Università degli Studi del Sannio

Università degli Studi del Molise

Improving IR-based Traceability Recovery Using Smoothing Filters

Andrea De Lucia

Massimiliano Di Penta

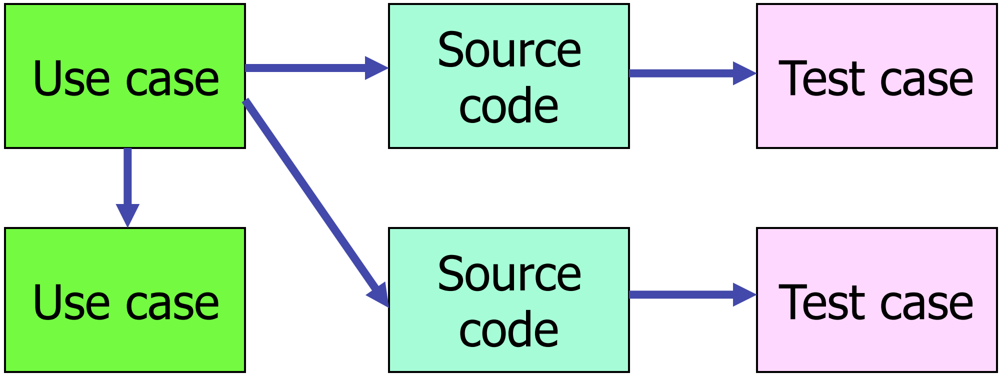
Rocco Oliveto

Annibale Panichella

Sebastiano Panichella

Software traceability

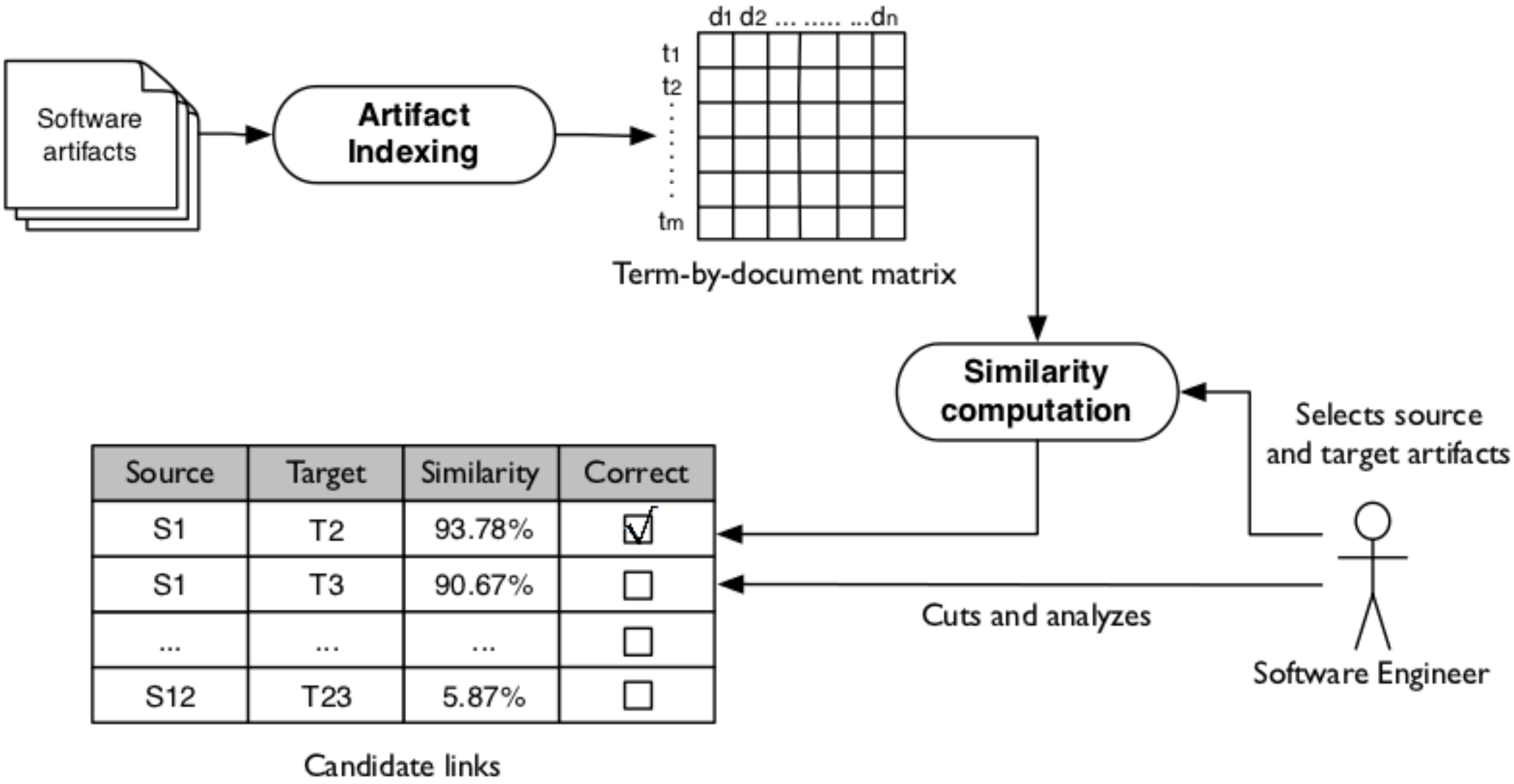
“ The degree to which a relationship can be established between two products of a software development process” [IEEE Glossary for Software Terminology]



- Important for:
  - program comprehension
  - requirement tracing
  - impact analysis
  - software reuse
  - …

Up-to-date traceability links rarely exist → need to recover them

IR-based traceability recovery



Source	Target	Similarity	Correct
S1	T2	93.78%	☑
S1	T3	90.67%	☐
...	...	...	☐
S12	T23	5.87%	☐

Candidate links

Antoniol et al., 2002 (VSM+Probabilistic model)
Marcus and Maletic, 2003 (LSI)

Traditional IR vs.
IR applied to Software Engineering

Traditional IR

- Deals with heterogeneous documents for what concerns:
  - Linguistic choices
  - Syntax
  - Semantics
- We just live with that differences

IR applied to SE

- We have sets of homogeneous documents for what concerns
  - Syntax, linguistic choices
- Examples:
  - Use cases, test documents, design documents follow a common template and contain recurrent words

Problem

- Different kinds of software artifacts require specific preprocessing

```
Test case   Change the date for a visit:

   C51      Version: 0 02 000

Use case    Satisfies the request to modify a visit
            for a patient

UcModVis

Priority    High

....

Test description

Input       Select a visit:

            26/09/2003 11:00  First visit

            Change: 03/10/2003 11:00

Oracle      Invalid sequence: The system does not allow
            to change a booking

Coverage    Valid classes: CE1  CE8  CE14  CE19  CE21

            Invalid classes: None
```

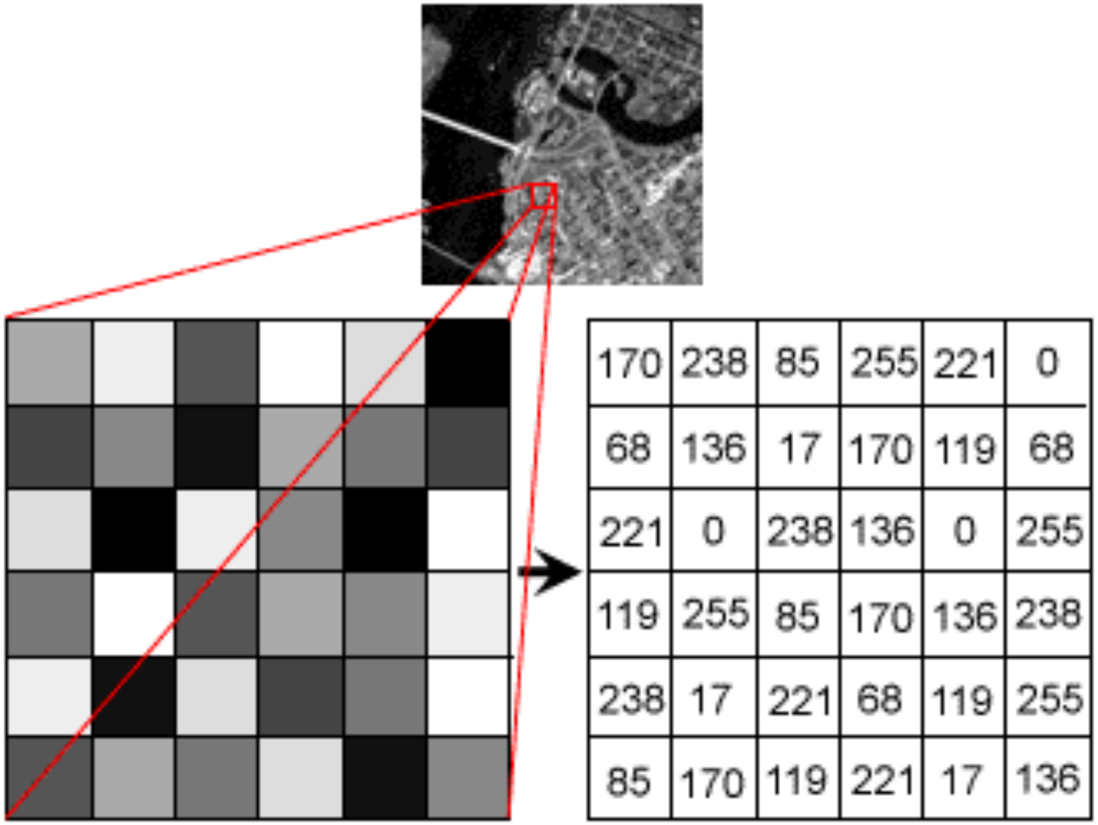
Problem

- Different kinds of software artifacts require specific preprocessing

```
Test case    Change the date for a visit:
   C51       Version: 0 02 000
Use case     Satisfies the request to modify a visit
             for a patient
UcModVis
Priority     High
....
Test description
Input        Select a visit:
             26/09/2003 11:00  First visit
             Change: 03/10/2003 11:00
Oracle       Invalid sequence: The system does not allow
             to change a booking
Coverage     Valid classes: CE1  CE8  CE14  CE19  CE21
             Invalid classes: None
```

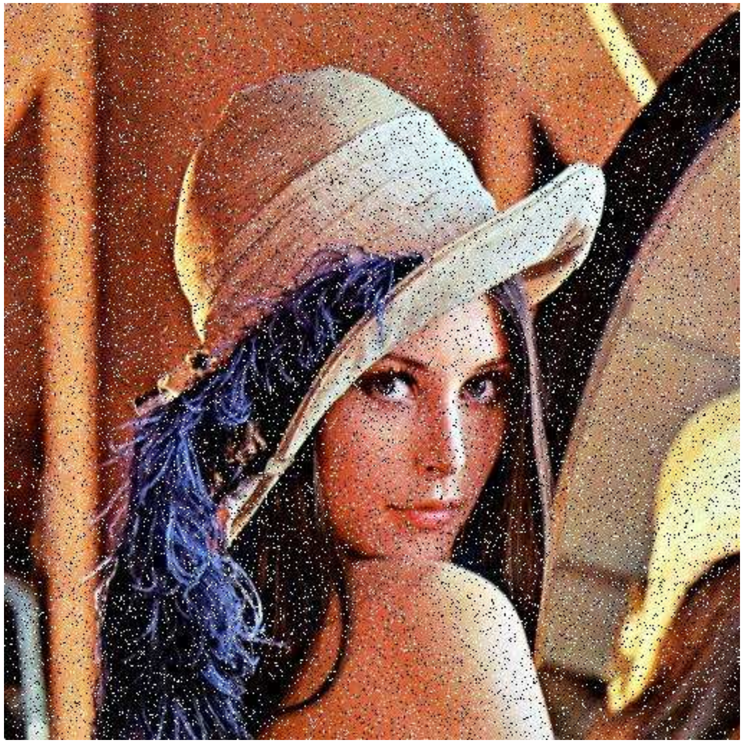
Artifact-specific words do not bring useful information

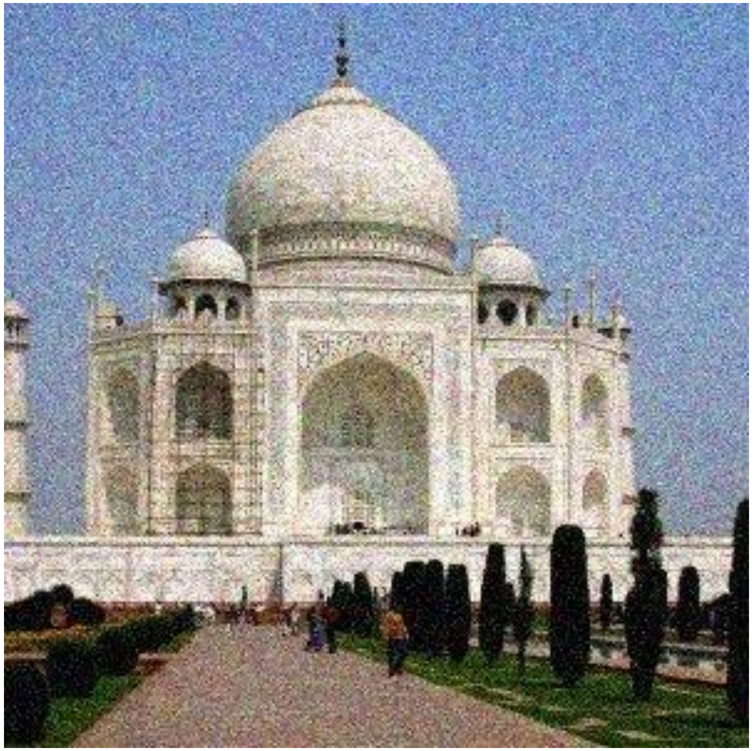
A similar problem: image processing



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

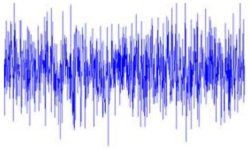
Noisy images





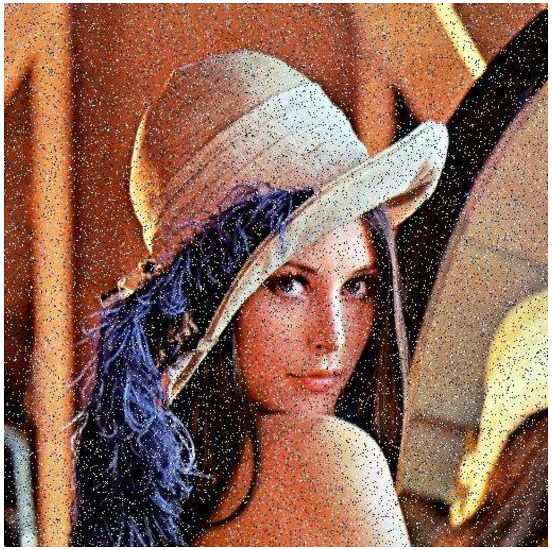
Pixels with peaks of low color intensity

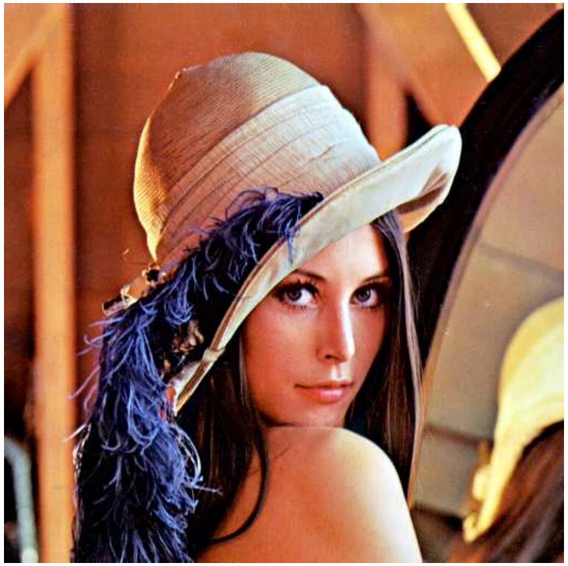
Pixels with peaks of high color intensity

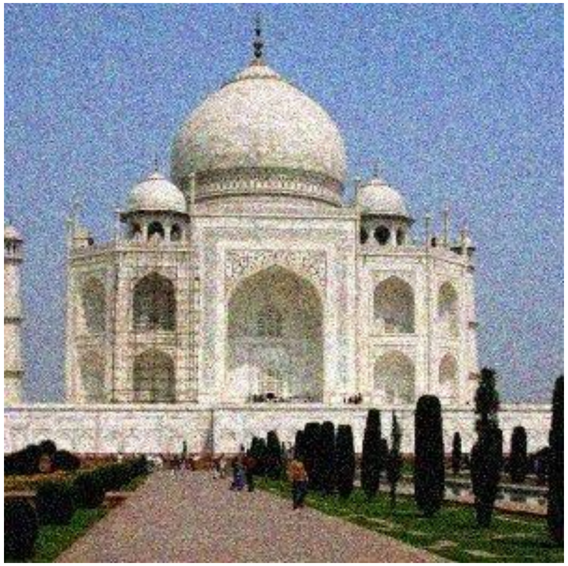


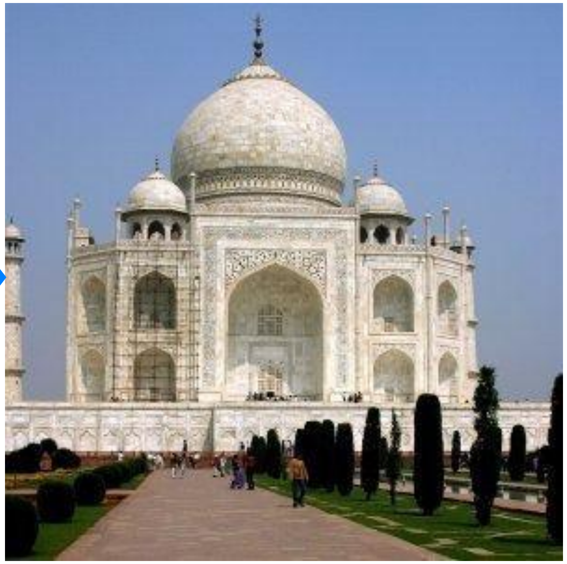
Noise

Reducing noise using smoothing filters









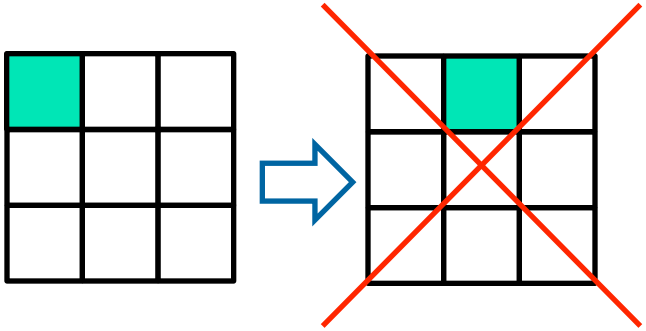
Mean filter

$$g(x,y) = \frac{1}{M}\sum_{f(n,m)\in S} f(n,m)$$

Image vs. traceability noise

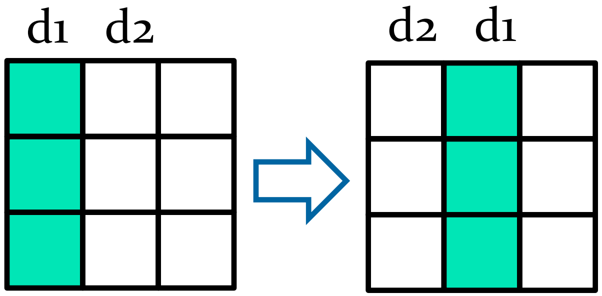
Image noise:

- Pixels with high or low color intensity
- Pixels are position dependent

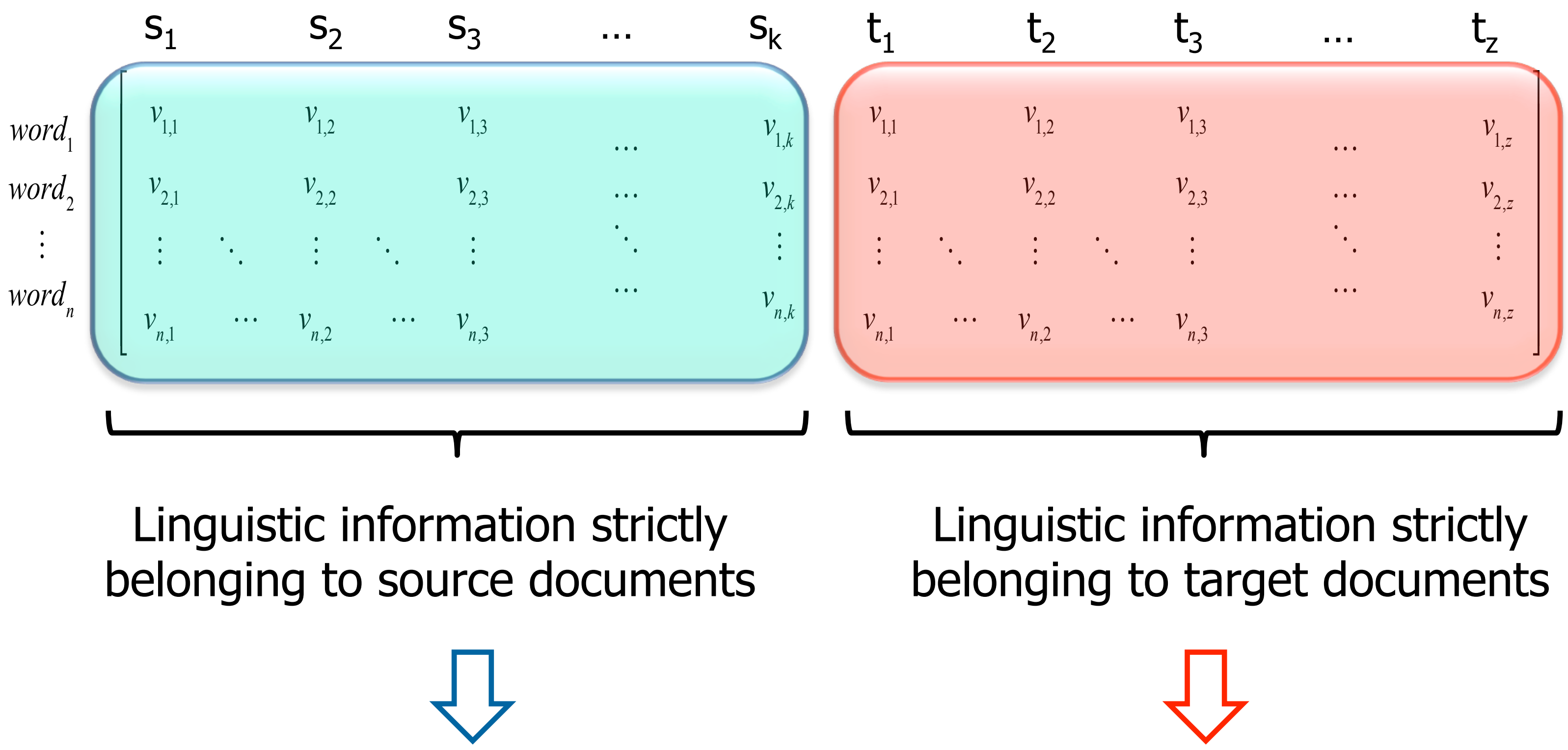


Traceability noise:

- Terms and linguistic patterns occurring in many artifacts of a given category
  - Use cases, test cases..
- Artifacts (columns) are position independent



Representing the noise



Source Documents

$$\begin{array}{c|ccccc} & s_1 & s_2 & s_3 & \cdots & s_k \\ \hline word_1 & v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,k} \\ word_2 & v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,k} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ word_n & v_{n,1} & v_{n,2} & v_{n,3} & \cdots & v_{n,k} \end{array}$$

Linguistic information strictly belonging to source documents

Common Information for source documents

Target Documents

$$\begin{array}{ccccc} t_1 & t_2 & t_3 & \cdots & t_z \\ \hline v_{1,1} & v_{1,2} & v_{1,3} & \cdots & v_{1,z} \\ v_{2,1} & v_{2,2} & v_{2,3} & \cdots & v_{2,z} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{n,1} & v_{n,2} & v_{n,3} & \cdots & v_{n,z} \end{array}$$

Linguistic information strictly belonging to target documents

Common Information For target documents

Representing the noise

Source Documents **Target Documents**

$$\begin{array}{c|ccccccc|ccccccc}
 & s_1 & & s_2 & & s_3 & \ldots & s_k & t_1 & & t_2 & & t_3 & \ldots & t_z \\
\hline
word_1 & v_{1,1} & & v_{1,2} & & v_{1,3} & \cdots & v_{1,k} & v_{1,1} & & v_{1,2} & & v_{1,3} & \cdots & v_{1,z} \\
word_2 & v_{2,1} & & v_{2,2} & & v_{2,3} & \cdots & v_{2,k} & v_{2,1} & & v_{2,2} & & v_{2,3} & \cdots & v_{2,z} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
word_n & v_{n,1} & \cdots & v_{n,2} & \cdots & v_{n,3} & \cdots & v_{n,k} & v_{n,1} & \cdots & v_{n,2} & \cdots & v_{n,3} & \cdots & v_{n,z}
\end{array}$$

Mean source vector $S=\begin{bmatrix} \frac{1}{k}\sum_{j=1}^{k} v_{1,j} \\ \frac{1}{k}\sum_{j=1}^{k} v_{2,j} \\ \vdots \\ \frac{1}{k}\sum_{j=1}^{k} v_{n,j} \end{bmatrix}$

⇩

Common Information
for source documents

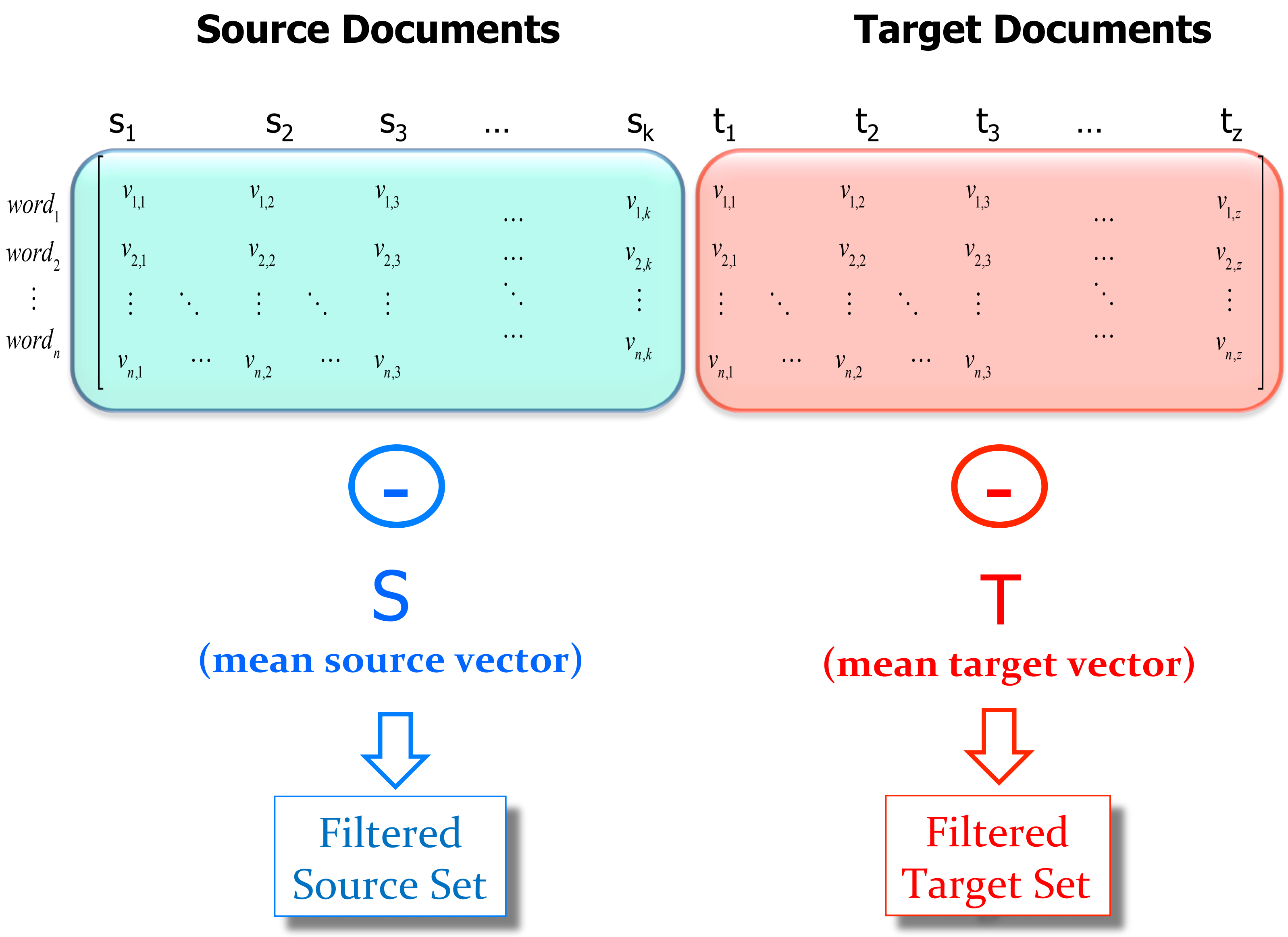
Mean target vector $T=\begin{bmatrix} \frac{1}{z}\sum_{j=k+1}^{m} v_{1,j} \\ \frac{1}{z}\sum_{j=k+1}^{m} v_{2,j} \\ \vdots \\ \frac{1}{z}\sum_{j=k+1}^{m} v_{n,j} \end{bmatrix}$

⇩

Common Information
for target documents

The Mean vectors are like the continuous component of a signal…

Representing the noise



Source Documents

$$\begin{array}{c|cccccccc} & s_1 & & s_2 & & s_3 & \ldots & s_k \\ \hline word_1 & v_{1,1} & & v_{1,2} & & v_{1,3} & \cdots & v_{1,k} \\ word_2 & v_{2,1} & & v_{2,2} & & v_{2,3} & \cdots & v_{2,k} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ word_n & v_{n,1} & \cdots & v_{n,2} & \cdots & v_{n,3} & \cdots & v_{n,k} \end{array}$$

-

S

(mean source vector)

Filtered Source Set

Target Documents

$$\begin{array}{cccccccc} t_1 & & t_2 & & t_3 & \ldots & t_z \\ \hline v_{1,1} & & v_{1,2} & & v_{1,3} & \cdots & v_{1,z} \\ v_{2,1} & & v_{2,2} & & v_{2,3} & \cdots & v_{2,z} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{n,1} & \cdots & v_{n,2} & \cdots & v_{n,3} & \cdots & v_{n,z} \end{array}$$

-

T

(mean target vector)

Filtered Target Set

Empirical Study

- Goal: analyze the effect of smoothing filter
- Purpose: investigating how the filter affects traceability recovery
- Quality focus: traceability recovery performance
- Perspective:
  - Researchers: evaluating the novel technique
- Context: artifacts from two systems
  - EasyClinic and Pine

Context

	EasyClinic	Pine
Description	Medical doctor office management	Text-based email client
Language	Java	C
Files/Classes	37	31
KLOC	20	130
Documents	113	100
Language	Italian	English
Artifacts	Use cases Interaction diagrams Source code Test cases	Requirements Use cases

Research Questions and Factors

- RQ1: Does the smoothing filter improve the recovery performances of traceability recovery?
- RQ2: How effective is the smoothing filter in filtering out non-relevant words, as compared to stop word removal?

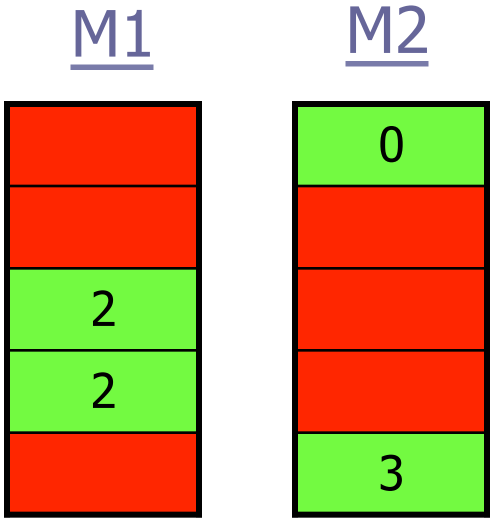
- Factors:
  - Use of filter: YES, NO
  - Technique: VSM, LSI
  - Artifact: Req., UC, Int. Diagrams, Code, TC
  - System: Easyclinic, Pine

Analysis Method – RQ1

- Performances evaluated by precision and recall:

$$precision = \frac{|correct \cap retrieved|}{|retrieved|} \qquad recall = \frac{|correct \cap retrieved|}{|correct|}$$

- We statistically compare the # of false positives of different methods for each correct link identified
  - Wilcoxon Rank Sum test
  - Cliff' s delta effect size



Analysis Method – RQ2

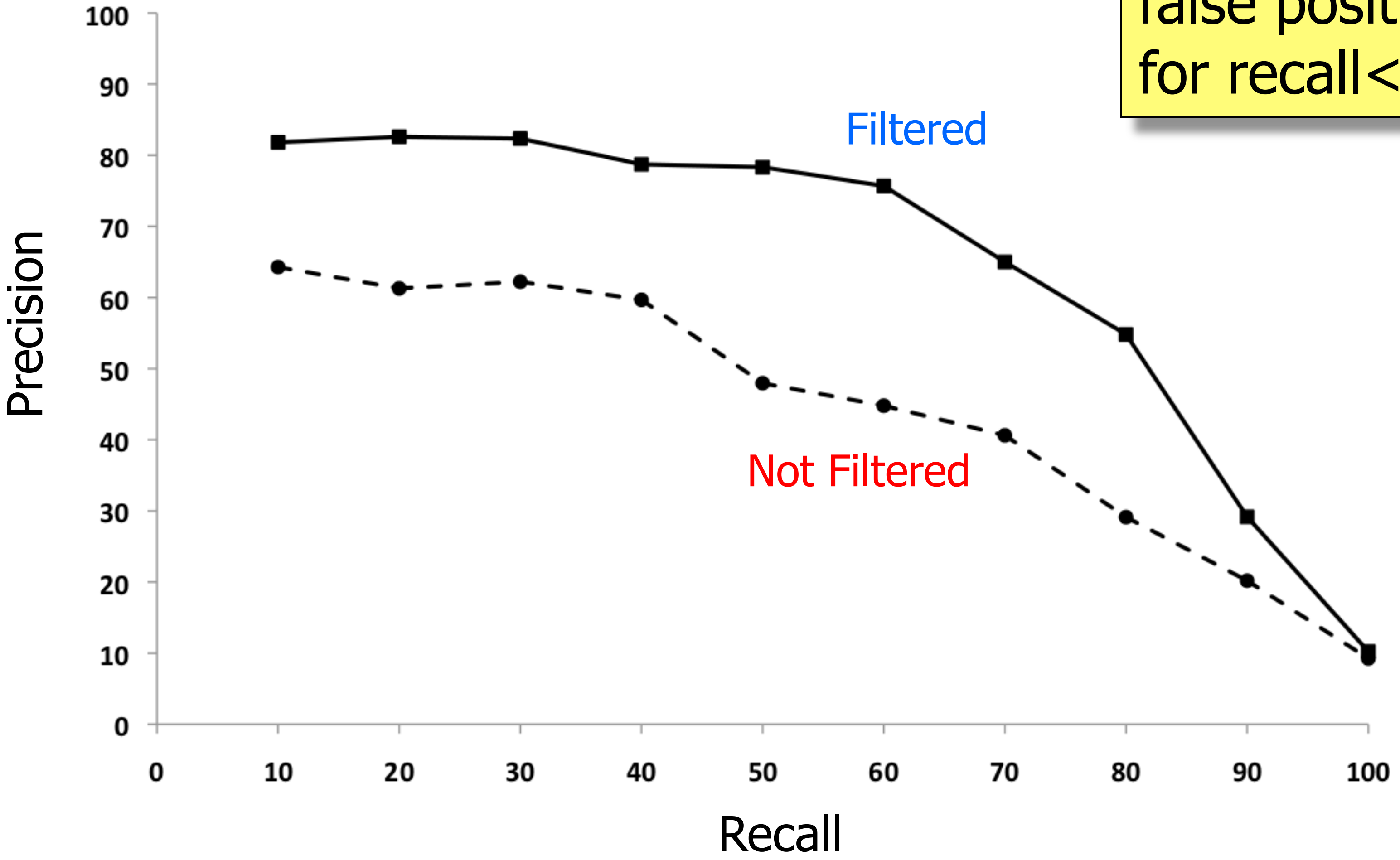
- We replace stop word filtering by one of the following treatments:
  1. Standard stop word removal
  2. Manually customized stop word removal
  3. Smoothing filter
  4. Standard stop word removal + filter
  5. Customized stop word removal + filter

- …and compare the performances

Results

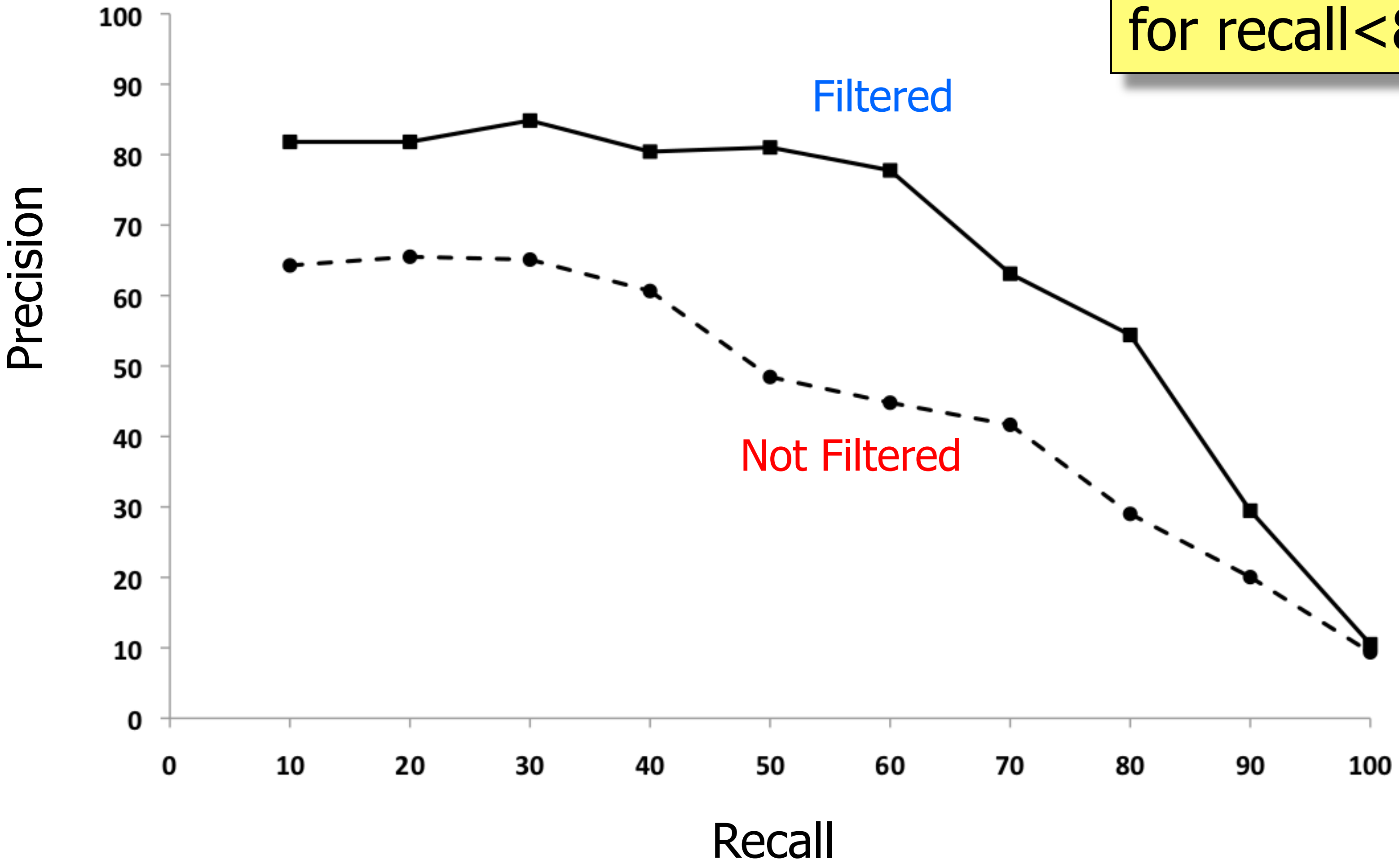
EasyClinic: Use cases into source (VSM)

[-60, -74]% of false positives for recall<80%



EasyClinic: Use cases into source (LSI)

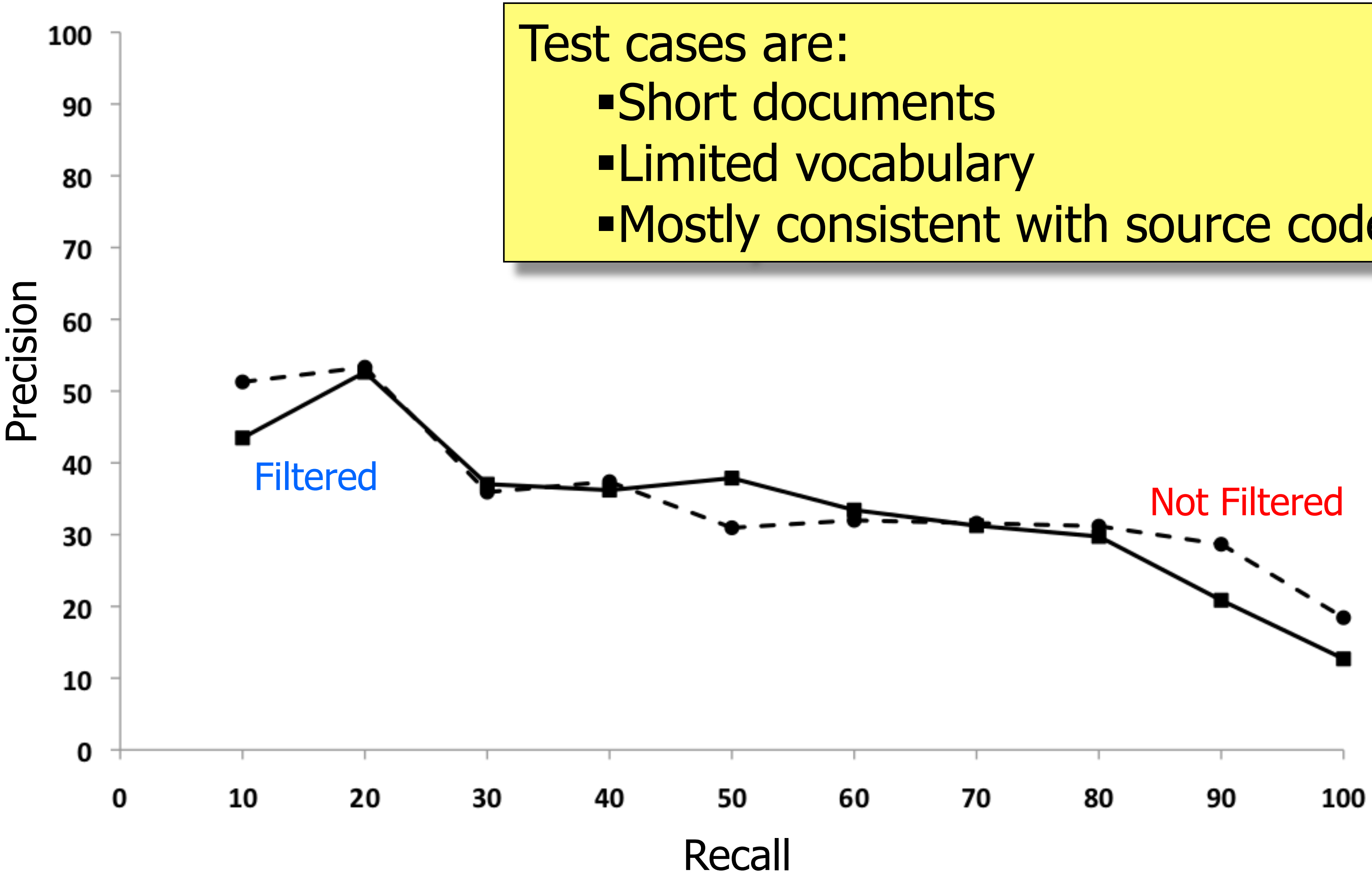
[-60, -77]% of false positives for recall<80%



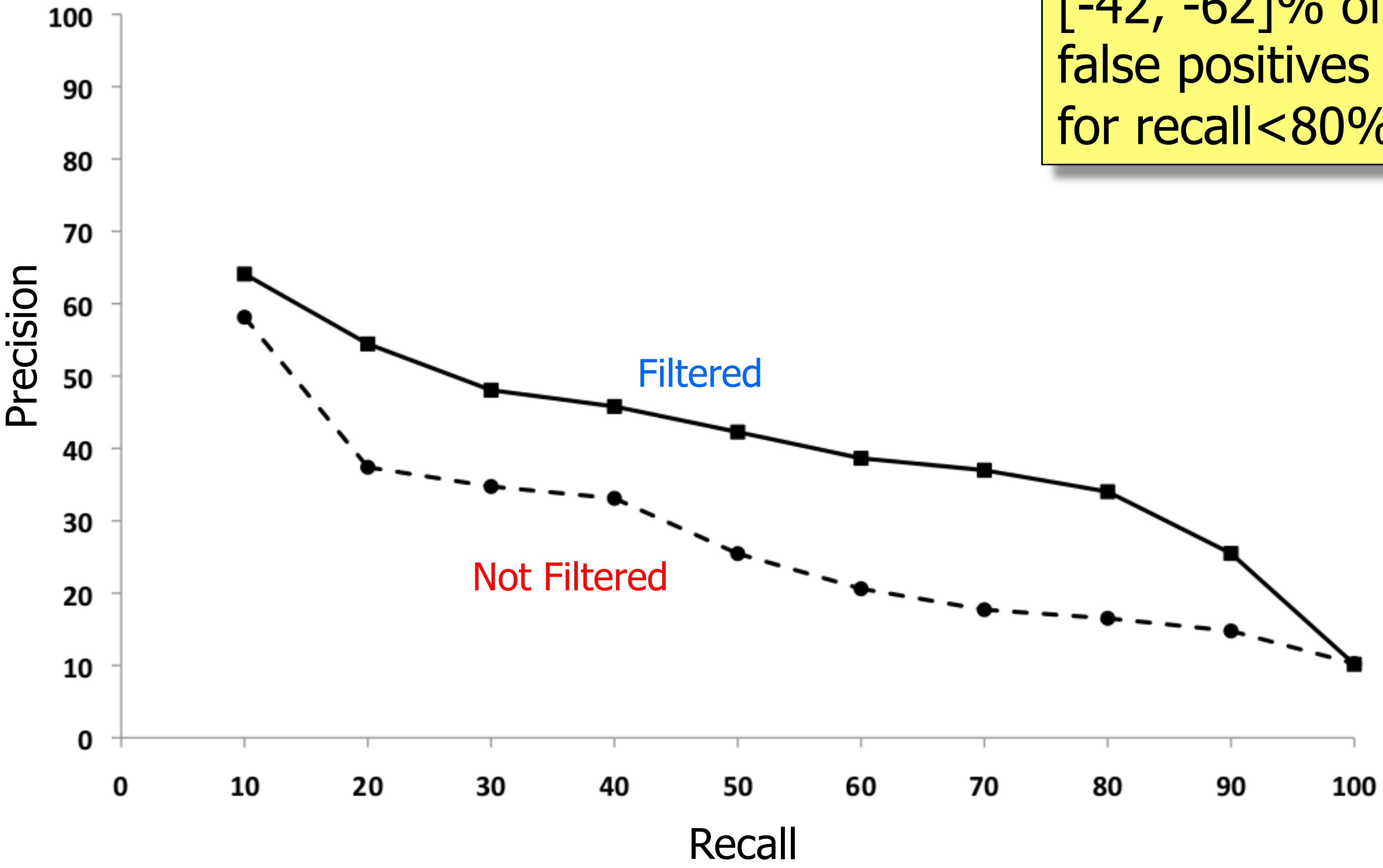
EasyClinic: Test cases into source (LSI)

Test cases are:

- Short documents
- Limited vocabulary
- Mostly consistent with source code



Pine: Use cases into requirements (LSI)



[-42, -62]% of false positives for recall<80%

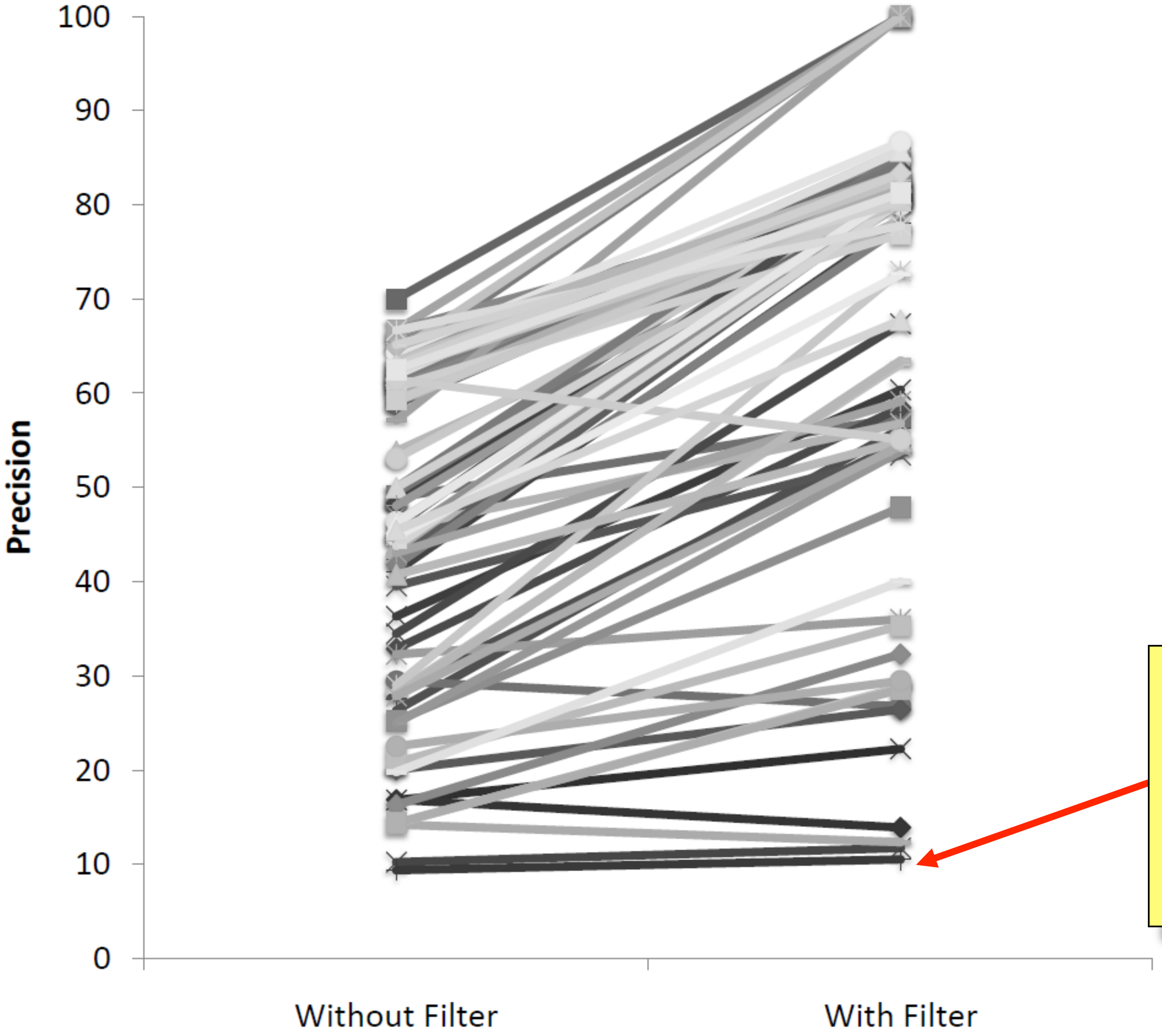
Statistical Comparison

Data set	Traced Artifacts	VSM		LSI	
		p-value	Effect size	p-value	Effect size
EasyClinic	UC→Code	**<0.01**	0.50 (**large**)	**<0.01**	0.50 (**large**)
	Int. Diag. → Code	**<0.01**	0.52 (**large**)	**<0.01**	0.34 (**medium**)
	TC → Code	1.00	- (**negligible**)	1.00	- (**negligible**)
Pine	Req. → UC	**<0.01**	0.58 (**large**)	**<0.01**	0.58 (**large**)

RQ2 – Summary of results

Comparison		EasyClinic			Pine
		UC→CC	**ID→CC**	**TC→CC**	**HLR→ UC**
Smoothing filter	Standard list	YES (small)	YES (small)	NO (large)	YES (large)
Smoothing filter	Cust. list	YES (small)	YES (small)	NO (large)	YES (large)
Standard list+ Smoothing filter	Cust. list	YES (large)	YES (large)	NO (medium)	YES (large)
Standard list+ Smoothing filter	Cist list + Smoothing filter	NO (small)	-	YES (medium)	YES (small)

Link precision improvement



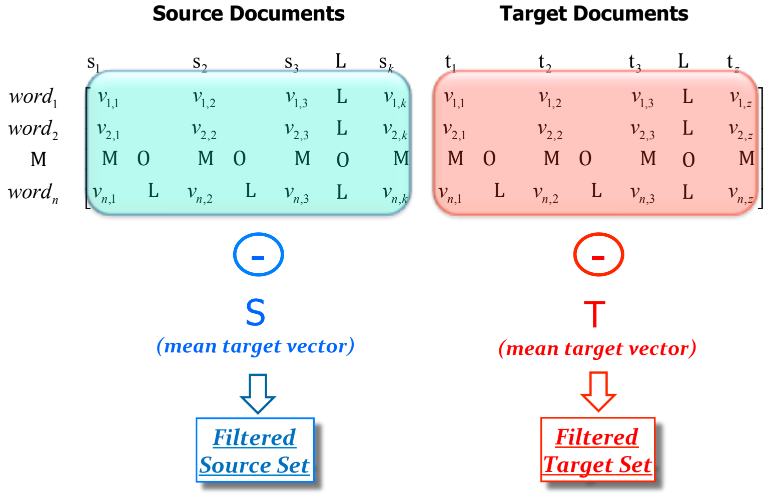
Login Patient vs. Person
Poor vocabulary overlap (10%)

Threats to validity

- Construct validity
  - Mainly related to our oracle
  - Provided by developers and for EasyClinic also peer-reviewed
- Internal validity
  - Improvements could be due to other reasons…
  - However, we compared different techniques (VSM, LSI)
  - The approach works well regardless of stop word removal, stemming and use of tf-idf
- Conclusion validity
  - Conclusions based on proper (non-parametric) statistics
- External validity
  - We considered systems with different characteristics and artifacts
  - … but further studies are desirable

Conclusions

Representing the noise



Source Documents

$$\begin{array}{c|ccccc} & s_1 & s_2 & s_3 & L & s_k \\ \hline word_1 & v_{1,1} & v_{1,2} & v_{1,3} & L & v_{1,k} \\ word_2 & v_{2,1} & v_{2,2} & v_{2,3} & L & v_{2,k} \\ M & M \quad O & M \quad O & M \quad O & & M \\ word_n & v_{n,1} \quad L & v_{n,2} \quad L & v_{n,3} & L & v_{n,k} \end{array}$$

Target Documents

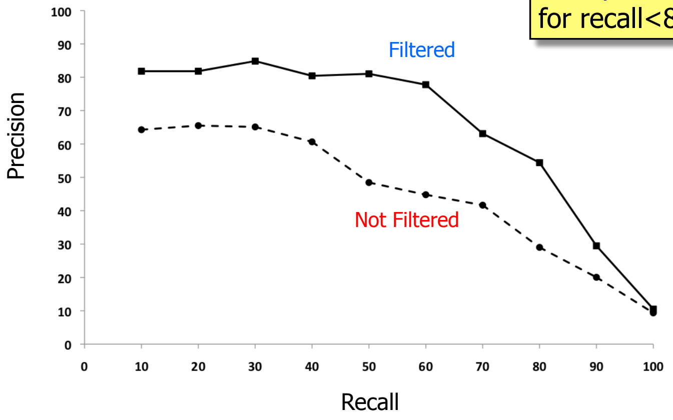
$$\begin{array}{ccccc} t_1 & t_2 & t_3 & L & t_z \\ \hline v_{1,1} & v_{1,2} & v_{1,3} & L & v_{1,z} \\ v_{2,1} & v_{2,2} & v_{2,3} & L & v_{2,z} \\ M \quad O & M \quad O & M \quad O & & M \\ v_{n,1} \quad L & v_{n,2} \quad L & v_{n,3} & L & v_{n,z} \end{array}$$

- S *(mean target vector)* ⇩ ***Filtered Source Set***

- T *(mean target vector)* ⇩ ***Filtered Target Set***

EasyClinic: Use cases into source (LSI)

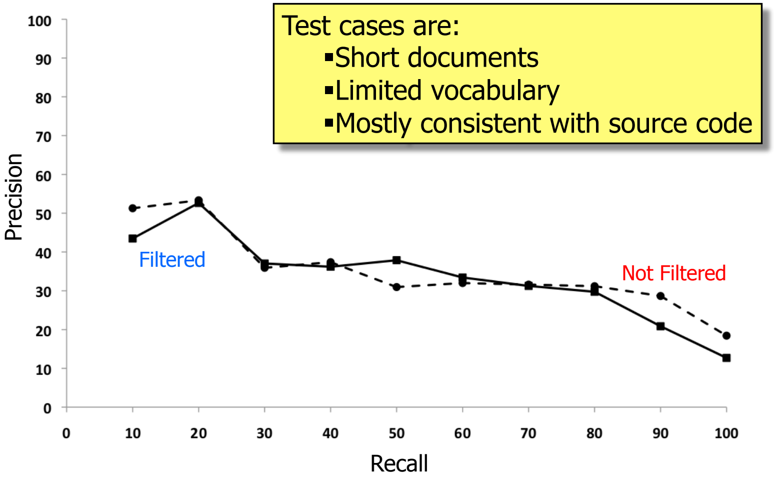
[-60, -77]% of false positives for recall<80%



EasyClinic: Test cases into source (LSI)

Test cases are:

- Short documents
- Limited vocabulary
- Mostly consistent with source code



RQ2 – Summary of results

Comparison		**EasyClinic**			**Pine**
		UC→CC	**ID→CC**	**TC→CC**	**HLR→ UC**
Smoothing filter	Standard list	YES (small)	YES (small)	NO (large)	YES (large)
Smoothing filter	Cust. list	YES (small)	YES (small)	NO (large)	YES (large)
Standard list+ Smoothing filter	Cust. list	YES (large)	YES (large)	NO (medium)	YES (large)
Standard list+ Smoothing filter	Cist list + Smoothing filter	NO (small)	-	YES (medium)	YES (small)

Work-in-progress

- Study replication
  - Different systems and artifacts
  - Use of relevance feedback
- More sophisticated smoothing technique
  - Non-linear filters
- Use in other applications of IR to software engineering
  - impact analysis
  - feature location