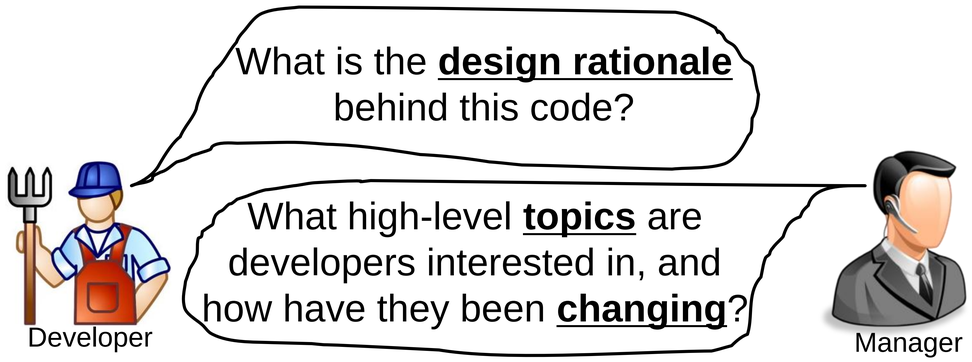
Mining Software Repositories with Topic Models

Stephen W. Thomas (sthomas@cs.queensu.ca) Queen's University, Canada

Software Analysis & Intelligence Lab

Queen's University

1. Problems to Address

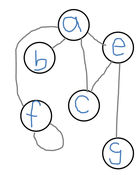


Mining software repositories can help:

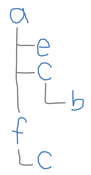
- Developer thoughts and intentions are encoded in **source code identifiers and comments**, and
- discussions about code design is contained in **email archives**.

Both of these repositories are *unstructured*.

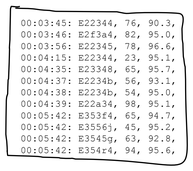
2. Challenges with Unstructured Data



Static call graphs



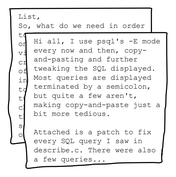
Stack traces



Execution logs

Structured data.

Code comments and identifiers

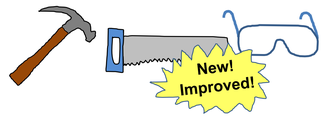


Email archives

Bug reports

Unstructured data.

Need better methodologies and techniques to use (exploit!) unstructured data.



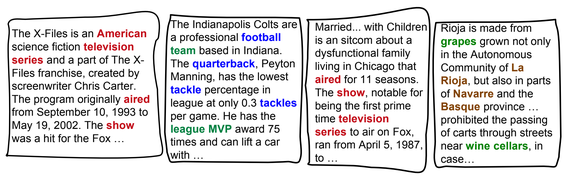
Challenges:

- Natural language and noise
- Huge volumes of data, evolving over time
- No explicit links between data

3. Existing (Advanced) Approach to Exploit Unstructured Data

Use statistical **topic models**, from NLP, to structure and analyze the unstructured data.
BONUS! Topic models scale and are meta-model agnostic (i.e., work on any text documents).
WARNING! Topics models to designed for use on source code (Thomas et al. 2011).

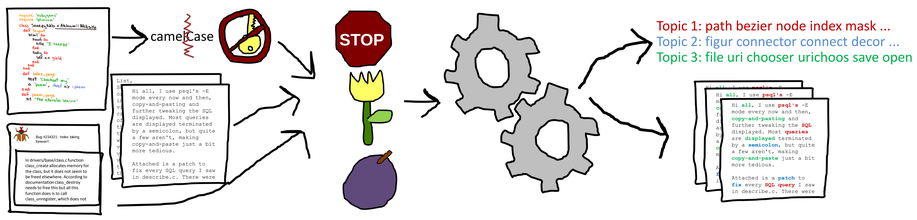
Latent Dirichlet Allocation (LDA) Unsupervised; automatic; discovers *topics* from any corpus



Topic 1: american, television, show, series, aired, ...
Topic 2: football, quarterback, touchdown, tackle, ...
Topic 3: wine, grape, taste, cellar, spain, glass, ...

Traditional applications of LDA Clustering; IR; concept location

Applying to code, emails, and bugs split identifiers; remove programming keywords; stop; stem; prune



Topic 1: path bezier node index mask ...
Topic 2: figur connector connect decor ...
Topic 3: file uri chooser urichoos save open

4. Software Engineering Applications

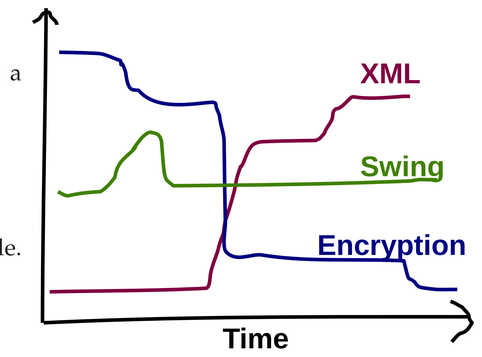
Concept location and concept evolution (In progress)
Method: Use LDA to discover topics and their evolutions

Benefits: Allows practitioners to monitor and reason about code at a higher level:

- Code refactoring and understanding
- Project monitoring
- Acquisition support

Contribution: Prior work did not consider characteristics of source code. Our ***Diff model*** accounts for source code characteristics, resulting in:

- Better topics
- More sensitive and accurate topic evolutions

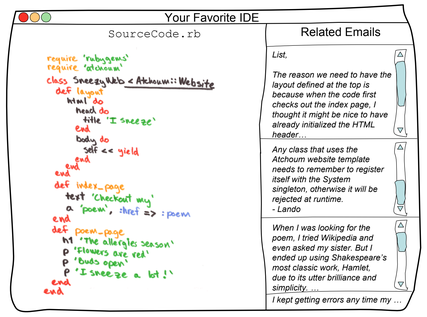


Source-email traceability link recovery (In progress)
Method: Link engine based on keywords (Bacchelli et al. 2010) and topics

Benefits: Allows developers to locate design discussions of code

- Understand design rationale
- Recover the flow of knowledge

Contribution: Prior work was based on keywords only. Our work also considers higher-level concepts, resulting in better traceability links.



For more information: http://cs.queensu.ca/∼sthomas