On the Equivalence of Information Retrieval Methods for Automated Traceability Link Recovery

Rocco Oliveto, Malcom Gethers, ***Denys Poshyvanyk***,
Andrea De Lucia

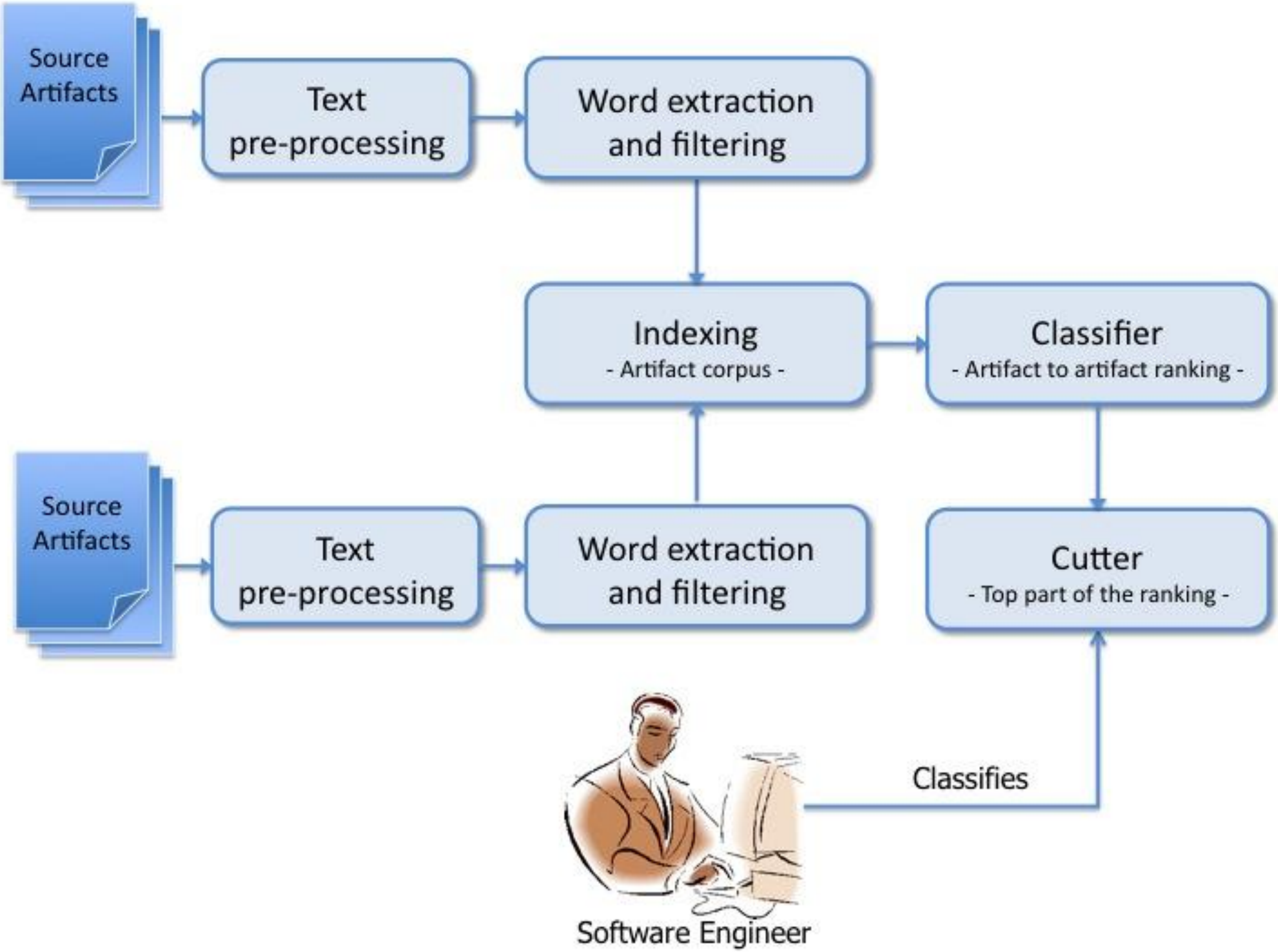
Traceability Management

- Traceability...
  - “the ability to describe and follow the life of an artifact, in both a forwards and backwards direction”
- Maintaining traceability between software artifacts is important for software development and maintenance
  - program comprehension
  - impact analysis
  - software reuse

Traceability Link Recovery

- Most software artifacts contains text
- Conjecture: artifacts having a high text similarity are likely good candidates to be traced onto each other
- IR techniques can be used to calculate the similarity between software artifacts

Tracing Software Artifacts Using IR Methods



Classifier: two basic models

- Probabilistic model
  - The similarity between a source and a target artifact is based on the probability that the target artifact is related to the source artifact (i.e., Jensen-Shannon)
- Vector space model
  - Source and target artifacts are represented in a vector space (of terms) and the similarity is computed through vector operations
- Improvements to basic models:
  - Latent Semantic Indexing
  - Latent Dirichlet Allocation

Vector Space Model

- Software artifacts are represented as vectors in the space of terms (vocabulary)
- Vector values might be values (the term is or is not in the artifact)
- Usually computed as the product of a local and a global weights
  - Local weight: based on the frequency of occurrences of the term in the document
  - Global weight: the more the term is spread in the artifact space the less it is relevant to the subject document

Latent Semantic Indexing

- Extension of the Vector Space Model based on Singular Value Decomposition (SVD)
  - The term-by-document matrix is decomposed into a set of k orthogonal factors from which the original matrix can be approximated by linear combination
- Overcomes some of the deficiencies of assuming independence of words (co-occurrences analysis)
  - Provides a way to automatically deal with synonymy
  - Avoids preliminary text pre-processing and morphological analysis (stemming)

Latent Dirichlet Allocation

- LDA is a generative probabilistic model where documents are modeled as random mixtures over latent topics
- LDA is similar to pLSA, except that in LDA the topic distribution is assumed to have a Dirichlet distribution
- We use Hellinger distance, a symmetric similarity measure between two probability distributions

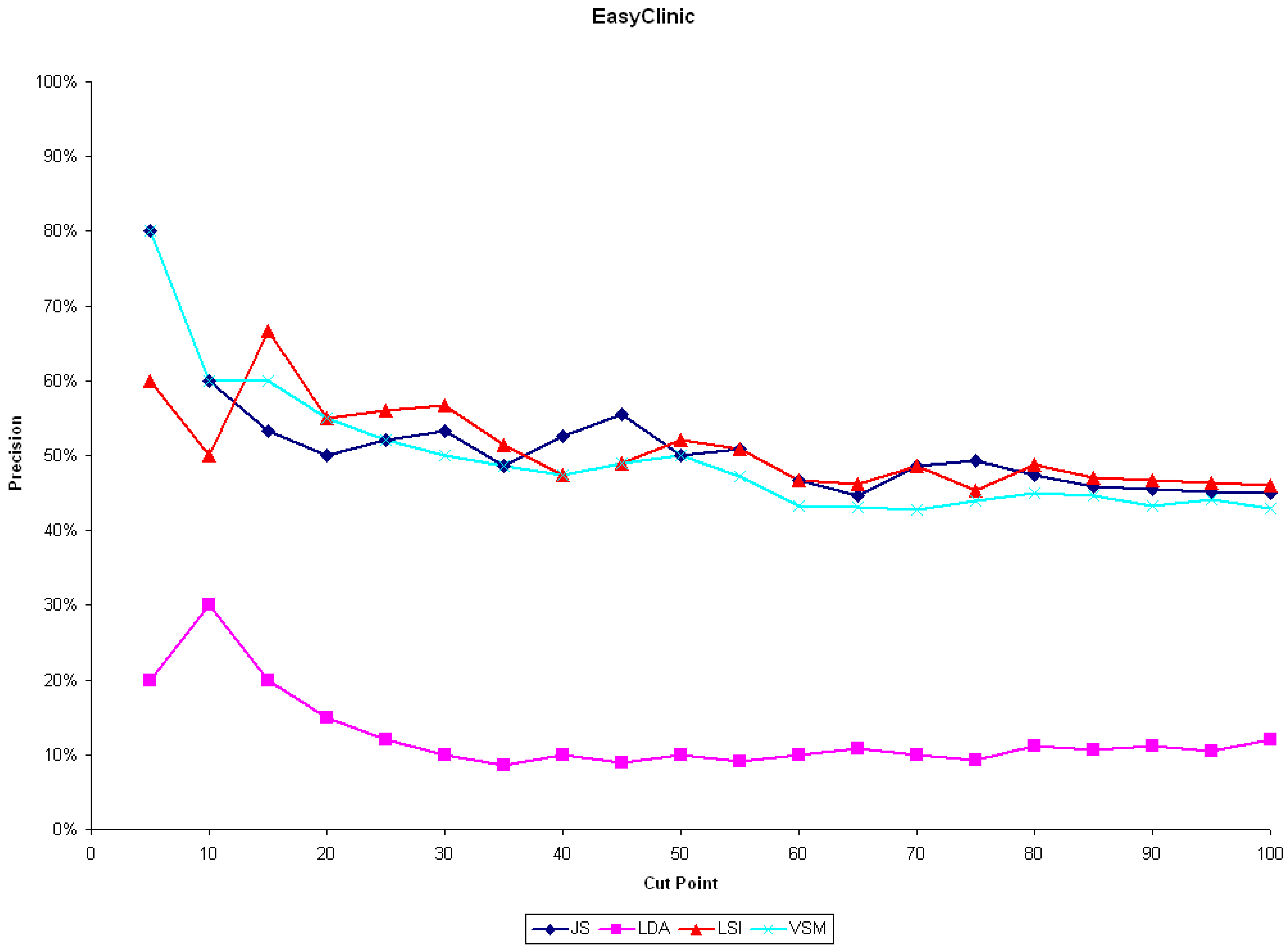
Motivation

- No empirical studies on evaluating multiple IR methods for traceability link recovery:
  - Latent Semantic Indexing (LSI)
  - Vector Space Model (VSM)
  - Jenson-Shannon (JS)
  - Latent Dirichlet Allocation (LDA)
- Some studies indicate controversial results
- *Which IR technique should I use?*

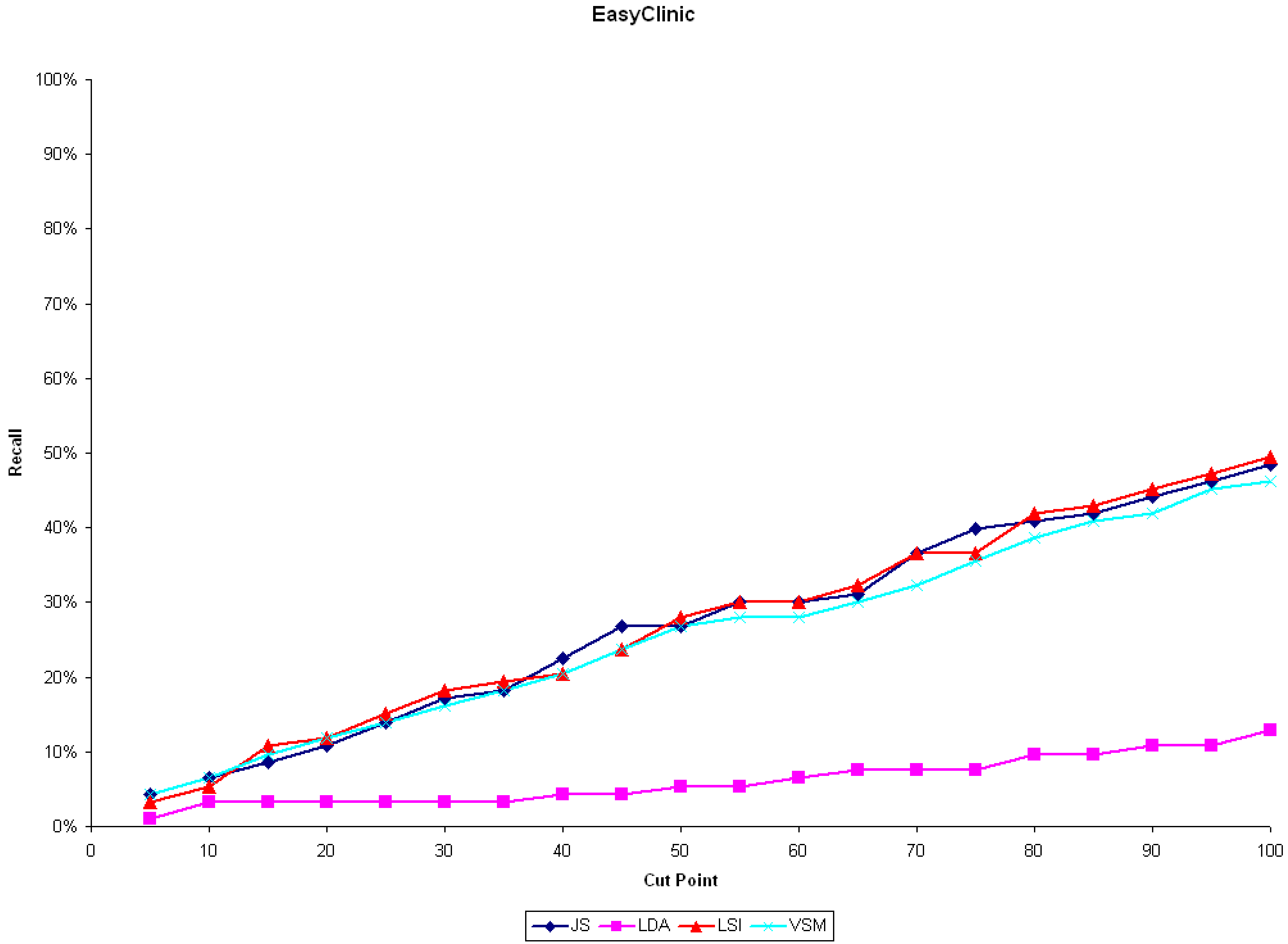
Empirical Assessment of Traceability Link Recovery Techniques

- Research questions (RQ)
  - **RQ1**: Which is the IR method that provides the more accurate list of candidate links?
  - **RQ2**: Do different types of IR methods provide orthogonal similarity measures?
- Design of the case studies
  - EasyClinic and eTour software systems
    - EasyClinic: 93 out of 1,410 possible links
    - eTour: 364 out of 6,728 possible links
  - IR techniques: JS, VSM, LSI and LDA
  - Case study data: www.cs.wm.edu/semeru/data/icpc10-tr-lda

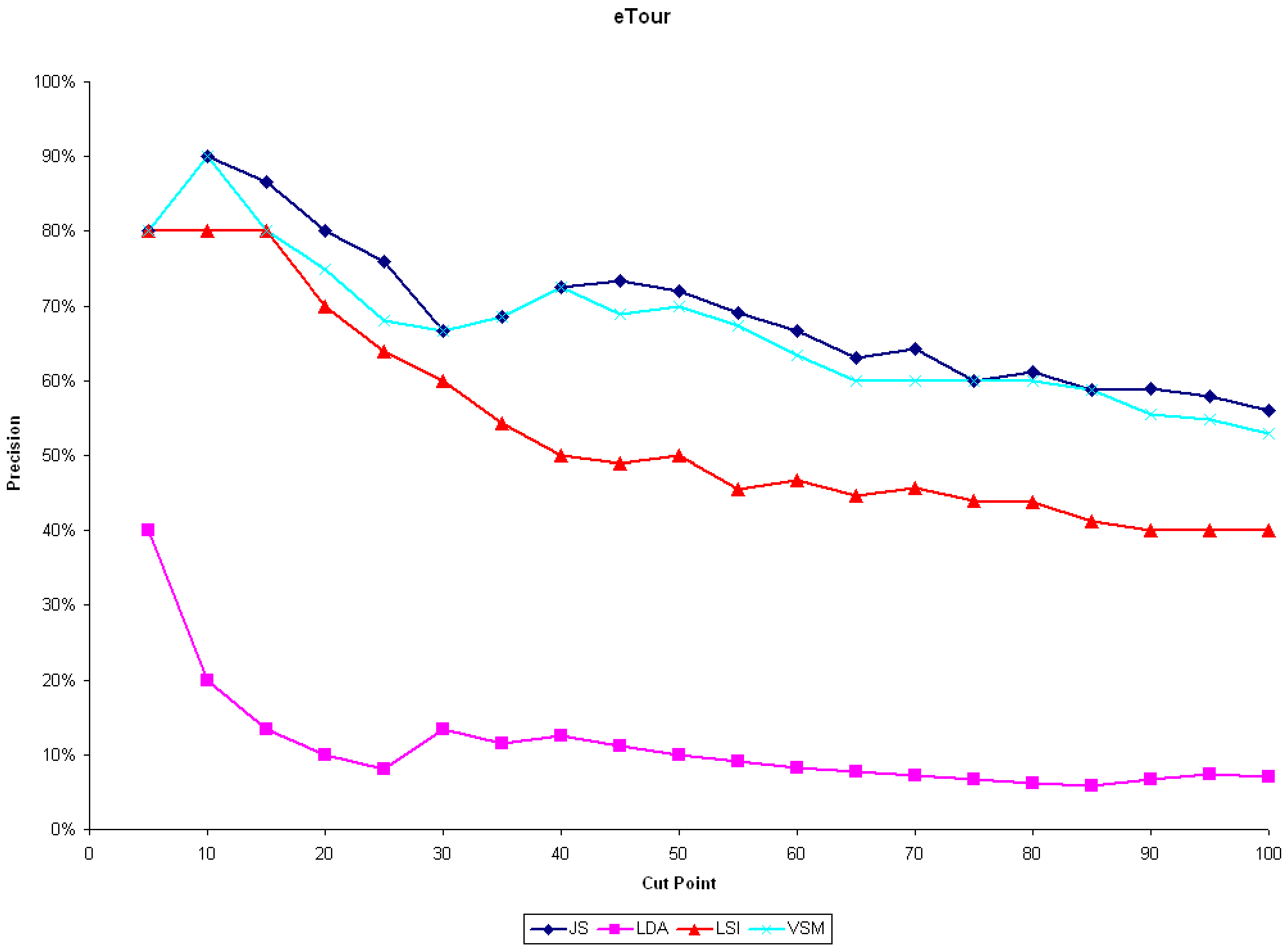
RQ$_1$ – Traceability Link Recovery Accuracy



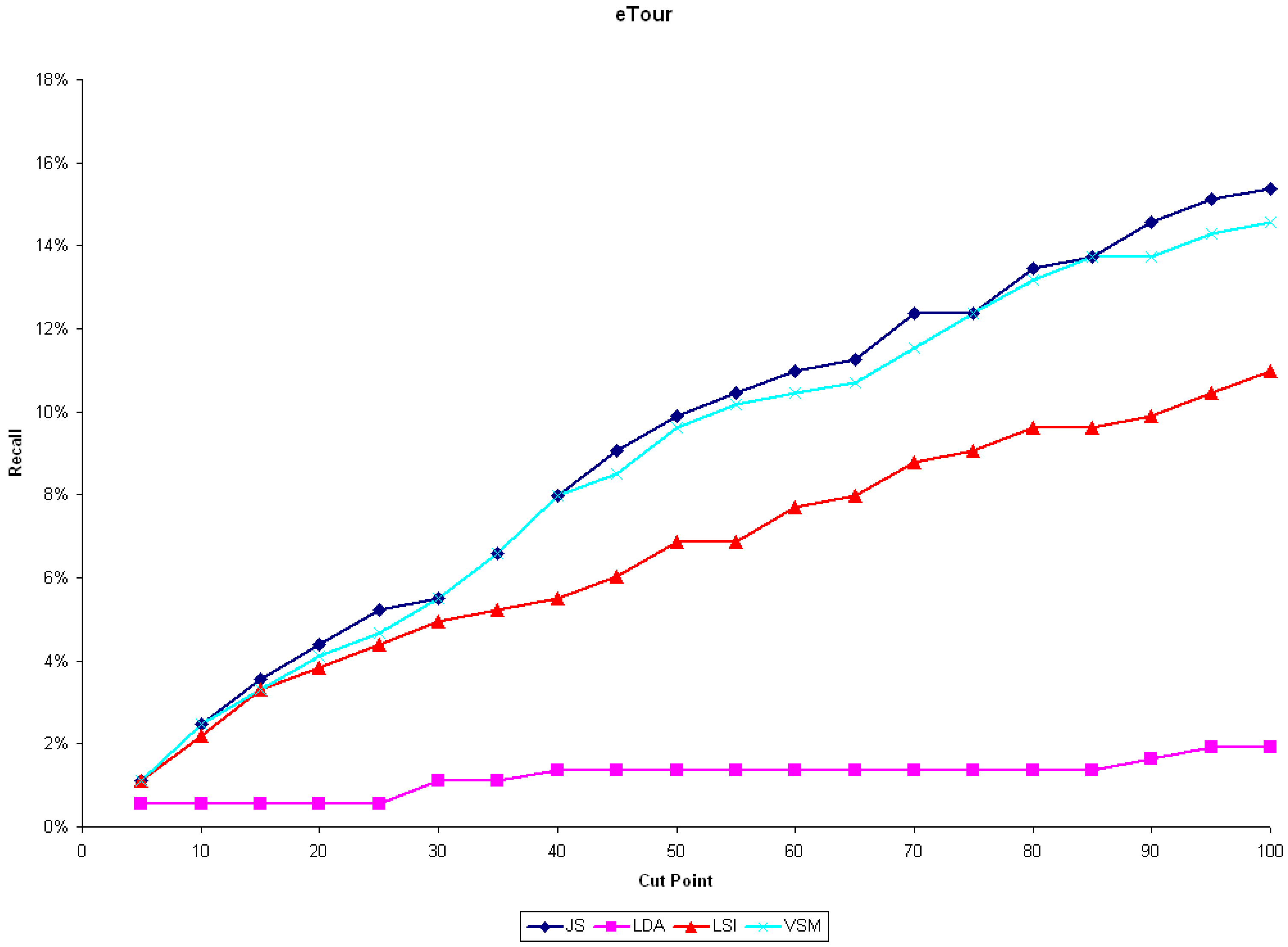
RQ_1 - Traceability Link Recovery Accuracy



RQ$_1$ - Traceability Link Recovery Accuracy



RQ_1 - Traceability Link Recovery Accuracy



RQ_2 - Principal Component Analysis (PCA)

- Do different types of IR methods provide orthogonal similarity measures?

- PCA procedure:
  - collect data
  - identify outliers
  - perform PCA

PCA Results: Rotated Components

	PC1	PC2	PC3	PC4
Proportion	73.79	25.11	0.96	0.14
Cumulative	73.79	98.9	99.86	100
JS	**0.993**	0.041	-0.101	-0.047
LDA(250)	-0.092	**0.996**	0.017	-0.004
LSI	**0.986**	-0.046	0.158	-0.01
VSM	**0.992**	0.097	-0.055	0.057

RQ_2 - Overlap Among Techniques

- Do different types of IR methods provide orthogonal similarity measures?
- Overlap Metrics

$$correct_{m_i \cap m_j} = \frac{correct_{m_i \cap m_j}}{correct_{m_i \cup m_j}} \%$$

$$correct_{m_i \setminus m_j} = \frac{correct_{m_i \setminus m_j}}{correct_{m_i \cup m_j}} \%$$

Results for Overlap Metrics for eTour

	25	50	75	100	300	500	700	1K
correct LDA\JS	0%	5%	4%	5%	9%	19%	25%	27%
correct LDAnJS	10%	8%	6%	7%	6%	6%	6%	8%
correct LDA\VSM	0%	5%	4%	5%	10%	17%	25%	26%
correct LDAnVSM	11%	8%	6%	7%	6%	8%	7%	9%
correct LDA\LSI	13%	11%	9%	9%	15%	22%	28%	30%
correct LDAnLSI	0%	7%	6%	7%	3%	5%	5%	7%

Results for Overlap Metrics for eTour

	25	50	75	100	300	500	700	1K
correct LDA\JS	0%	5%	4%	5%	9%	19%	25%	27%
correct LDAnJS	10%	8%	6%	7%	6%	6%	6%	8%
correct LDA\VSM	0%	5%	4%	5%	10%	17%	25%	26%
correct LDAnVSM	11%	8%	6%	7%	6%	8%	7%	9%
correct LDA\LSI	13%	11%	9%	9%	15%	22%	28%	30%
correct LDAnLSI	0%	7%	6%	7%	3%	5%	5%	7%

Results for Overlap Metrics for eTour

	25	50	75	100	300	500	700	1K
correct LDA\JS	0%	5%	4%	5%	9%	19%	25%	27%
correct LDAnJS	10%	8%	6%	7%	6%	6%	6%	8%
correct LDA\VSM	0%	5%	4%	5%	10%	17%	25%	26%
correct LDAnVSM	11%	8%	6%	7%	6%	8%	7%	9%
correct LDA\LSI	13%	11%	9%	9%	15%	22%	28%	30%
correct LDAnLSI	0%	7%	6%	7%	3%	5%	5%	7%

Work in Progress

- More software systems (currently working with six datasets)
- Traceability links among different types of artifacts (use cases, design, source code and test cases)
- Impact of the number of dimensions (LSI) and the number of topics (LDA) on performance
- Impact of keyword filtering techniques (all terms vs. nouns)
- Combinations of different IR techniques

Conclusions

- JS, VSM, LSI are able to provide almost the same information when used for documentation-to-code traceability recovery.
- LDA is able to capture some information missed by VSM, LSI, and JS when used for recovering traceability links between code and documentation.
- LDA’s performance based on Hellinger Distance similarity measure is somewhat lower as compared to JS, VSM, and LSI

Thank you. Questions?

SEMERU @ William and Mary

http://www.cs.wm.edu/semeru/

denys@cs.wm.edu