.4	-10.3	-58	-148
IPython	10	-2.0	-13.9	-67	-172	-1.9	-12.1	-60	-145
	15	-1.9	-12.8	-68	-167	-1.4	-10.3	-57	-150
	20	-2.1	-13.1	-70	-169	-1.4	-9.2	-55	-144
	25	-1.7	-12.2	-63	-164	-1.5	-10.4	-56	-142

Table 4 Concepts inferred from the *EPIC* digital simulator

	Names	Comment (Interpretation)
0	*event if true mouse hand*	A cross-cutting concept handling mouse interaction
1	*morph layout panel create box*	The editor window
2	*canvas draw box center color*	Drawing circuitry
3	*is simulation if event not*	Event-driven simulator
4	*color rectangle string fill at*	Drawing shapes and text
5	*wire point anchor points bundle*	Wires, bundles of wires, and their connections
6	*input output values first with*	Expectations encoded in unit tests
7	*components panel component command all*	The panel containing pre-defined components
8	*xml circuit element named as*	The (de)serializer
9	*file name stream as named*	File reading/writing

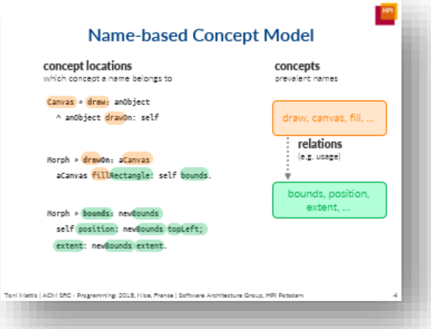


	0	1	2	3	4	5	6	7	8	9
0	-0	-3	-11	-5	-8	-8	-7	-6	-11	-11
1	-10	0	-10	-10	-10	-10	-6	-7	-10	-10
2	-11	-9	-0	-11	-3	-11	-11	-11	-11	-11
3	-10	-10	-10	0	-10	-10	-10	-10	-10	-10
4	-11	-7	-11	-11	0	-8	-8	-11	-8	-11
5	-11	-11	-11	-4	-6	0	-11	-6	-11	-11
6	-8	-10	-10	-3	-10	-10	0	-6	-10	-10
7	-4	-11	-11	-4	-8	-4	-11	-0	-8	-4
8	-11	-7	-11	-8	-11	-11	-4	-11	0	-7
9	-10	-5	-10	-10	-10	-10	-10	-10	-7	0

Figure 4 Abstract concepts (left) and how likely they relate to implementation-specific concepts (top). Values are logarithmically scaled.

Summary

- » Graph-based concept modeling is a **framework** based on a co-occurrence relation over names
- » **Future work:**
  extend tools to exploit conceptual information
- » By giving programmers **feedback** how well their modules align with concepts, they can **counteract** architectural drift



Name-based Concept Model

concept locations
which concept a name belongs to

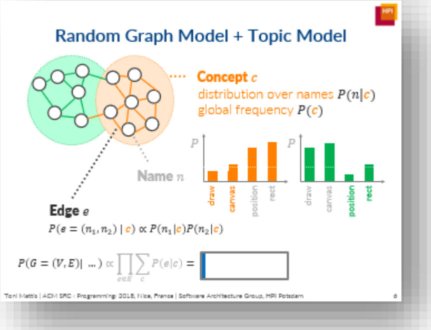
```
Canvas » draw: anObject
  ^ anObject drawOn: self

Morph » drawOn: aCanvas
  aCanvas fillRectangle: self bounds.

Morph » bounds: newBounds
  self position: newBounds topLeft;
  extent: newBounds extent.
```

concepts
prevalent names

Toni Mattis | ACM SRC ‹ Programming› 2018, Nice, France | Software Architecture Group, HPI Potsdam 4

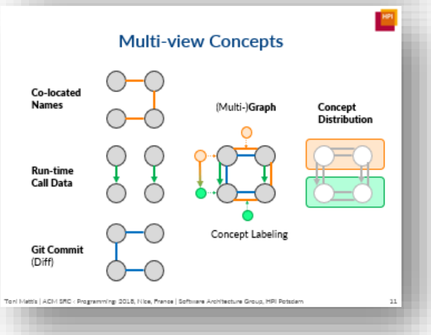


Random Graph Model + Topic Model

$P(e = (n_1, n_2) \mid c) \propto P(n_1|c)P(n_2|c)$

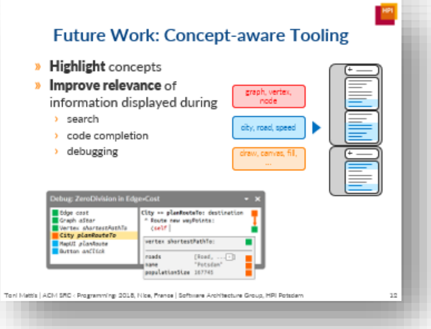
$P(G = (V,E)| \dots) \propto \prod_{e \in E} \sum_{c} P(e|c) =$

Toni Mattis | ACM SRC ‹ Programming› 2018, Nice, France | Software Architecture Group, HPI Potsdam 6



Multi-view Concepts

Toni Mattis | ACM SRC ‹ Programming› 2018, Nice, France | Software Architecture Group, HPI Potsdam 11



Future Work: Concept-aware Tooling

- » **Highlight** concepts
- » **Improve relevance** of information displayed during
  - › search
  - › code completion
  - › debugging

```
Edge cost
Graph aStar
Vertex shortestPathTo
City planRouteTo
MapUI planRoute
Button onClick
```

```
City » planRouteTo: destination
  ^ Route new wayPoints:
    (self |
```

```
vertex shortestPathTo:
roads            {Road, ...}
name             'Potsdam'
populationSize   167745
```

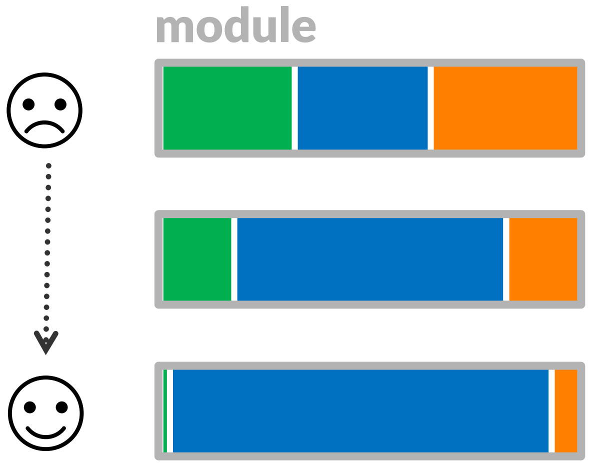
Toni Mattis | ACM SRC ‹ Programming› 2018, Nice, France | Software Architecture Group, HPI Potsdam 12

Backup Slides

A Perspective on Modularity

module entropy:

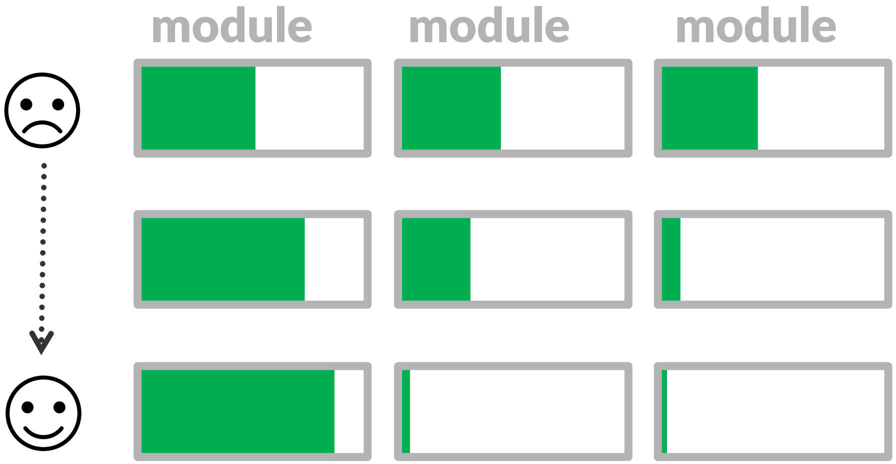
tangling



module

concept entropy:

scattering



module module module

$$H(m) = -\sum_{c} p(c|m) \log_2 p(c|m)$$

$$H(c) = -\sum_{m} p(m|c) \log_2 p(m|c)$$

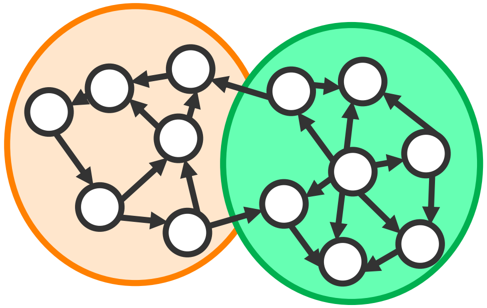
…high values indicate need for refactoring or cross-cutting concerns

E. Linstead, P. Rigor, S. Bajracharya, C. Lopes, and P. Baldi, “Mining Concepts from Code with Probabilistic Topic Models,” *ASE*, 2007

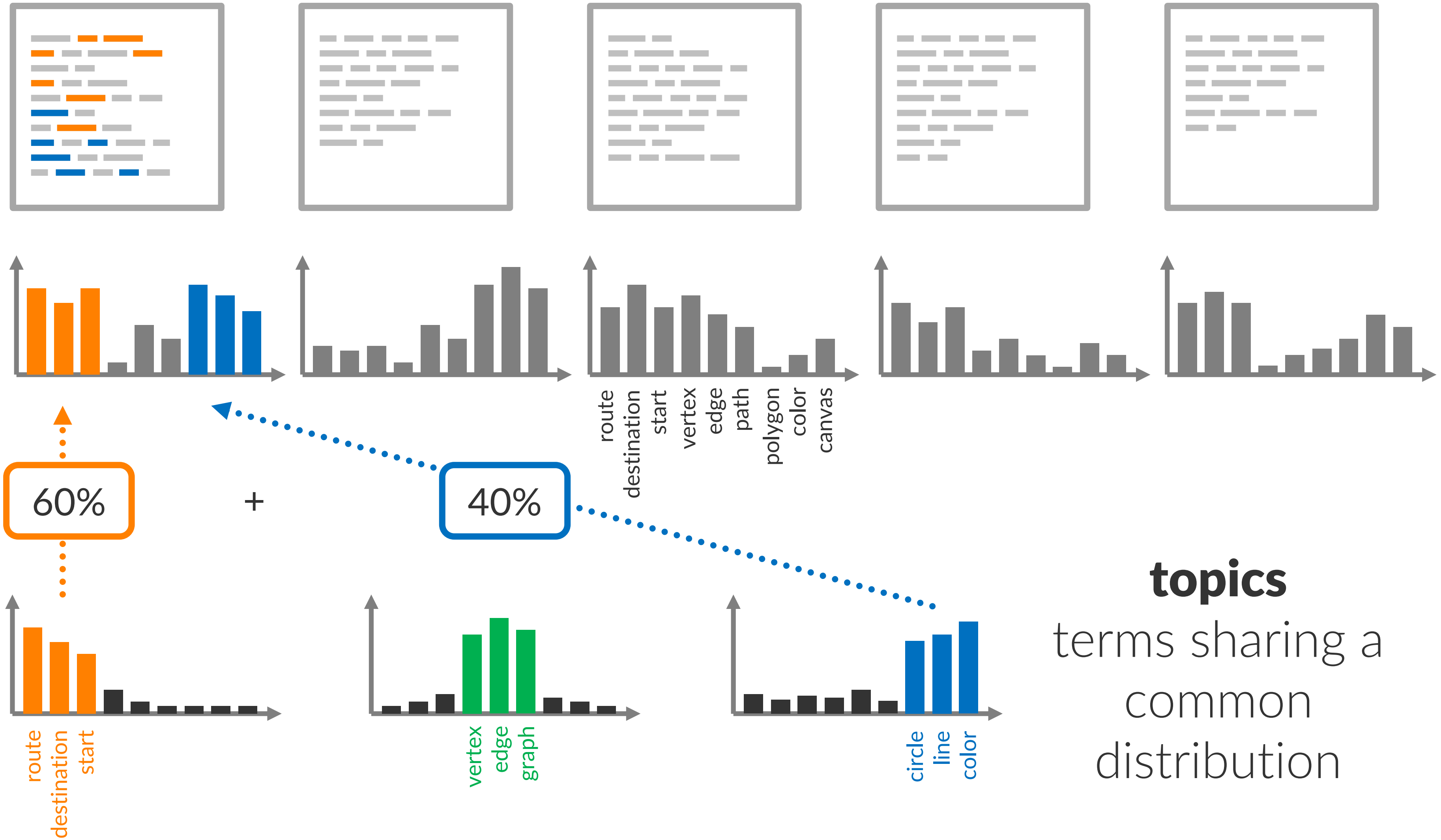
Related Work

Topic Models

Random Graph Models with Community Structure



Topic Models



topics

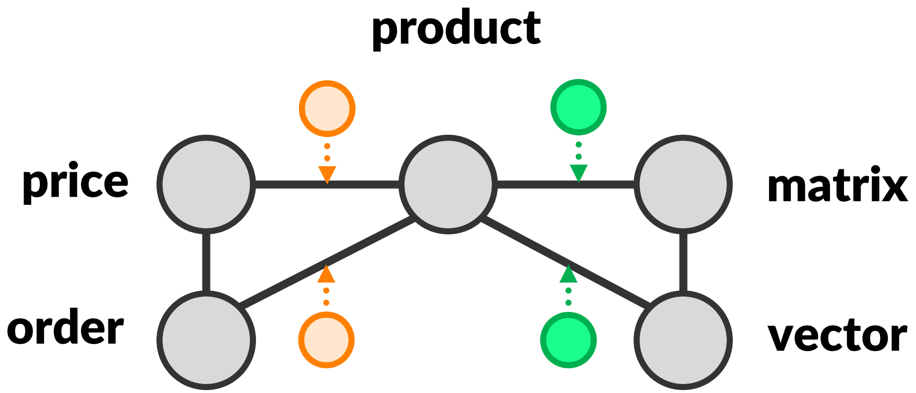
terms sharing a common distribution

Disambiguating Names

« product »

```
order.total += product.price;
```

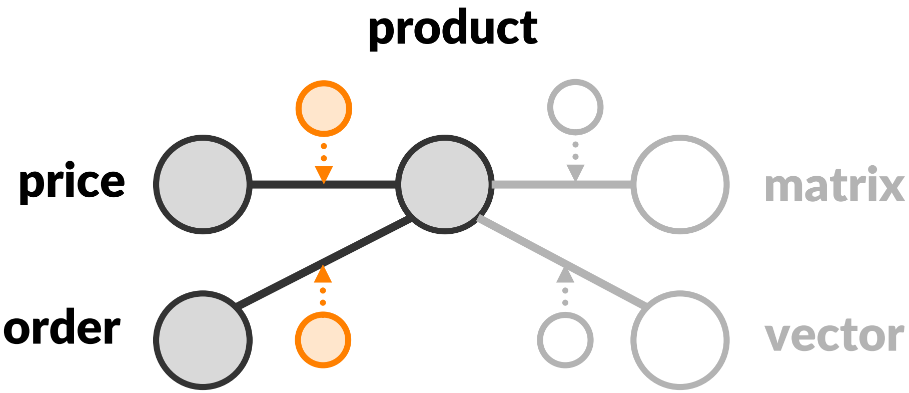
```
product = matrix * vector;
```



Disambiguating Names

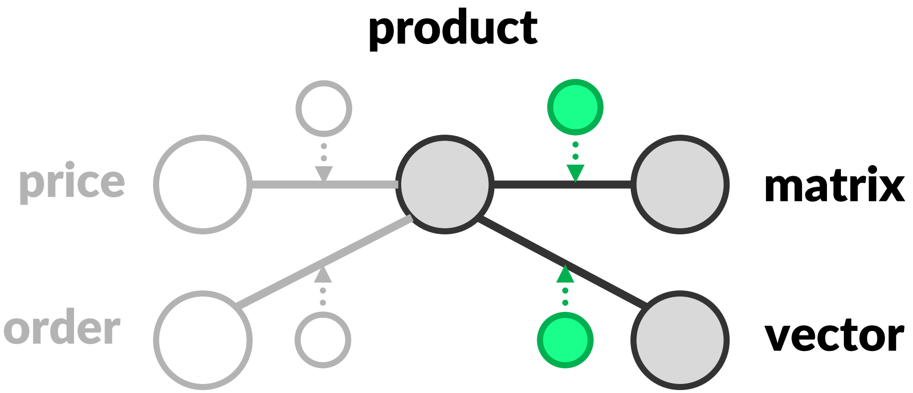
```
order.total += product.price;
```

```
product = matrix * vector;
```



Disambiguating Names

```
order.total += product.price;          product = matrix * vector;
```


product
price
matrix
order
vector