# Tracking the Flow of Ideas through the Programming Languages Literature

Michael Greenberg, Kathleen Fisher, and David Walker

SNAPL 2015




How can we understand the PL literature?

Which **related work** should I cite?

Should I submit to **PLDI** or **POPL**?

Who should I invite to this **PC**?

Who should **review** this paper?

Was this a **typical year** for ICFP?

How has **OOPSLA** changed over the years?

**Types**

**Optimization**

**Verification**

**Synthesis**

**Abstract Interpretation**

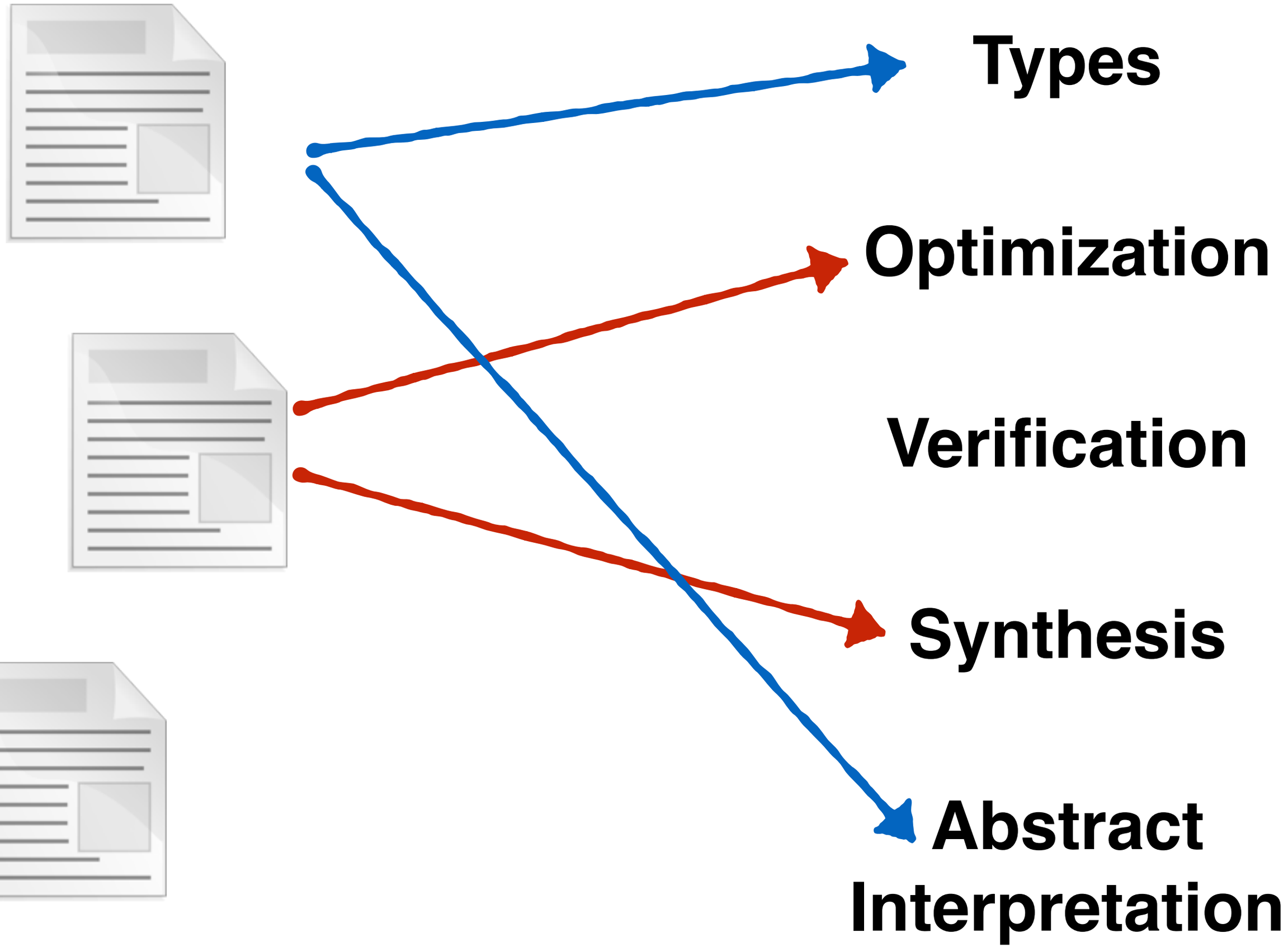
Types

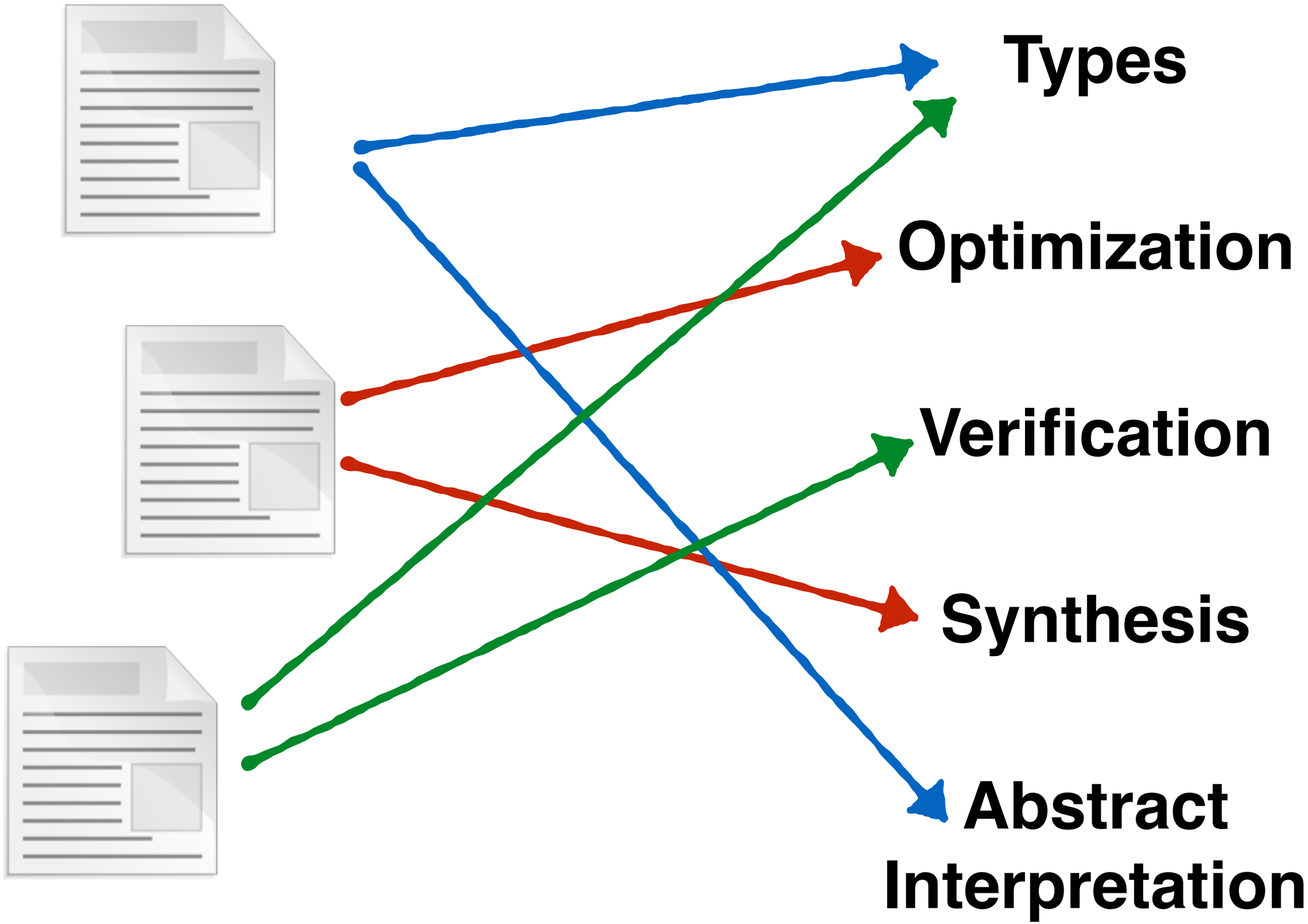
Optimization

Verification

Synthesis

Abstract Interpretation

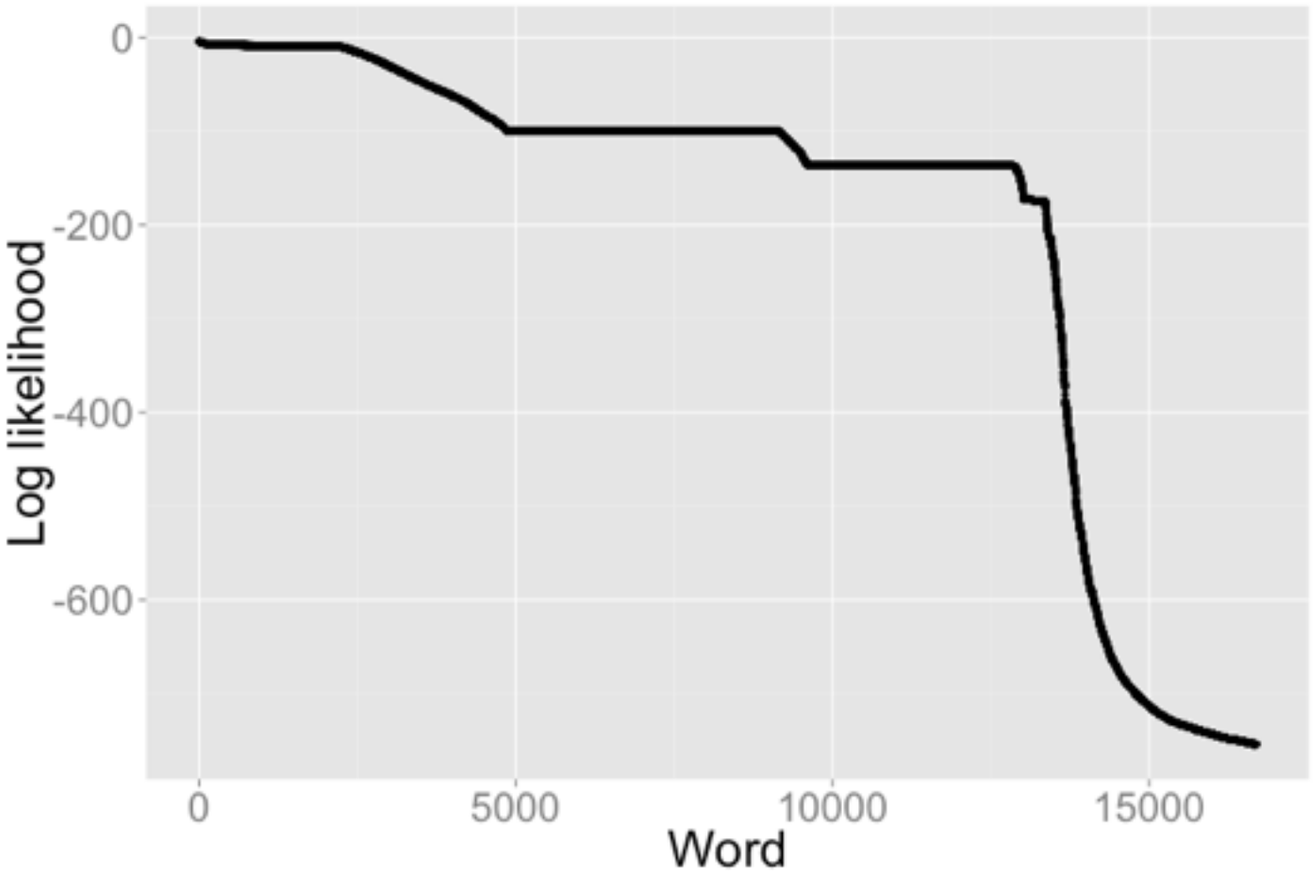

Types
Optimization
Verification
Synthesis
Abstract
Interpretation


Types
Optimization
Verification
Synthesis
Abstract
Interpretation

# What is a document's 'topic'?

| Word | Count |
|---|---|
| type | 120 |
| system | 83 |
| check | 34 |
| static | 21 |

# Topics are distributions of words




| “Parsing” topic | |
|---|---|
| **Word** | **Log likelihood** |
| **grammar** | -3.905040 |
| **language** | -4.206531 |
| **structure** | -4.308618 |
| **parser** | -4.513348 |
| **…** | … |

# Documents are a mix of topics

.6

| type systems | |
|---|---|
| **Word** | **Count** |
| **type** | 120 |
| **system** | 83 |
| **check** | 34 |
| **static** | 21 |

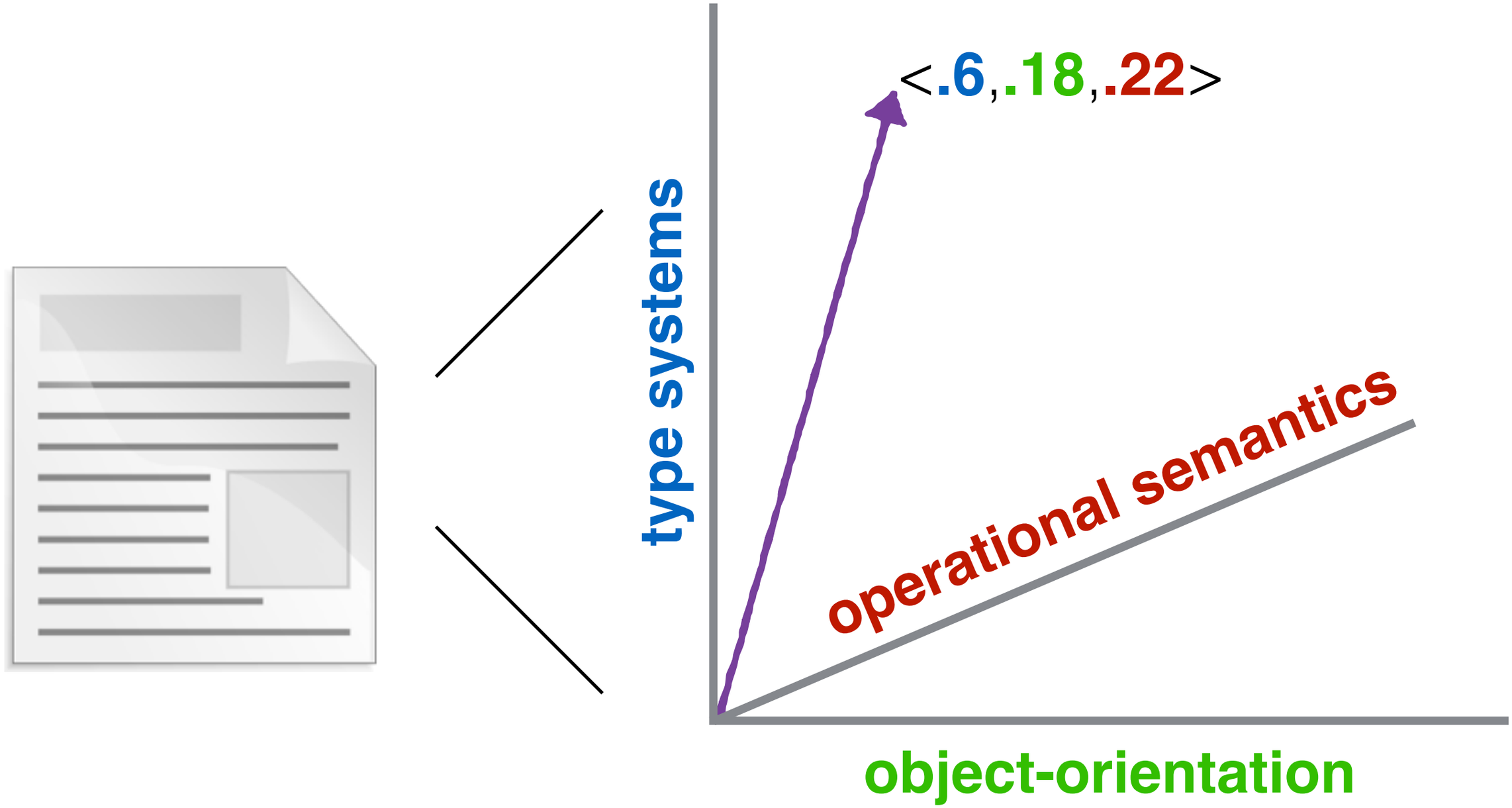
.18

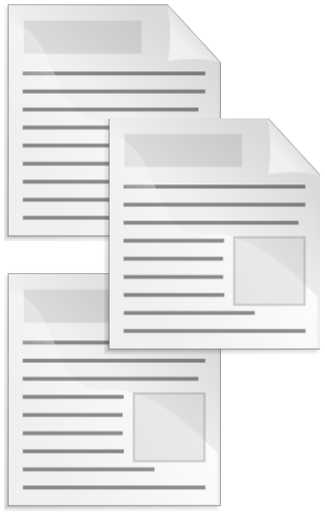
| operational semantics | |
|---|---|
| **Word** | **Count** |
| **semantics** | 90 |
| **step** | 45 |
| **reduce** | 38 |
| **evaluate** | 19 |

.22

| object-orientation | |
|---|---|
| **Word** | **Count** |
| **object** | 88 |
| **class** | 13 |
| **instance** | 12 |
| **method** | 7 |

# Documents are a mix of topics





two corpora

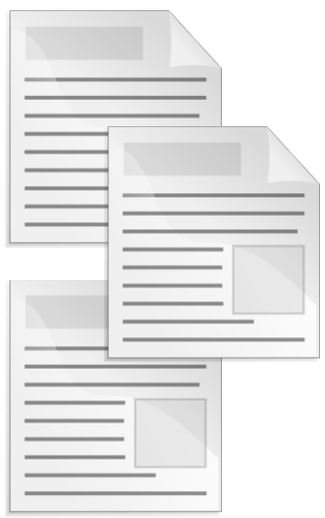
**N**=size

abstracts

(ICFP, OOPSLA, PLDI, POPL)

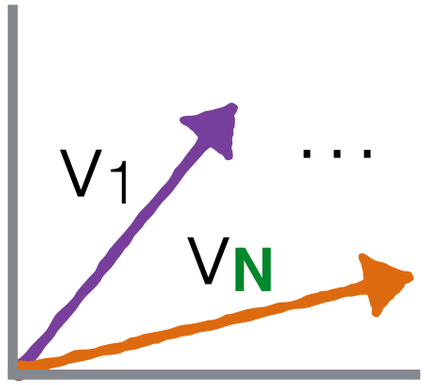
fulltext

(PLDI, POPL)

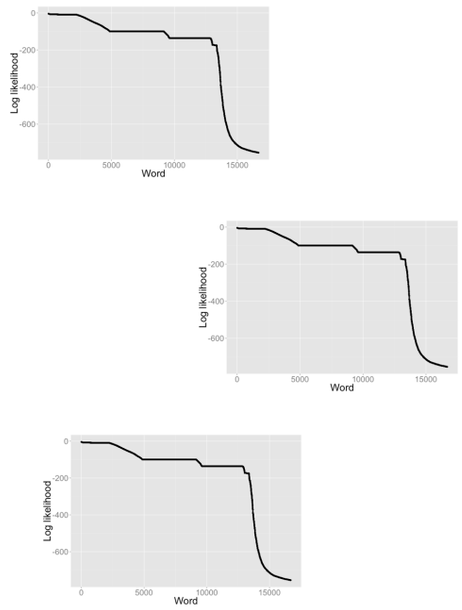


LDA-C

**k**

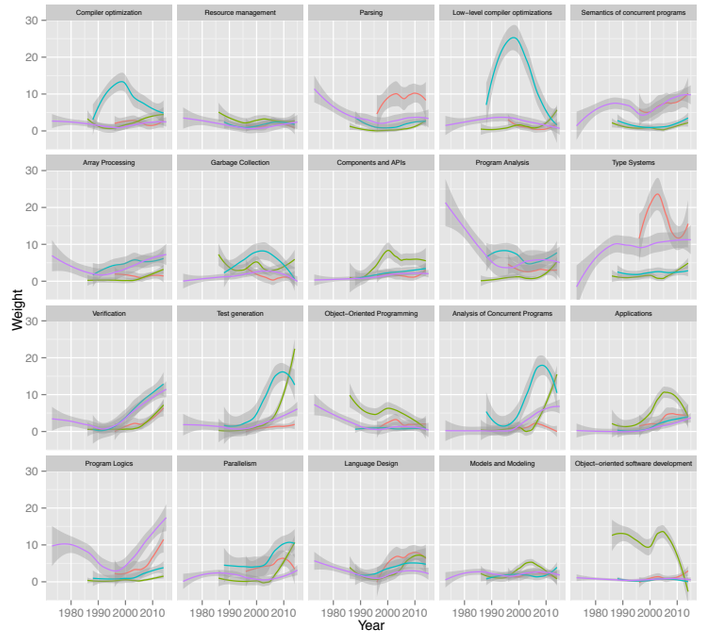


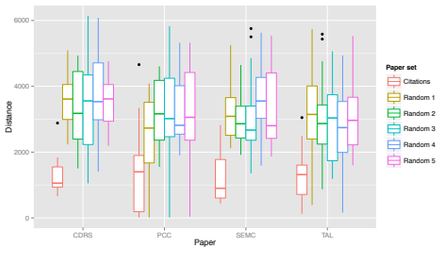

**N** vectors



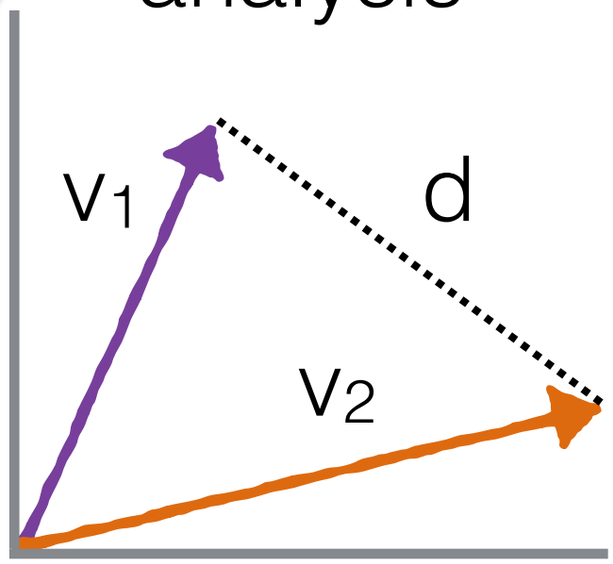

**k** topics

post



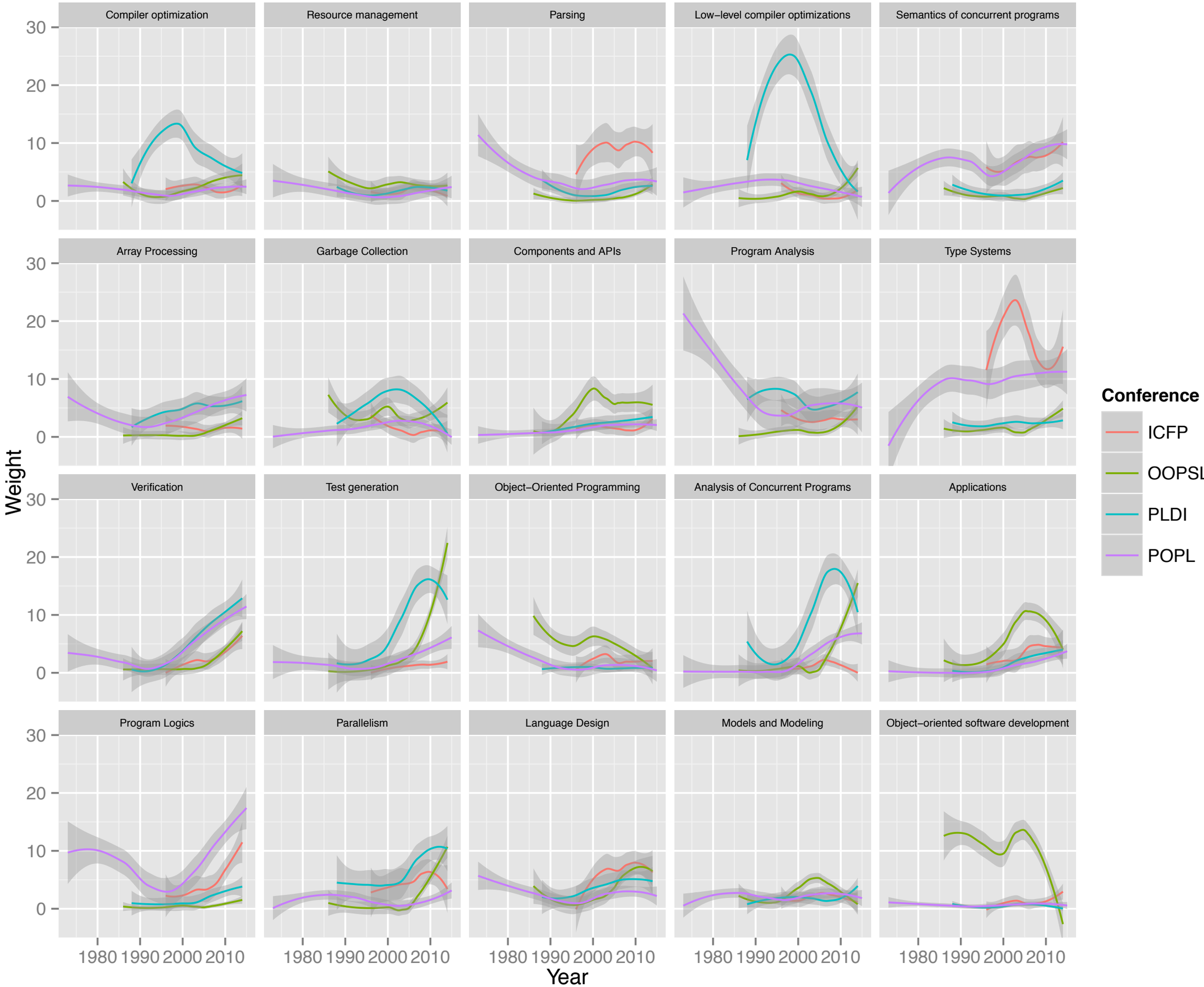

analysis

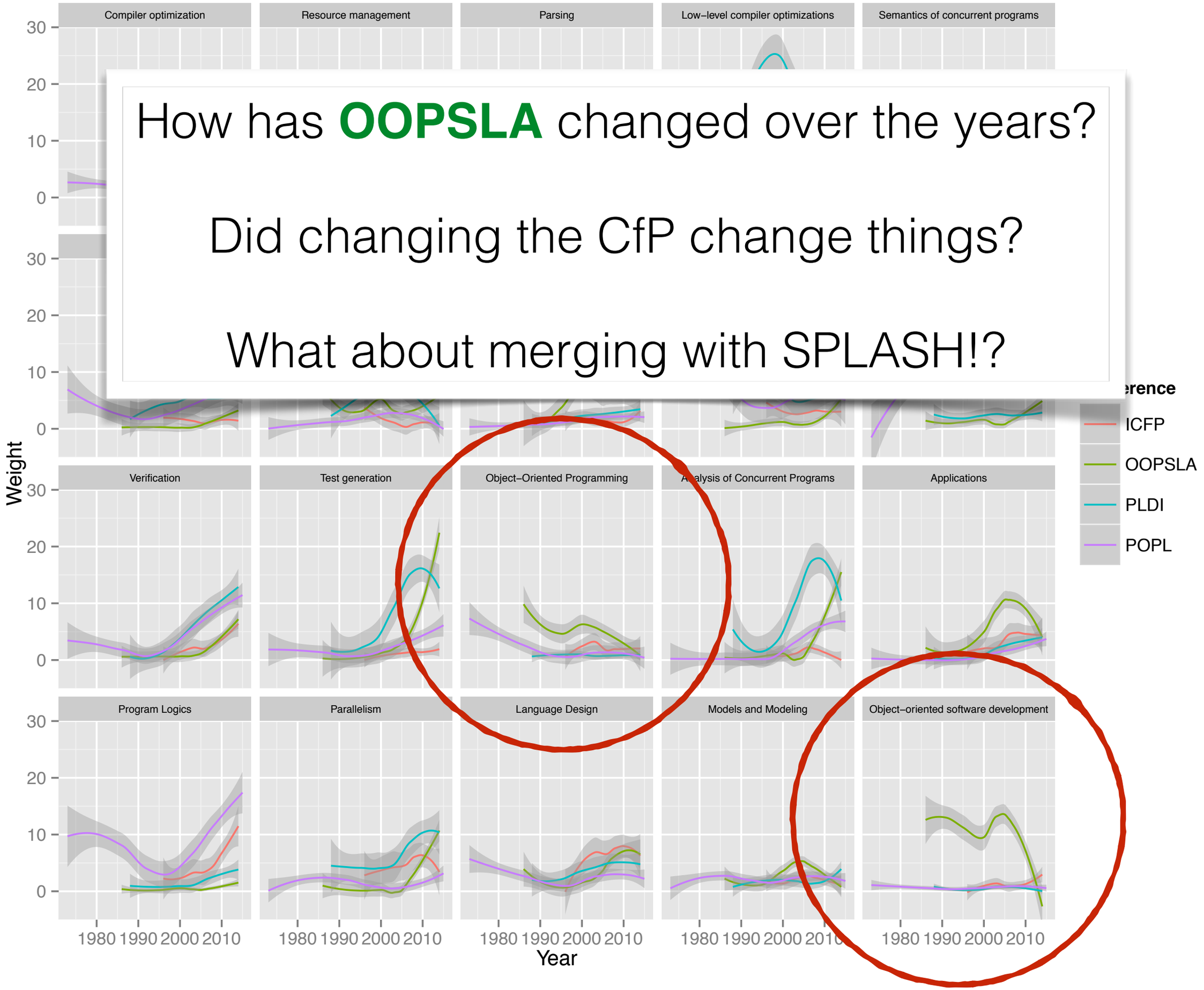



related work search

# Topic names for k=20, abstracts

| | | | |
|---|---|---|---|
| Compiler optimization | Array Processing | Verification | Program Logics |
| Resource management | Garbage Collection | Test generation | Parallelism |
| Parsing | Components and APIs | Object-Oriented Programming | Language Design |
| Low-level compiler optimizations | Program Analysis | Analysis of Concurrent Programs | Models and Modeling |
| Semantics of concurrent programs | Type Systems | Applications | Object-oriented software development |


Compiler optimization
Resource management
Parsing
Low−level compiler optimizations
Semantics of concurrent programs
Array Processing
Garbage Collection
Components and APIs
Program Analysis
Type Systems
Verification
Test generation
Object−Oriented Programming
Analysis of Concurrent Programs
Applications
Program Logics
Parallelism
Language Design
Models and Modeling
Object−oriented software development
Weight
Year
0
10
20
30
1980 1990 2000 2010
Conference
ICFP
OOPSLA
PLDI
POPL



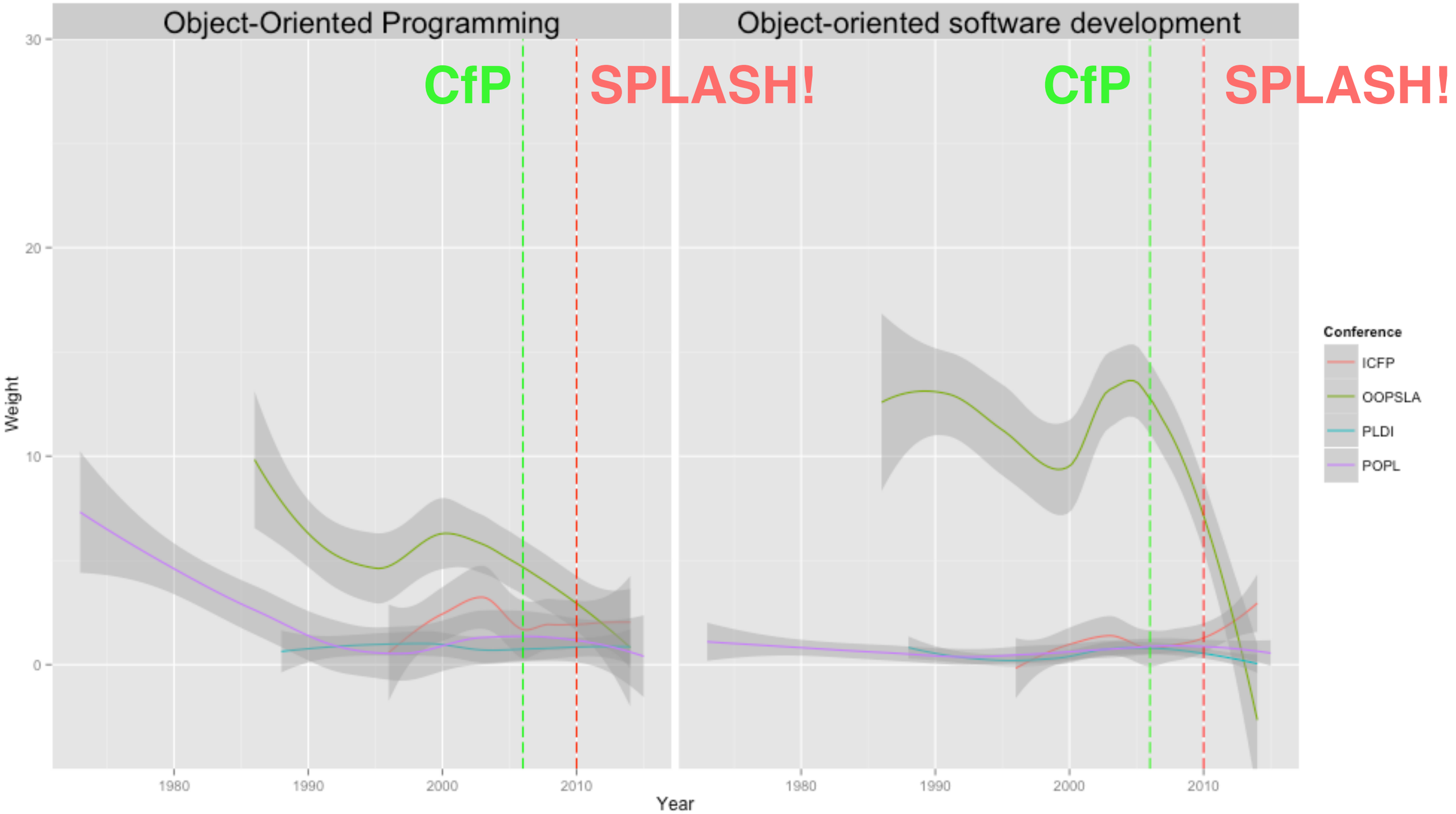

How has **OOPSLA** changed over the years?

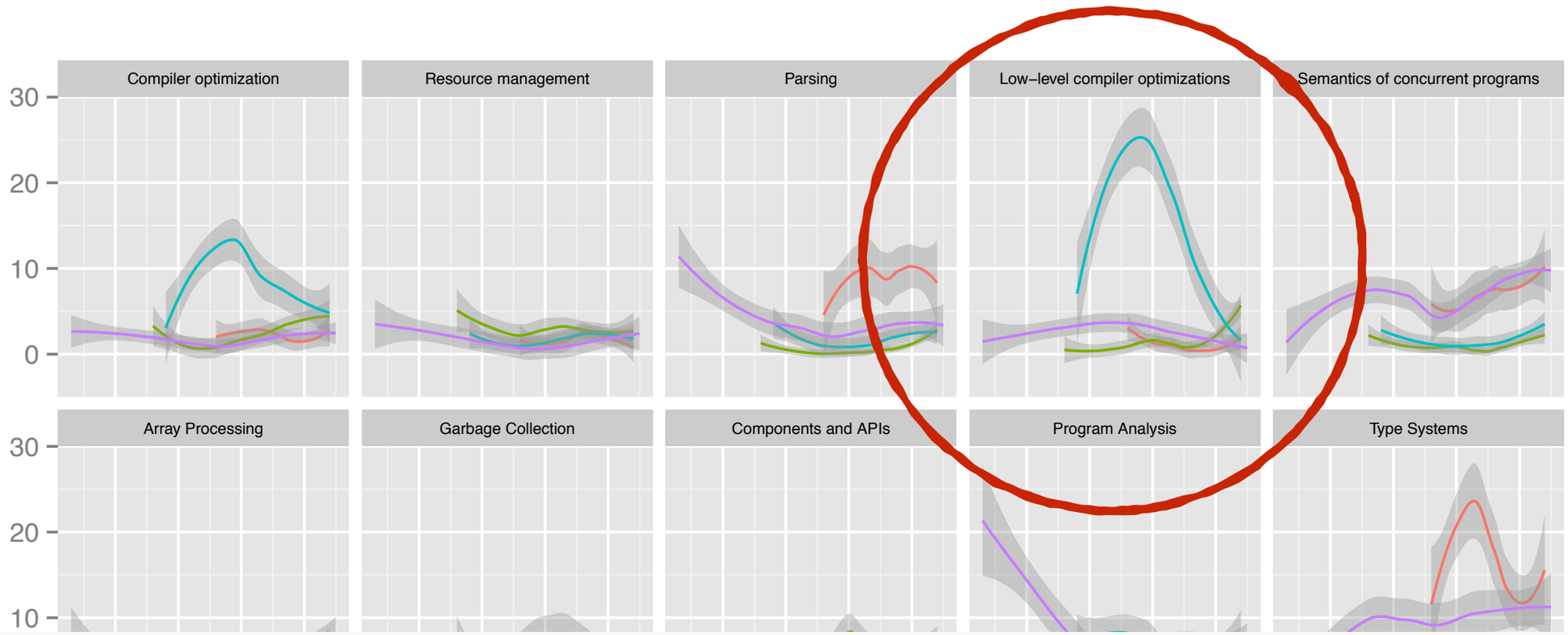
Did changing the CfP change things?

What about merging with SPLASH!?

# OOPSLA Call for Papers

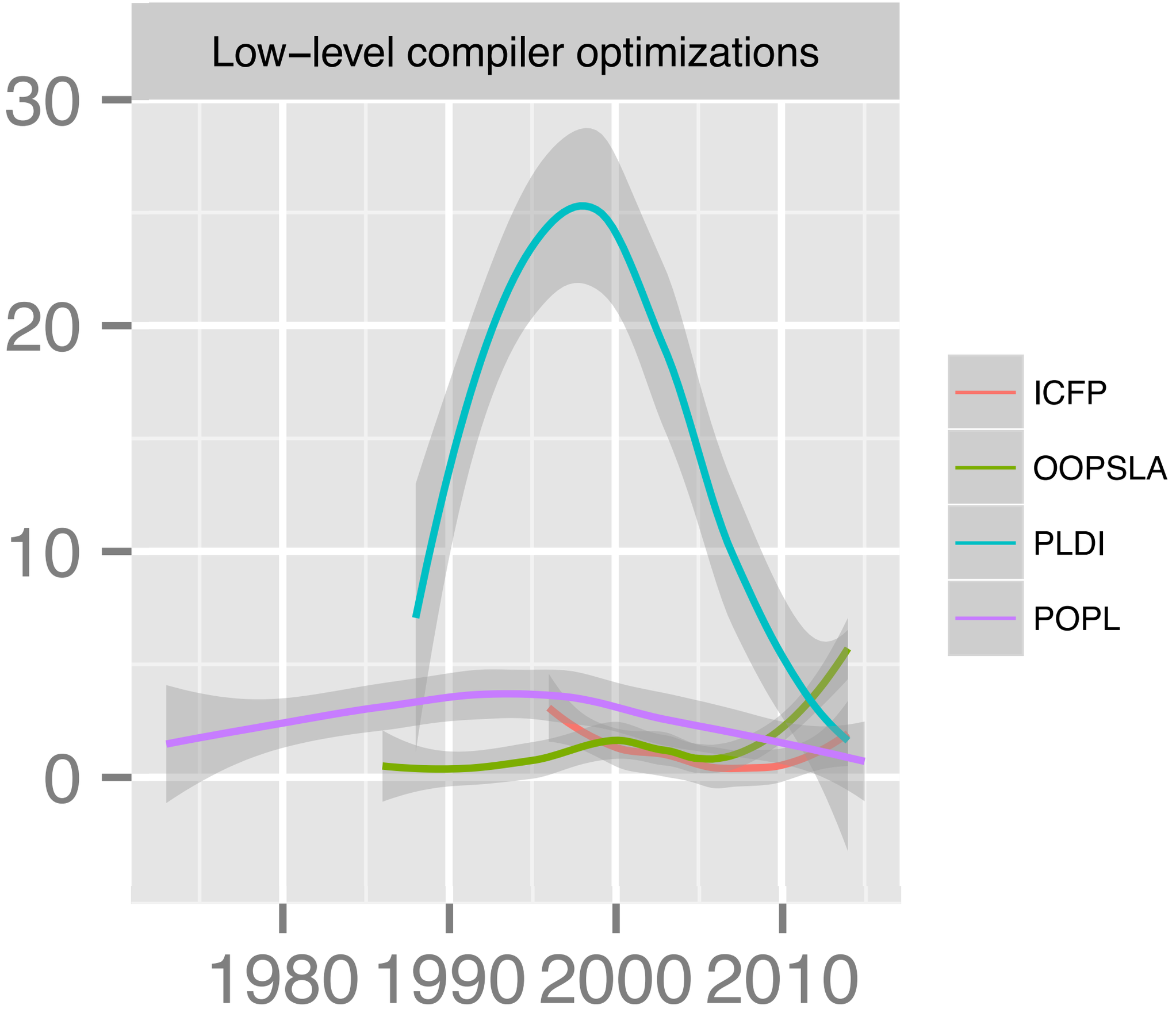
| 2006 | 2007 | 2010 |
|---|---|---|
| foundations of **object and related technologies** | paradigms **beyond the traditional concept** of object-oriented programming | **all aspects** of programming languages and software engineering, **broadly construed** |


Object-Oriented Programming
Object-oriented software development
CfP
SPLASH!
CfP
SPLASH!
Weight
30
20
10
0
1980
1990
2000
2010
Year
Conference
ICFP
OOPSLA
PLDI
POPL



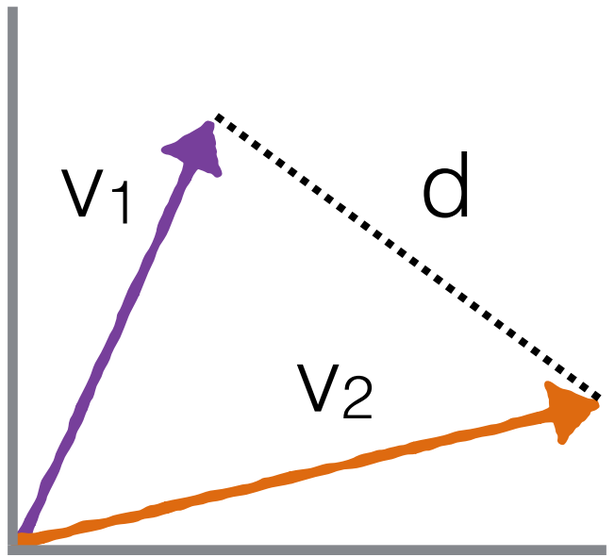

# How has **PLDI** changed over time?

# Per “Future of PLDI” session in Edinburgh, what is the state of the community?

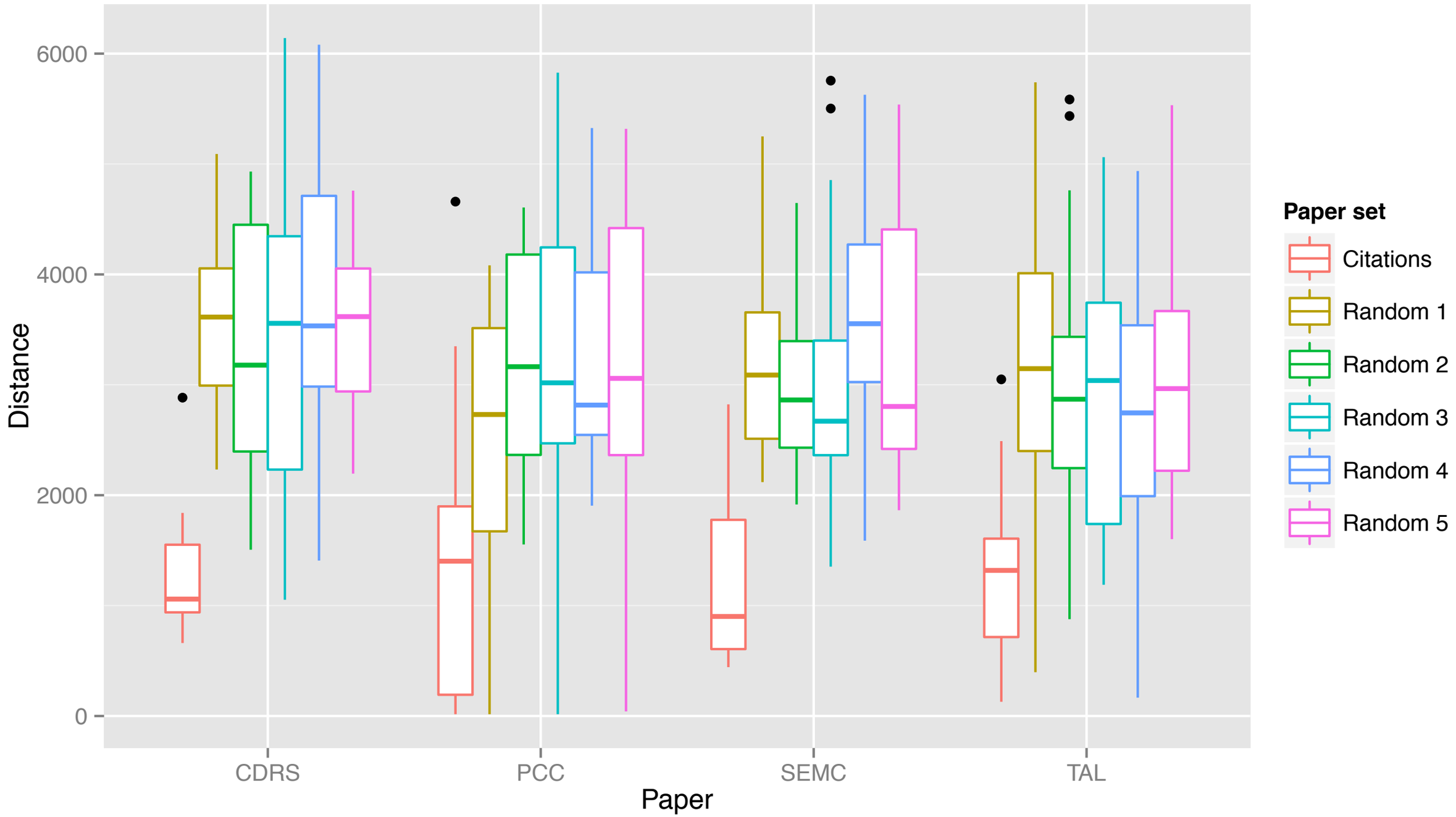

Low−level compiler optimizations
30
20
10
0
1980
1990
2000
2010
ICFP
OOPSLA
PLDI
POPL


Data-driven optimization
Register allocation
1000
750
500
250
0
Weight
1980
1990
2000
2010
Year
Conference
PLDI
POPL

# Comparing documents




Are papers with **close** topic vectors **related**?


6000
4000
2000
0
Distance
CDRS
PCC
SEMC
TAL
Paper
Paper set
Citations
Random 1
Random 2
Random 3
Random 4
Random 5

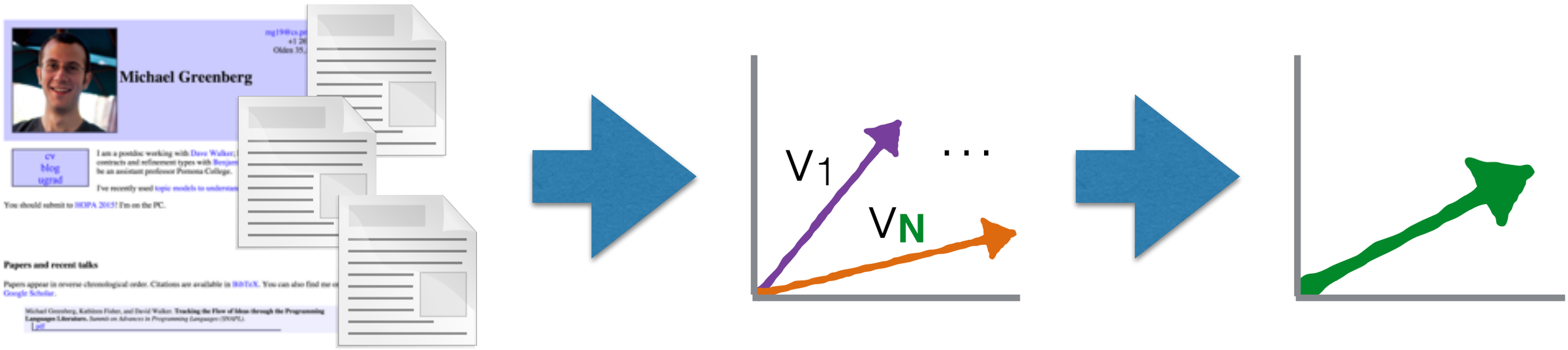
http://tmpl.weaselhat.com

# Ideas and plans

Beginning of a new project

What do *you* think we should do?

Models for researchers



# Discussion topics

# Topic names for k=20, full text

| | | | |
|---|---|---|---|
| Data-driven optimization | Abstract interpretation | Object-orientation | Code generation |
| Data-structure correctness | Languages and control | Security and bugfinding | Processes and message passing |
| Garbage collection | Parallelization | Program transformation | Dynamic analysis |
| Low-level systems | Design | Program analysis | Proofs and models |
| Register allocation | Types | Concurrency | Parsing |

# Let’s name a topic!

object

heap

region

memory

pointer

collector

garbage

collection

allocation

reference

Space overhead bounds for dynamic memory management with partial compaction

Schism: fragmentation-tolerant real-time garbage collection

Portable, unobtrusive garbage collection for multiprocessor systems

Limitations of partial compaction: towards practical bounds

Correctness-preserving derivation of concurrent garbage collection algorithms

The ramifications of sharing in data structures

A general framework for certifying garbage collectors and their mutators

Beltway: getting around garbage collection gridlock

On bounding time and space for multiprocessor garbage collection

Garbage collection without paging

# Let’s name a topic!

object

heap

region

memor

pointer

collector

garbage

collection

allocation

reference

Space overhead

ent with

n

r systems

llection

and their

and garbage collection gridlock

On bounding time and space for multiprocessor garbage collection

Garbage collection without paging

Garbage collection!

# Parsing

- Parsing drops standard **stopwords**
  - Added some extra ones with TF-IDF
- **Stemmed** words using nltk*
  - Removes plurals, etc.

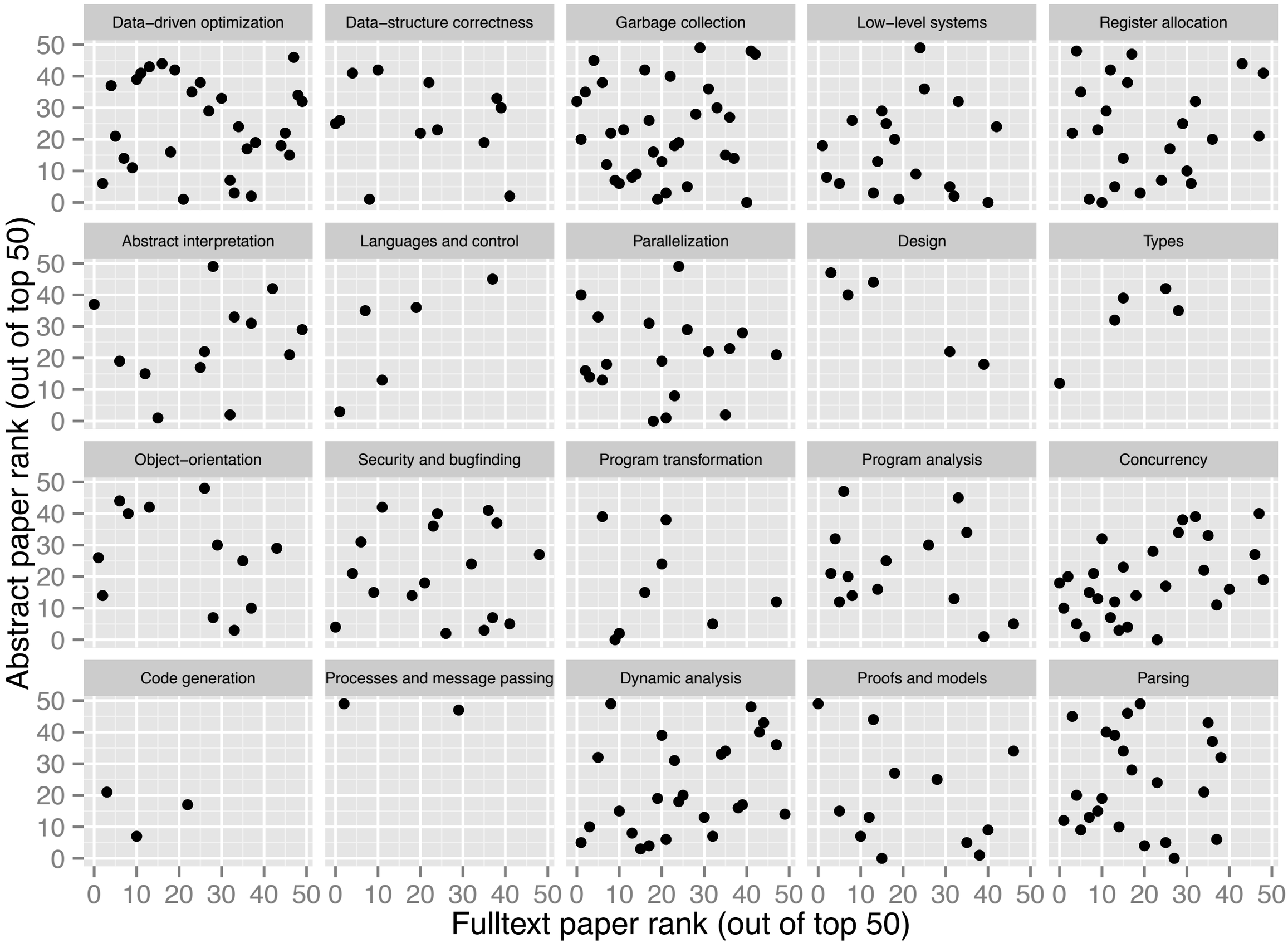
a
about
above
after
again
against
…

calculi → calculus
goes → go

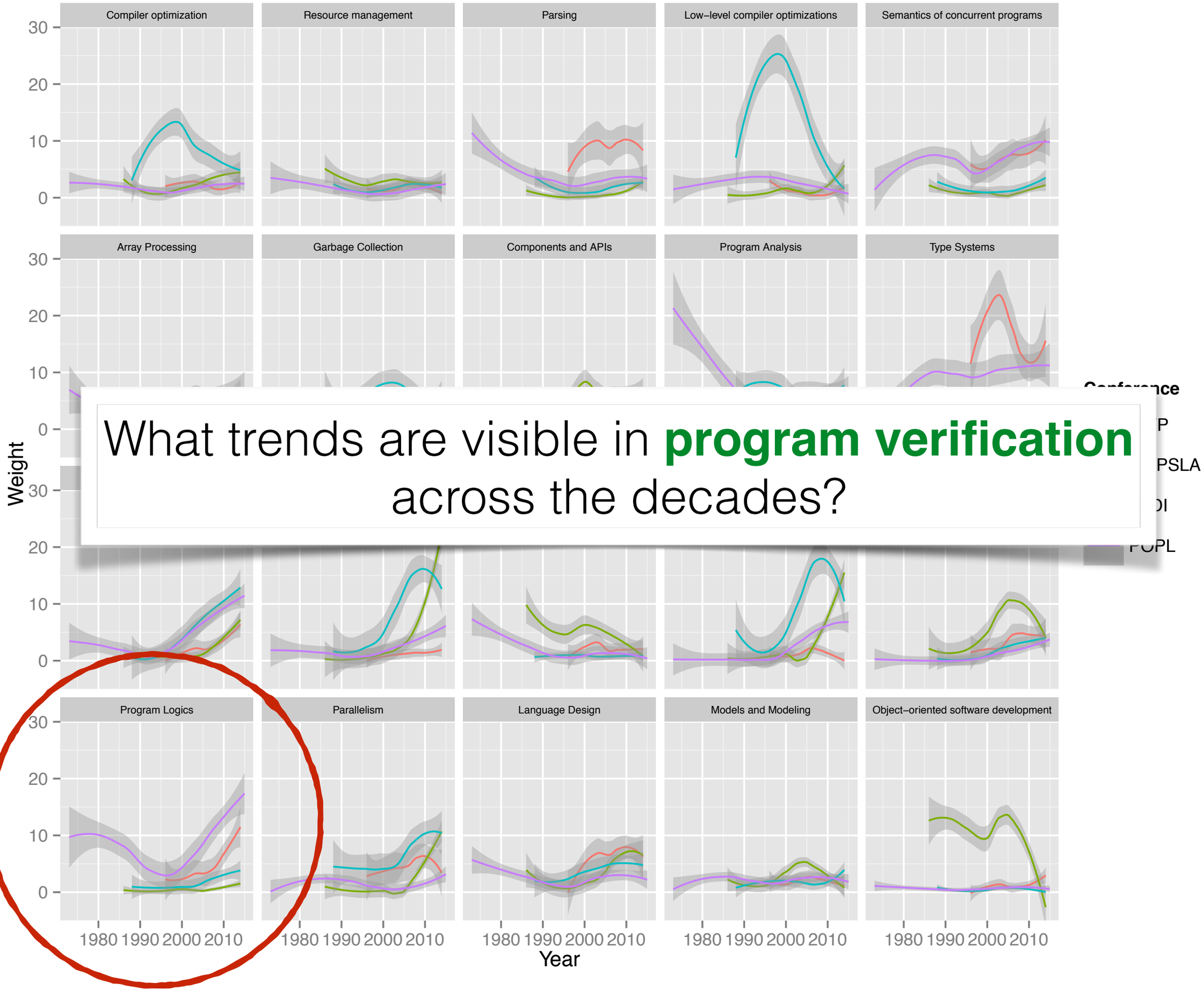
*http://www.nltk.org/

# Limitations/problems

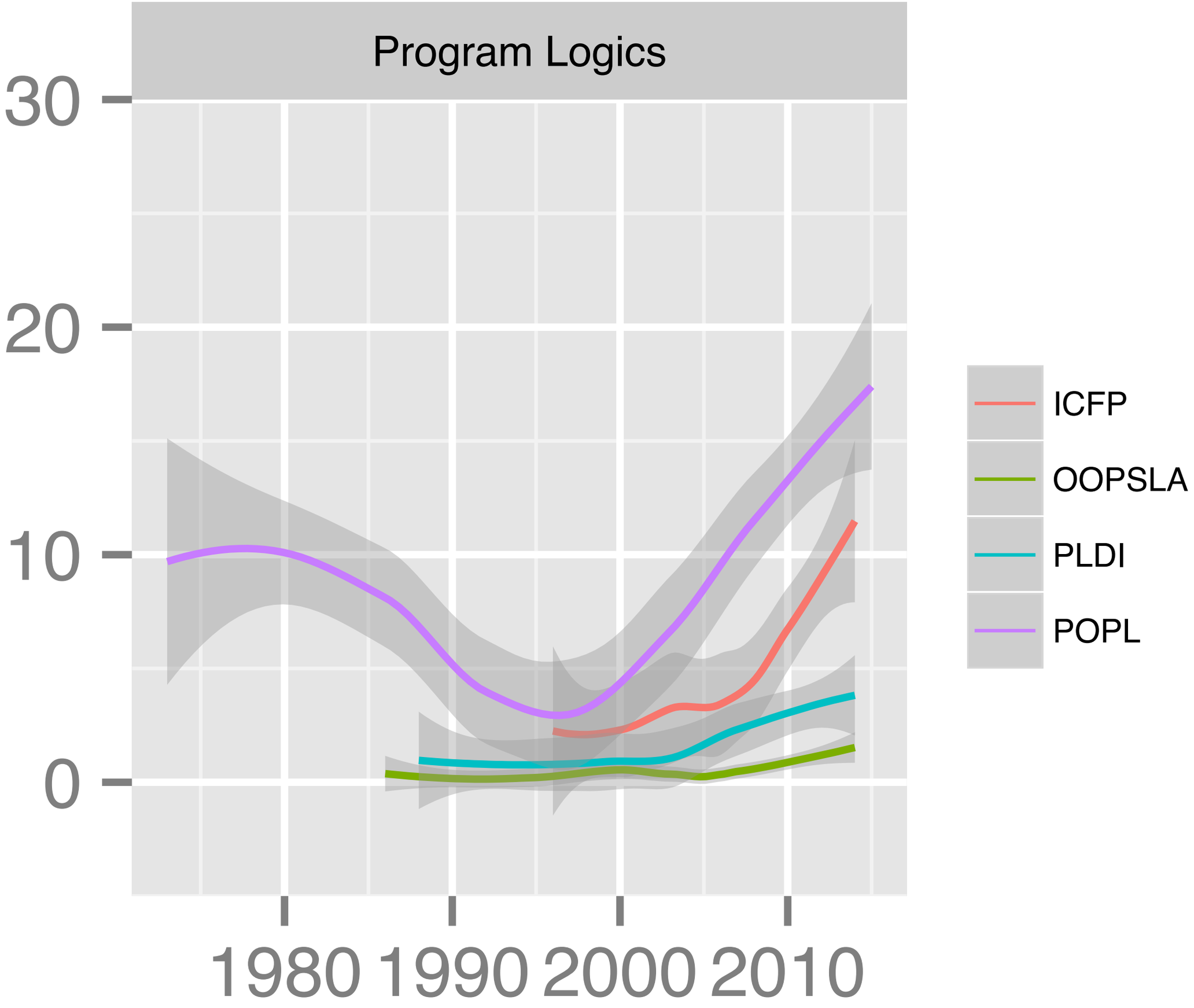
- ACM DL is missing data
  - No programmatic access
- Unclear choices about models
  - Abstracts or fulltext? k=20? k=30? k=200?
  - Which documents should ‘seed’ LDA?

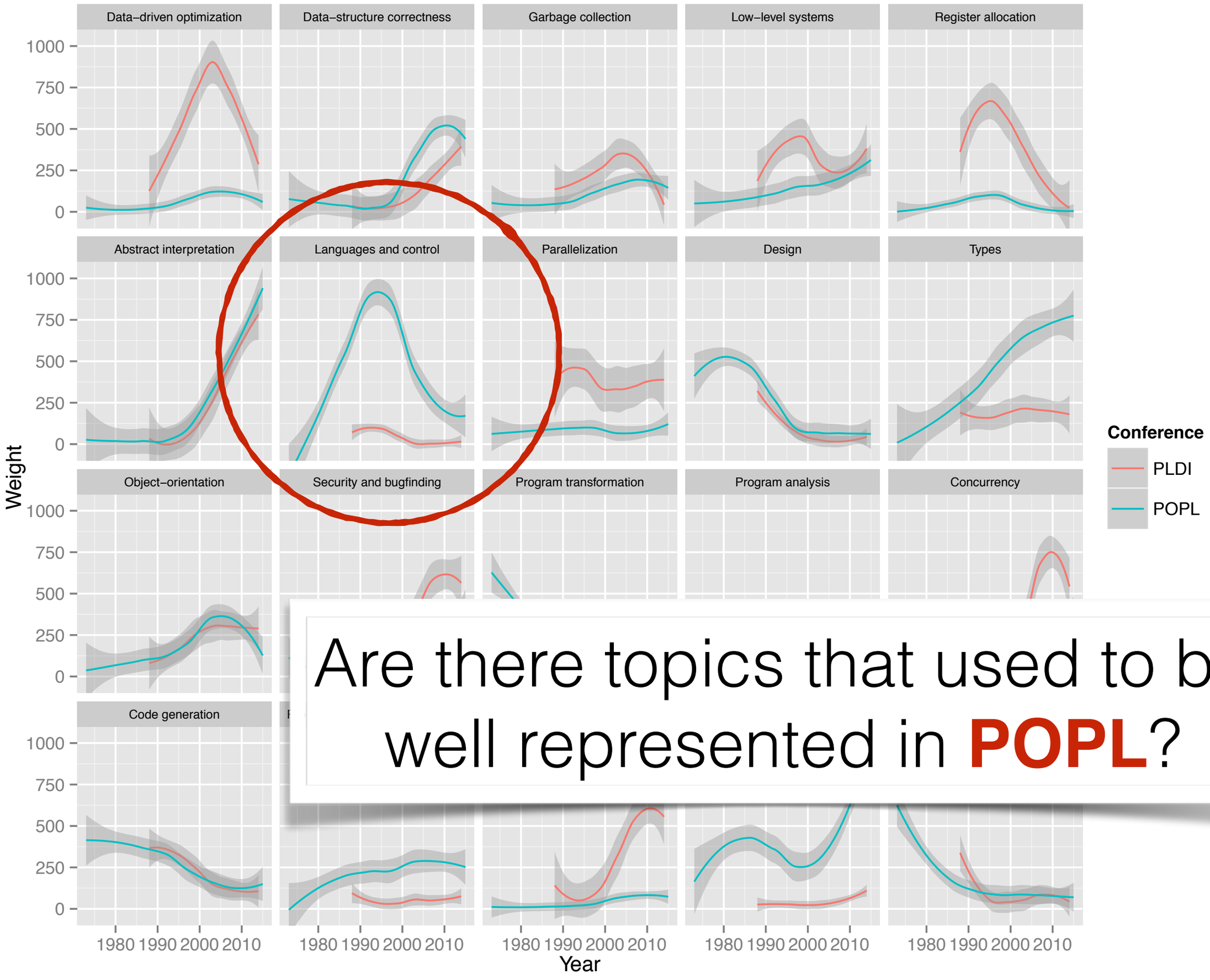

Data−driven optimization
Data−structure correctness
Garbage collection
Low−level systems
Register allocation
Abstract interpretation
Languages and control
Parallelization
Design
Types
Object−orientation
Security and bugfinding
Program transformation
Program analysis
Concurrency
Code generation
Processes and message passing
Dynamic analysis
Proofs and models
Parsing
Abstract paper rank (out of top 50)
Fulltext paper rank (out of top 50)
0
10
20
30
40
50

# More charts!

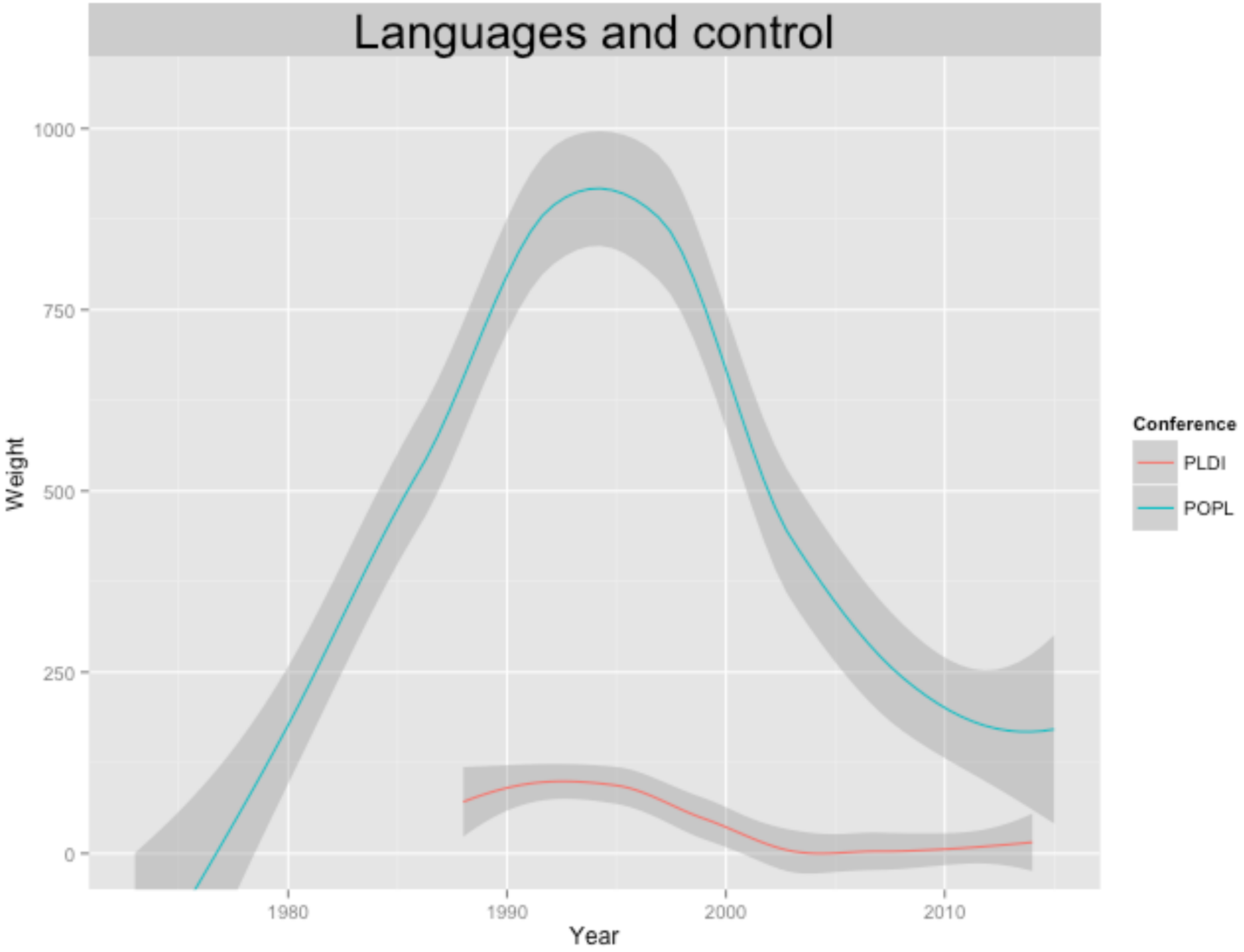


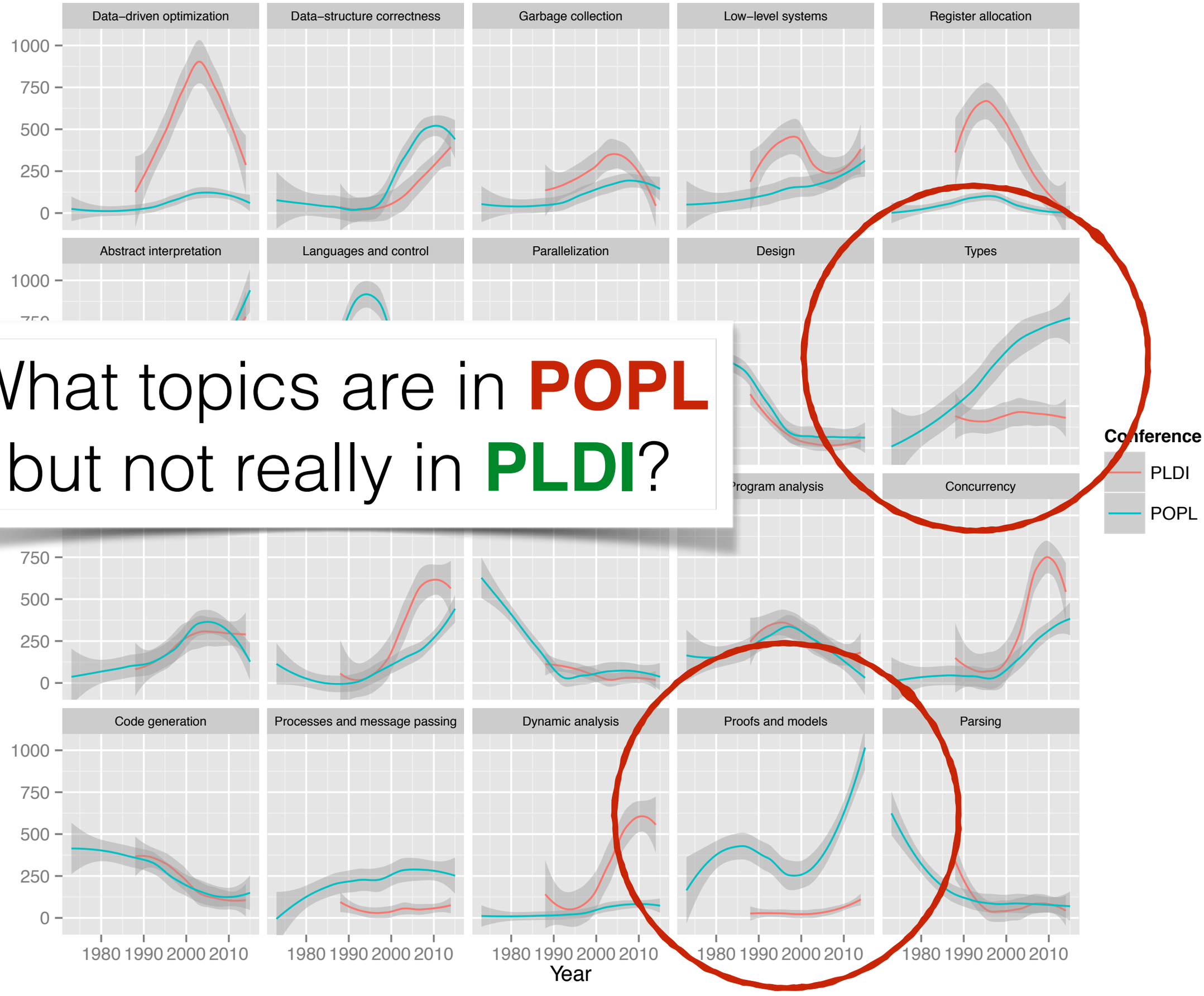

What trends are visible in **program verification** across the decades?


Program Logics
30
20
10
0
1980
1990
2000
2010
ICFP
OOPSLA
PLDI
POPL

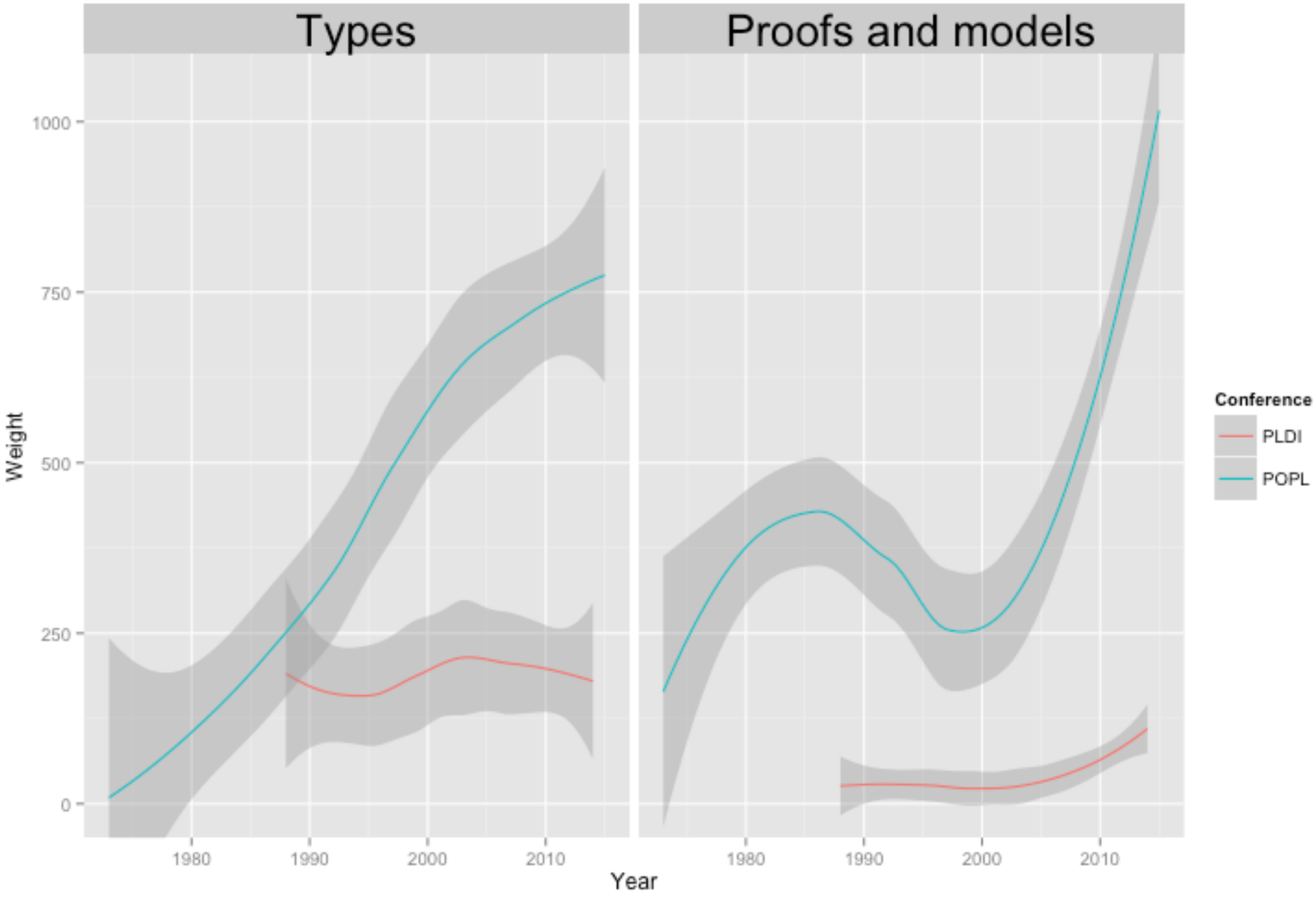



Are there topics that used to be well represented in **POPL**?


Languages and control
1000
750
500
250
0
Weight
1980
1990
2000
2010
Year
Conference
PLDI
POPL




# What topics are in **POPL** but not really in **PLDI**?


Types
Proofs and models
1000
750
500
250
0
Weight
1980
1990
2000
2010
Year
Conference
PLDI
POPL