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?

Alexandre Duret-Lutz
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
What is a document’s ‘topic’?

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>120</td>
</tr>
<tr>
<td>system</td>
<td>83</td>
</tr>
<tr>
<td>check</td>
<td>34</td>
</tr>
<tr>
<td>static</td>
<td>21</td>
</tr>
</tbody>
</table>
Topics are distributions of words

```
<table>
<thead>
<tr>
<th>Word</th>
<th>Log likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>grammar</td>
<td>-3.905040</td>
</tr>
<tr>
<td>language</td>
<td>-4.206531</td>
</tr>
<tr>
<td>structure</td>
<td>-4.308618</td>
</tr>
<tr>
<td>parser</td>
<td>-4.513348</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```

"Parsing" topic
Documents are a mix of topics:

**Type Systems**
- Type: 120
- System: 83
- Check: 34
- Static: 21

**Object-Oriented**
- Object-orientation: 88
- Class: 13
- Instance: 12
- Method: 7

**Operational Semantics**
- Semantics: 90
- Step: 45
- Reduce: 38
- Evaluate: 19
Documents are a mix of topics

- Type systems
- Operational semantics
- Object-orientation
two corpora \( N = \text{size} \)

abstracts (ICFP, OOPSLA, PLDI, POPL)

fulltext (PLDI, POPL)

LDA-C \( k \) topics

N vectors

post analysis

related work search
### Topic names for k=20, abstracts

<table>
<thead>
<tr>
<th>Compiler optimization</th>
<th>Array Processing</th>
<th>Verification</th>
<th>Program Logics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource management</td>
<td>Garbage Collection</td>
<td>Test generation</td>
<td>Parallelism</td>
</tr>
<tr>
<td>Parsing</td>
<td>Components and APIs</td>
<td>Object-Oriented Programming</td>
<td>Language Design</td>
</tr>
<tr>
<td>Low-level compiler optimizations</td>
<td>Program Analysis</td>
<td>Analysis of Concurrent Programs</td>
<td>Models and Modeling</td>
</tr>
<tr>
<td>Semantics of concurrent programs</td>
<td>Type Systems</td>
<td>Applications</td>
<td>Object-oriented software development</td>
</tr>
</tbody>
</table>
How has **OOPSLA** changed over the years?

Did changing the CfP change things?

What about merging with SPLASH!??
<table>
<thead>
<tr>
<th>2006</th>
<th>2007</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>foundations of object and related technologies</td>
<td>paradigms beyond the traditional concept of object-oriented programming</td>
<td>all aspects of programming languages and software engineering, broadly construed</td>
</tr>
</tbody>
</table>
How has PLDI changed over time?

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

Are papers with close topic vectors related?
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

<table>
<thead>
<tr>
<th>Data-driven optimization</th>
<th>Abstract interpretation</th>
<th>Object-orientation</th>
<th>Code generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-structure correctness</td>
<td>Languages and control</td>
<td>Security and bugfinding</td>
<td>Processes and message passing</td>
</tr>
<tr>
<td>Garbage collection</td>
<td>Parallelization</td>
<td>Program transformation</td>
<td>Dynamic analysis</td>
</tr>
<tr>
<td>Low-level systems</td>
<td>Design</td>
<td>Program analysis</td>
<td>Proofs and models</td>
</tr>
<tr>
<td>Register allocation</td>
<td>Types</td>
<td>Concurrency</td>
<td>Parsing</td>
</tr>
</tbody>
</table>
Let’s name a topic!

- **Object**
  - Space overhead bounds for dynamic memory management with partial compaction

- **Heap**
  - Schism: fragmentation-tolerant real-time garbage collection

- **Region**
  - Portable, unobtrusive garbage collection for multiprocessor systems

- **Memory**
  - Limitations of partial compaction: towards practical bounds

- **Pointer**
  - Correctness-preserving derivation of concurrent garbage collection algorithms

- **Collector**
  - The ramifications of sharing in data structures

- **Garbage**
  - A general framework for certifying garbage collectors and their mutators

- **Collection**
  - Beltway: getting around garbage collection gridlock

- **Allocation**
  - On bounding time and space for multiprocessor garbage collection

- **Reference**
  - Garbage collection without paging
Let’s name a topic!

- Object heap region memory pointer collector garbage collection
- 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
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
go 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?
More charts!
What trends are visible in program verification across the decades?
Are there topics that used to be well represented in POPL?
What topics are in **POPL** but not really in **PLDI**?