On Integrating Orthogonal Information Retrieval Methods to Improve Traceability Recovery

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

Traceability is...

USE CASE

SOURCE CODE

DIAGRAM

TEST CASE

"the ability to describe and follow the artifact life-cycle"

Traceability in practice

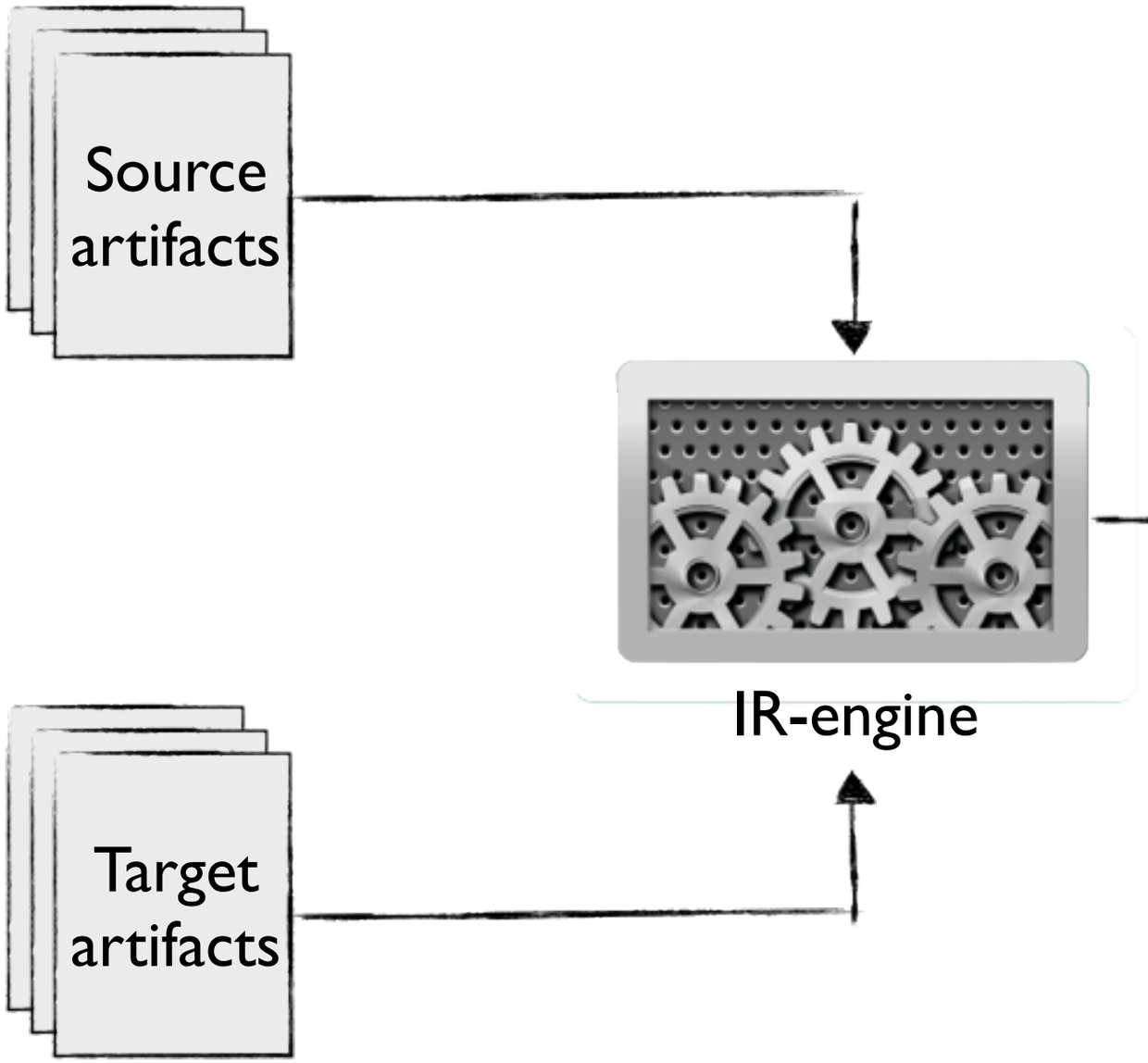
USE CASE

SOURCE CODE

DIAGRAM

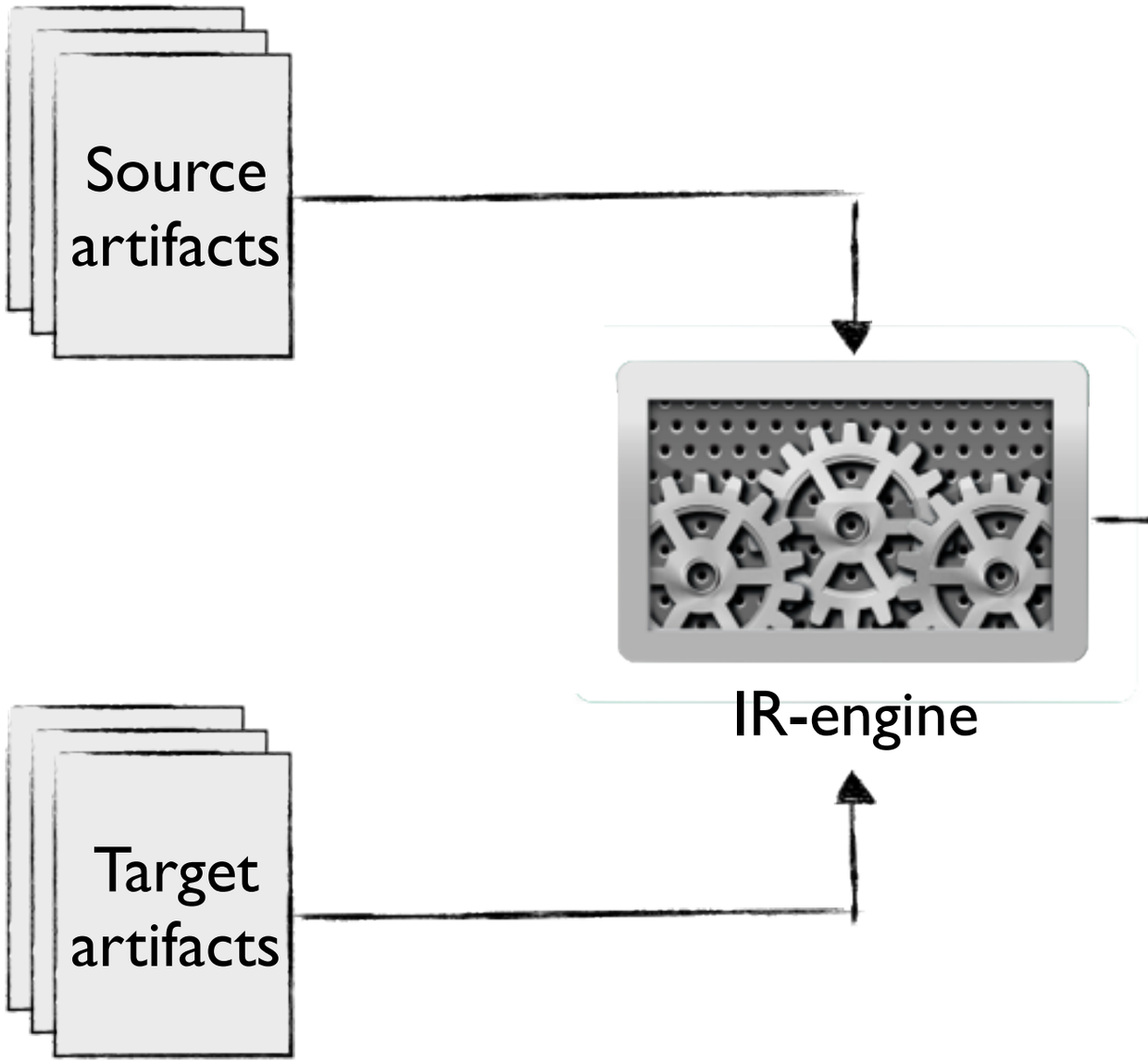
TEST CASE

Traceability information is still not commonplace in software projects!



Source	Target	Score
UC1	TC2	97%
UC1	TC1	91%
UC3	TC3	82%
...	...	...
...	...	...
UC9	TC12	8%

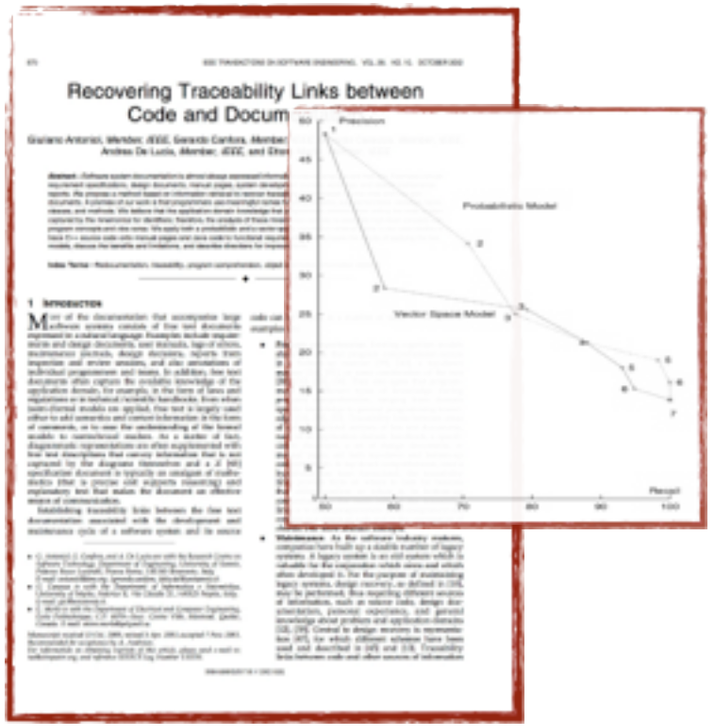
IR-based traceability recovery process



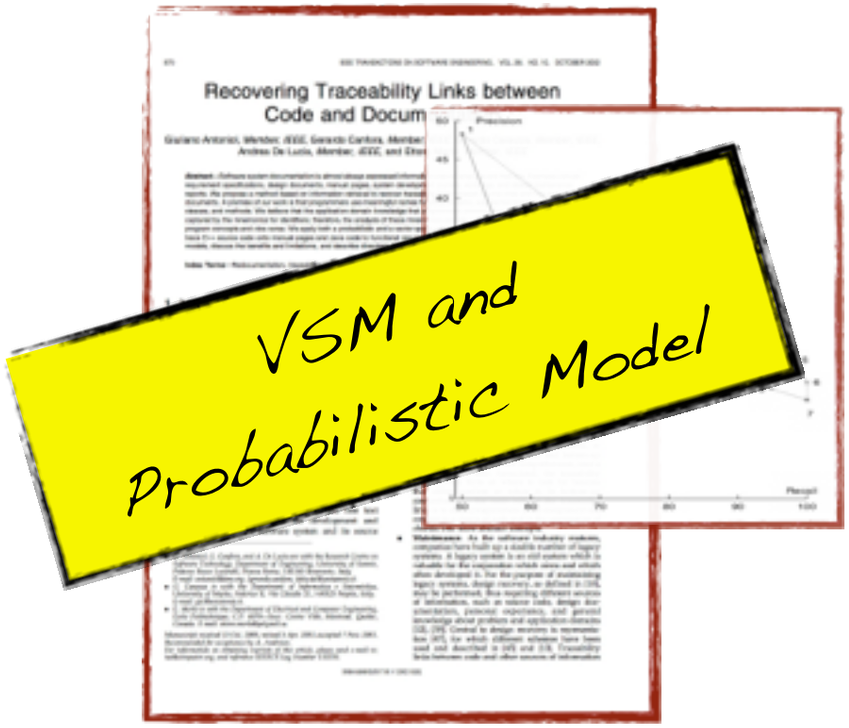
Source	Target	Score
UC1	TC2	97%
UC1	TC1	91%
UC3	TC3	82%
...	...	...
...	...	...
UC9	TC12	8%

IR-based traceability recovery process

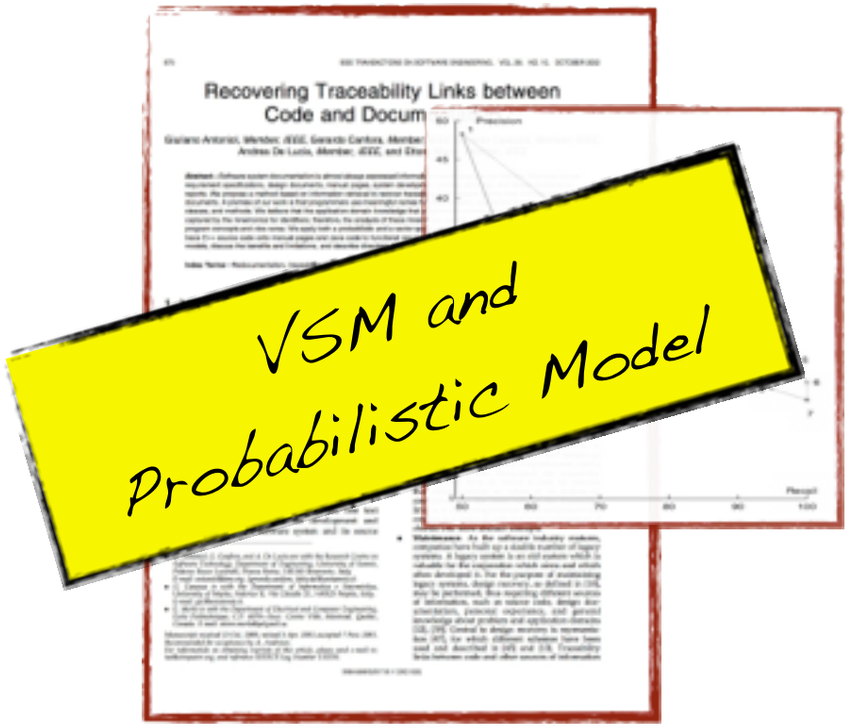
State of the art

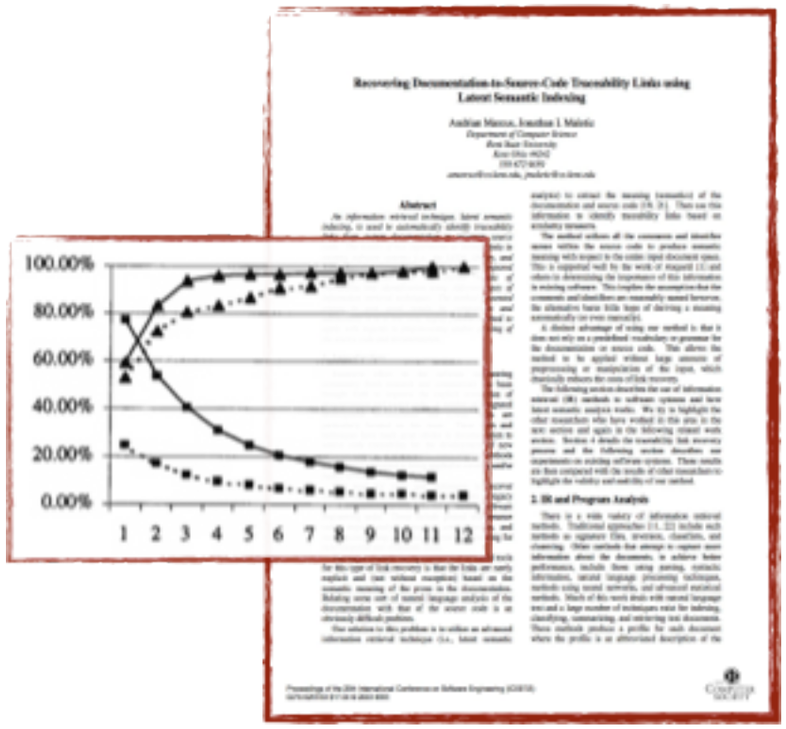


State of the art

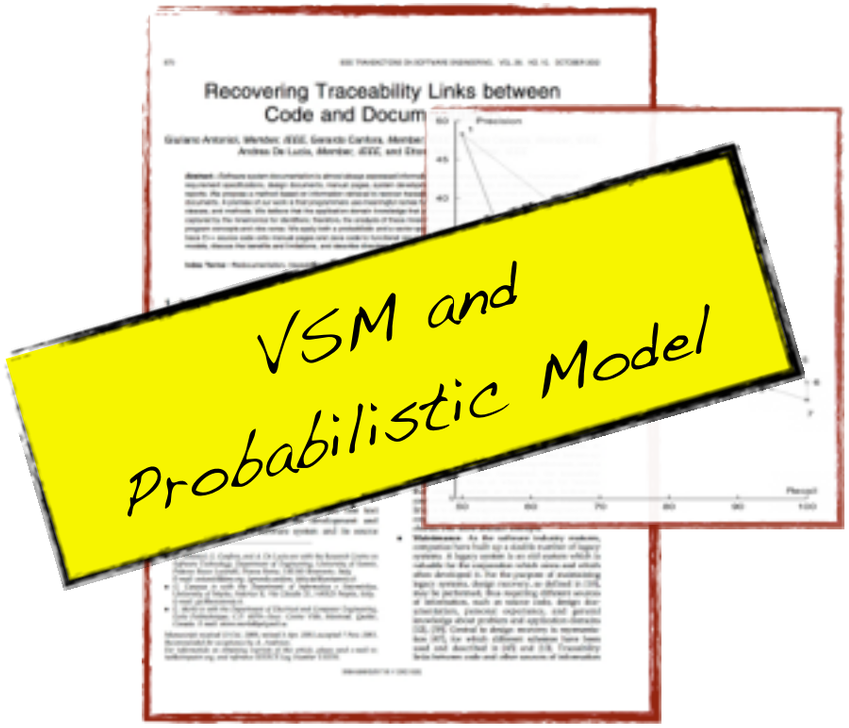


State of the art

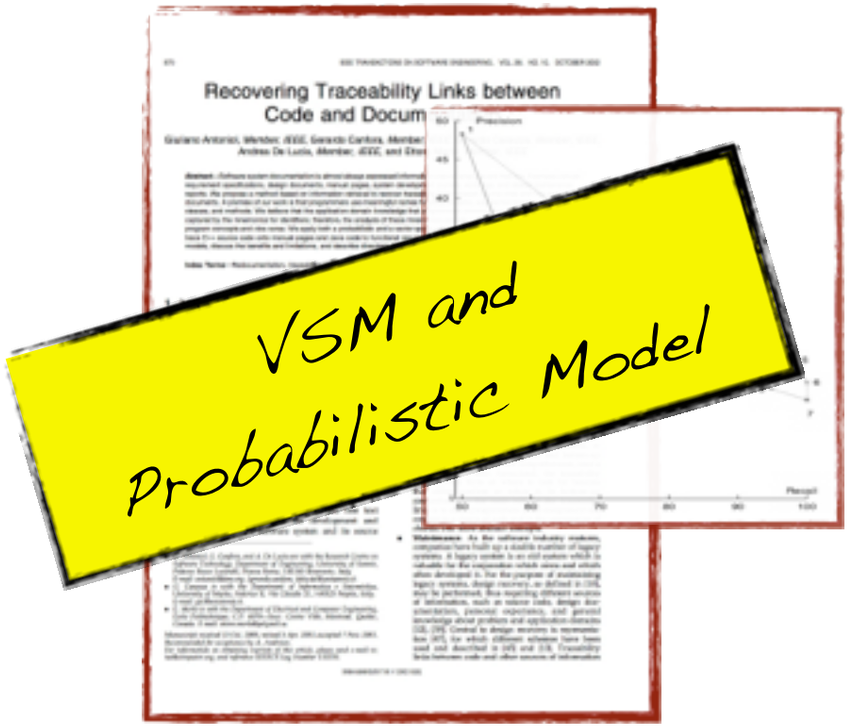


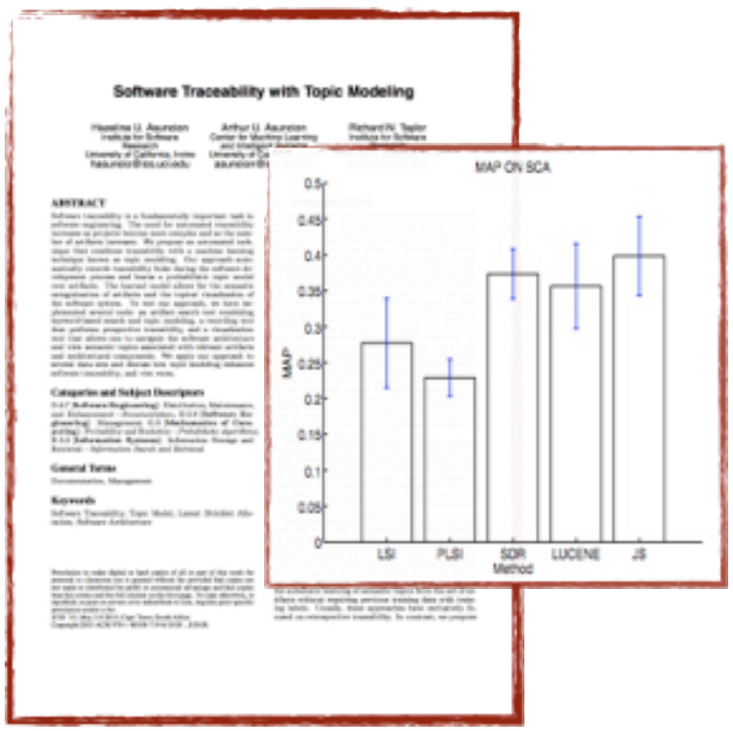


State of the art

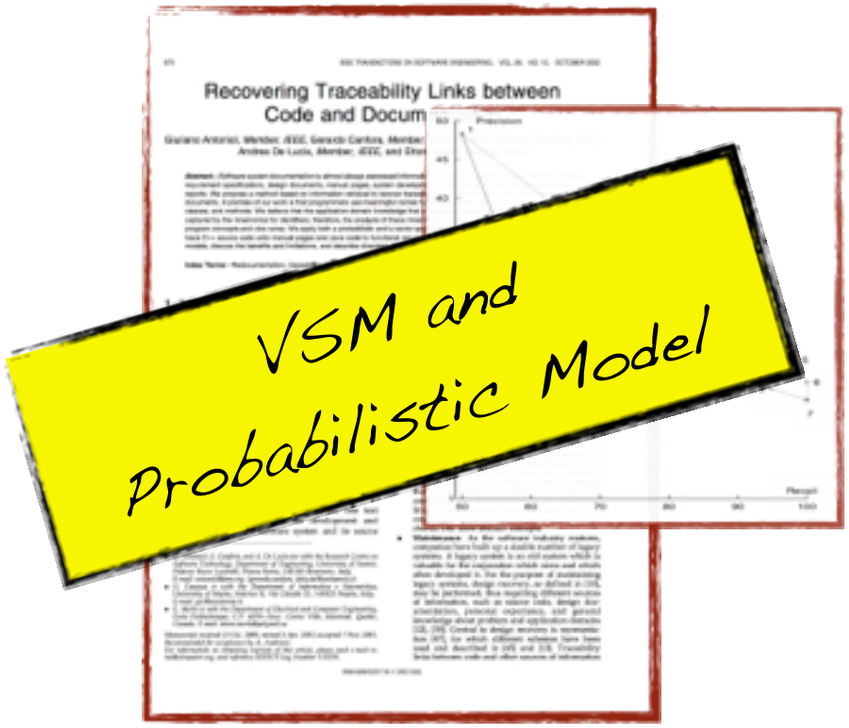


State of the art



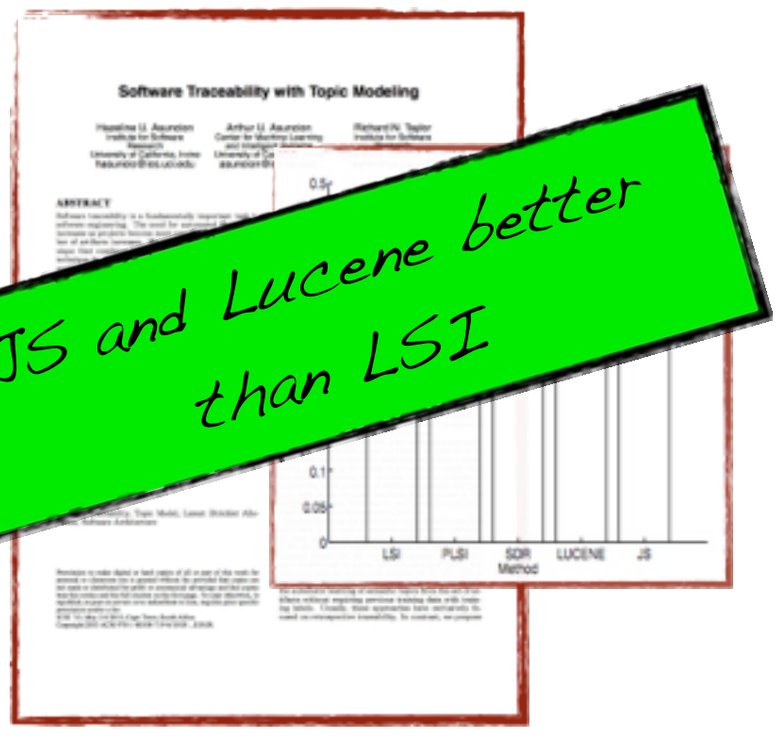


State of the art



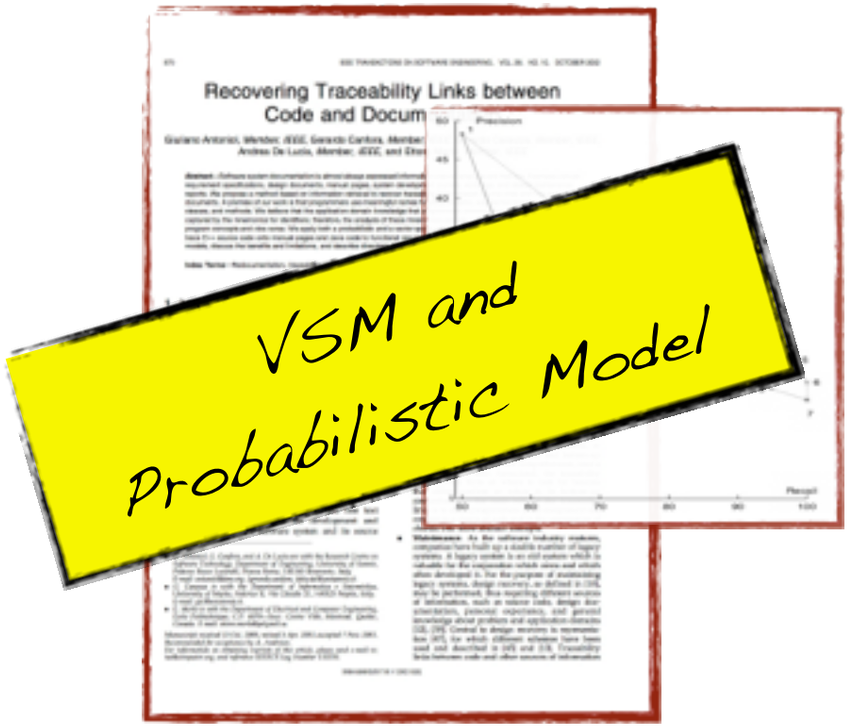
VSM and Probabilistic Model

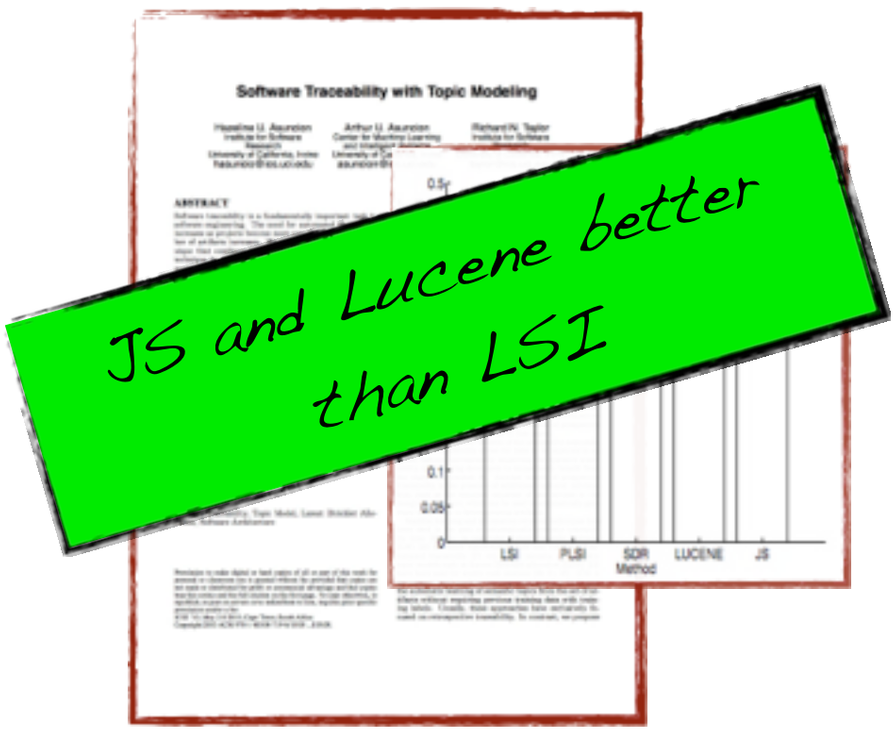
LSI better than VSM and Probabilistic Model

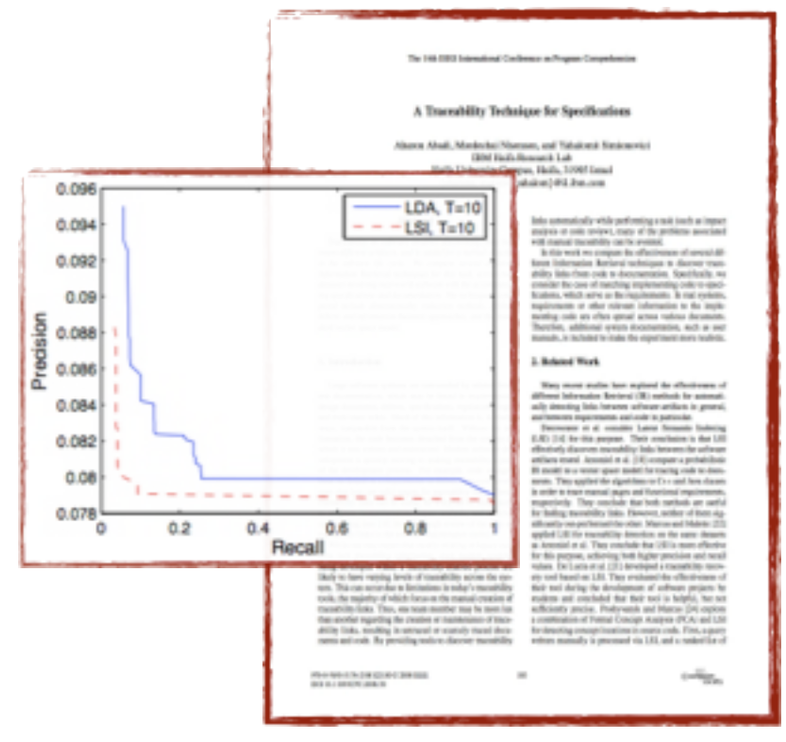


JS and Lucene better than LSI

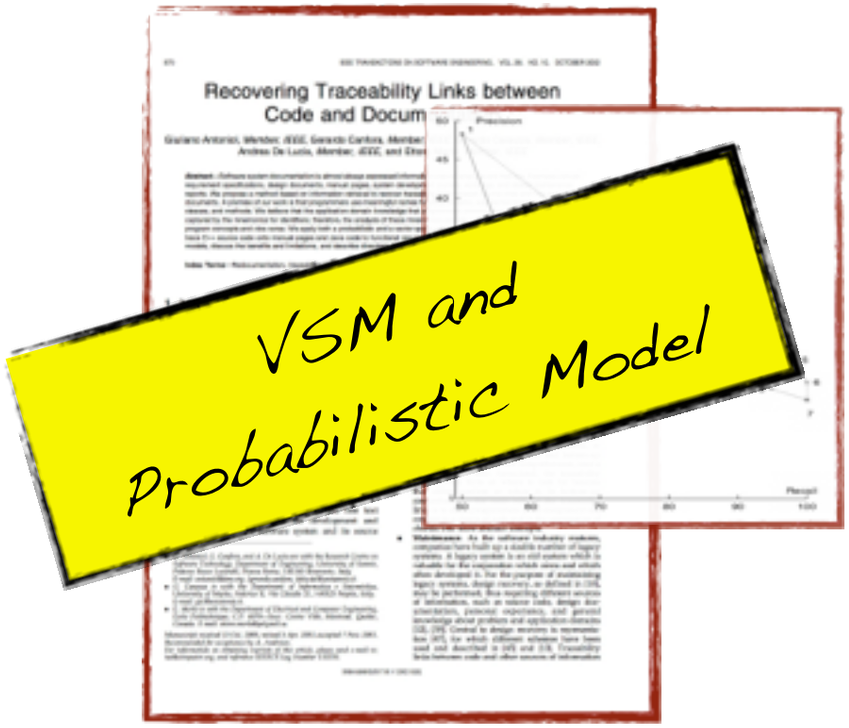
State of the art

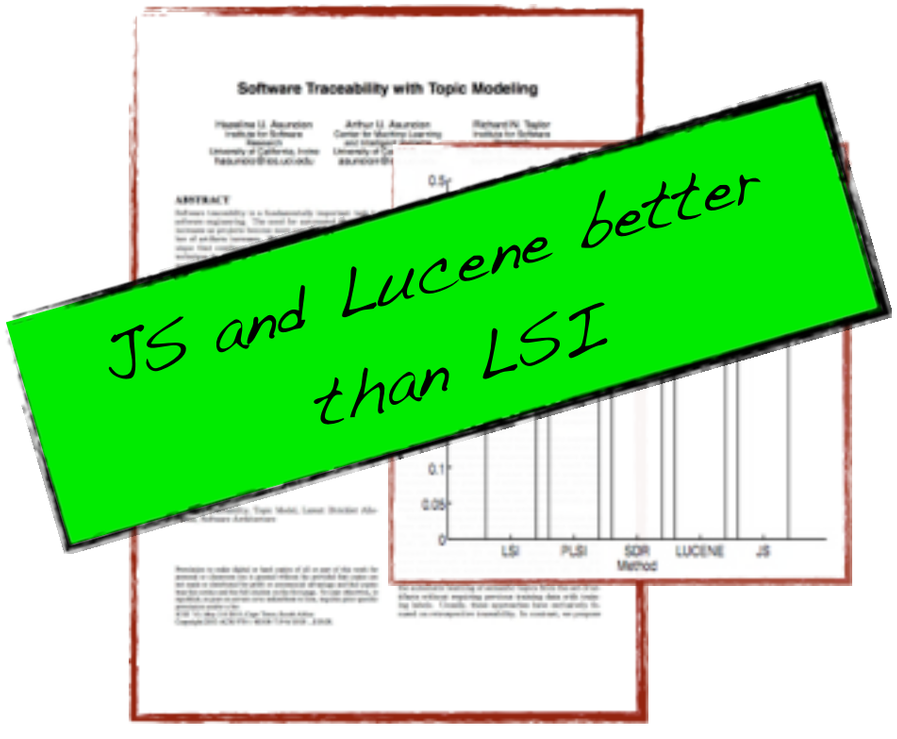


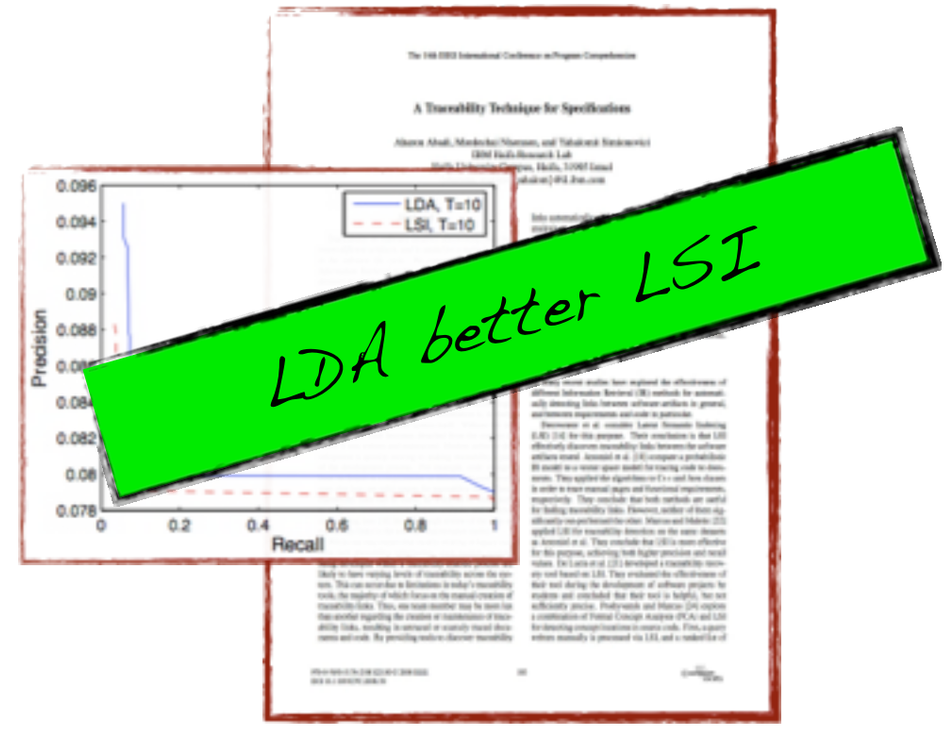




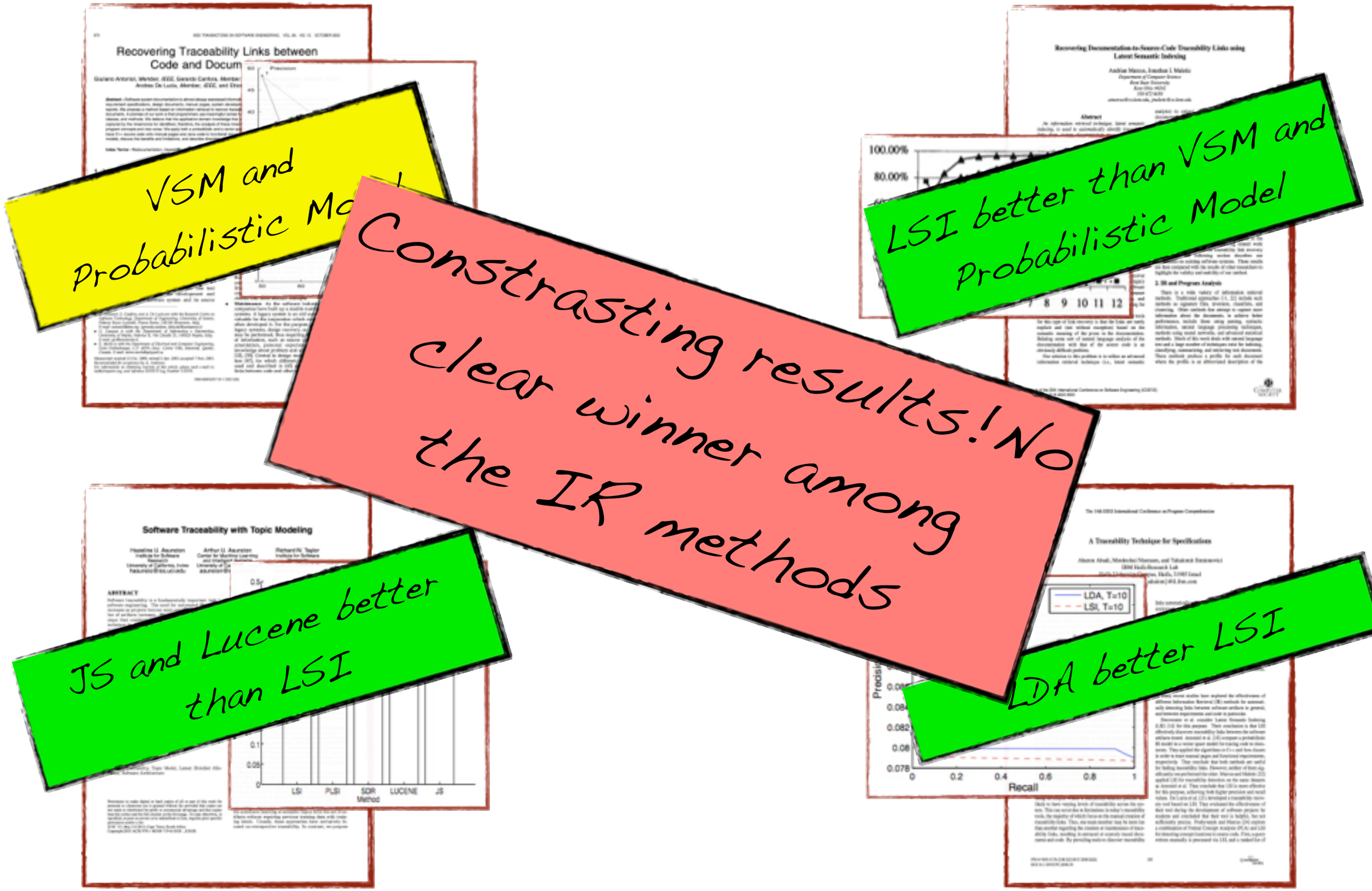
State of the art







State of the art



Our previous study...

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

Rocco Oliveto*, Malcom Gethers†, Denys Poshyvanyk†, Andrea De Lucia*
*Department of Mathematics and Informatics, University of Salerno, via Ponte don Melillo, Fisciano (SA), Italy
†Computer Science Department, The College of William and Mary, Williamsburg, VA 23185, USA
roliveto@unisa.it, mgethers@cs.wm.edu, denys@cs.wm.edu, adelucia@unisa.it

***Abstract*—We present an empirical study to statistically analyze the equivalence of several traceability recovery methods based on Information Retrieval (IR) techniques. The analysis is based on Principal Component Analysis and on the analysis of the overlap of the set of candidate links provided by each method. The studied techniques are the Jensen-Shannon (JS) method, Vector Space Model (VSM), Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation (LDA). The results show that while JS, VSM, and LSI are almost equivalent, LDA is able to capture a dimension unique to the set of techniques which we considered.**

***Keywords*-Traceability Recovery; Information Retrieval; Empirical Studies.**

I. Introduction

Extensive effort in the software engineering community (both research and commercial) has been brought forth to improve the explicit connection of documentation and source code. Promising results have been achieved using Information Retrieval (IR) techniques [1], [2] for traceability recovery (e.g., [3], [4]). IR-based methods propose a list of candidate traceability links on the basis of the similarity between the text contained in the software artifacts. Such methods are based on the conjecture that two artifacts having high textual similarity share several concepts thus they are good candidates to be traced on each other.

Several IR methods have been proposed for traceability recovery – e.g., Vector Space Model (VSM) and probabilistic model [1] or Latent Semantic Indexing (LSI) [2]. In general, the retrieval accuracy of IR-based traceability recovery methods is assessed through two measures: recall, measuring the percentage of correct links that were found, and precision, measuring the percentage of found links that were correct. The results achieved are sometimes contrasting and demonstrate no clear winner among IR techniques. Indeed, it seems that all the exploited techniques so far are able to capture the same information when used to calculate the textual similarity between software artifacts.

In this paper we present an empirical study aiming at statistically analyzing the equivalence of different IR-based traceability recovery methods. The comparison is based on Principal Component Analysis (PCA) and on the analysis of the overlap of the set of candidate links provided by each of the IR methods. The studied IR techniques are the Jensen-Shannon (JS) method [5], VSM, LSI, and Latent Dirichlet Allocation (LDA) [6]. The first three methods were selected because they are widely used and seem to be the techniques that give the best results [5], [3], [4]. LDA is not as widely used for traceability link recovery though it has been used recently [16]. However, we also experiment such a technique for traceability recovery because LDA is able to capture some aspects missed by other IR methods, such as LSI, when it is used in other contexts [7].

The empirical analysis has been conducted on two software repositories, namely EasyClinic and eTour. The studied IR methods have been used to recover traceability links between the use cases and the source code of the two software systems. The results prove that the accuracy of LDA is lower than previously used methods. However, while JS, VSM, and LSI are almost equivalent, LDA is able to capture some information missed by the other exploited IR methods. These considerations suggest that probably LDA can be used as a method to augment canonical methods—e.g., JS, VSM, and LSI—aiming at improving their accuracy.

The rest of the paper is organized as follows. Section II discusses related work, while Section III briefly describe an IR-based traceability recovery process. Sections IV and V provide details on the design of the case study and report the results achieved, respectively. Section VI gives concluding remarks.

II. Related Work

The use of IR methods for traceability recovery was introduced by Antoniol *et al.* [3]. They apply probabilistic and Vector Space Models [1] to trace source code onto software documentation. Later, other IR methods (e.g., LSI, JS method and Numerical Analysis) have been proposed to recover links between different types of artifacts [4], [5], [8]. In particular, IR methods have been used to recover traceability between requirements [9], between requirements and design artifacts [10], between maintenance requests and software documents [11], and between other types of artifacts (e.g., use cases and UML diagrams) [12], [13].

All these reported case studies compare different IR-based traceability recovery approaches using recall and precision.

Which is the best IR method?

Is the ranking of these methods different?

Our previous study...

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

Rocco Oliveto*, Malcom Gethers†, Denys Poshyvanyk†, Andrea De Lucia*
*Department of Mathematics and Informatics, University of Salerno, via Ponte don Melillo, Fisciano (SA), Italy
†Computer Science Department, The College of William and Mary, Williamsburg, VA 23185, USA
roliveto@unisa.it, mgethers@cs.wm.edu, denys@cs.wm.edu, adelucia@unisa.it

***Abstract*—We present an empirical study to statistically analyze the equivalence of several traceability recovery methods based on Information Retrieval (IR) techniques. The analysis is based on Principal Component Analysis and on the analysis of the overlap of the set of candidate links provided by each method. The studied techniques are the Jensen-Shannon (JS) method, Vector Space Model (VSM), Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation (LDA). The results show that while JS, VSM, and LSI are almost equivalent, LDA is able to capture a dimension unique to the set of techniques which we considered.**

***Keywords*-Traceability Recovery; Information Retrieval; Empirical Studies.**

I. Introduction

Extensive effort in the software engineering community (both research and commercial) has been brought forth to improve the explicit connection of documentation and source code. Promising results have been achieved using Information Retrieval (IR) techniques [1], [2] for traceability recovery (e.g., [3], [4]). IR-based methods propose a list of candidate traceability links on the basis of the similarity between the text contained in the software artifacts. Such methods are based on the conjecture that two artifacts having high textual similarity share several concepts thus they are good candidates to be traced on each other.

Several IR methods have been proposed for traceability recovery – e.g., Vector Space Model (VSM) and probabilistic model [1] or Latent Semantic Indexing (LSI) [2]. In general, the retrieval accuracy of IR-based traceability recovery methods is assessed through two measures: recall, measuring the percentage of correct links that were found, and precision, measuring the percentage of found links that were correct. The results achieved are sometimes contrasting and demonstrate no clear winner among IR techniques. Indeed, it seems that all the exploited techniques so far are able to capture the same information when used to calculate the textual similarity between software artifacts.

In this paper we present an empirical study aiming at statistically analyzing the equivalence of different IR-based traceability recovery methods. The comparison is based on Principal Component Analysis (PCA) and on the analysis of the overlap of the set of candidate links provided by each of the IR methods. The studied IR techniques are the Jensen-Shannon (JS) method [5], VSM, LSI, and Latent Dirichlet Allocation (LDA) [6]. The first three methods were selected because they are widely used and seem to be the techniques that give the best results [5], [3], [4]. LDA is not as widely used for traceability link recovery though it has been used recently [16]. However, we also experiment such a technique for traceability recovery because LDA is able to capture some aspects missed by other IR methods, such as LSI, when it is used in other contexts [7].

The empirical analysis has been conducted on two software repositories, namely EasyClinic and eTour. The studied IR methods have been used to recover traceability links between the use cases and the source code of the two software systems. The results prove that the accuracy of LDA is lower than previously used methods. However, while JS, VSM, and LSI are almost equivalent, LDA is able to capture some information missed by the other exploited IR methods. These considerations suggest that probably LDA can be used as a method to augment canonical methods—e.g., JS, VSM, and LSI—aiming at improving their accuracy.

The rest of the paper is organized as follows. Section II discusses related work, while Section III briefly describe an IR-based traceability recovery process. Sections IV and V provide details on the design of the case study and report the results achieved, respectively. Section VI gives concluding remarks.

II. Related Work

The use of IR methods for traceability recovery was introduced by Antoniol *et al.* [3]. They apply probabilistic and Vector Space Models [1] to trace source code onto software documentation. Later, other IR methods (e.g., LSI, JS method and Numerical Analysis) have been proposed to recover links between different types of artifacts [4], [5], [8]. In particular, IR methods have been used to recover traceability between requirements [9], between requirements and design artifacts [10], between maintenance requests and software documents [11], and between other types of artifacts (e.g., use cases and UML diagrams) [12], [13].

All these reported case studies compare different IR-based traceability recovery approaches using recall and precision.

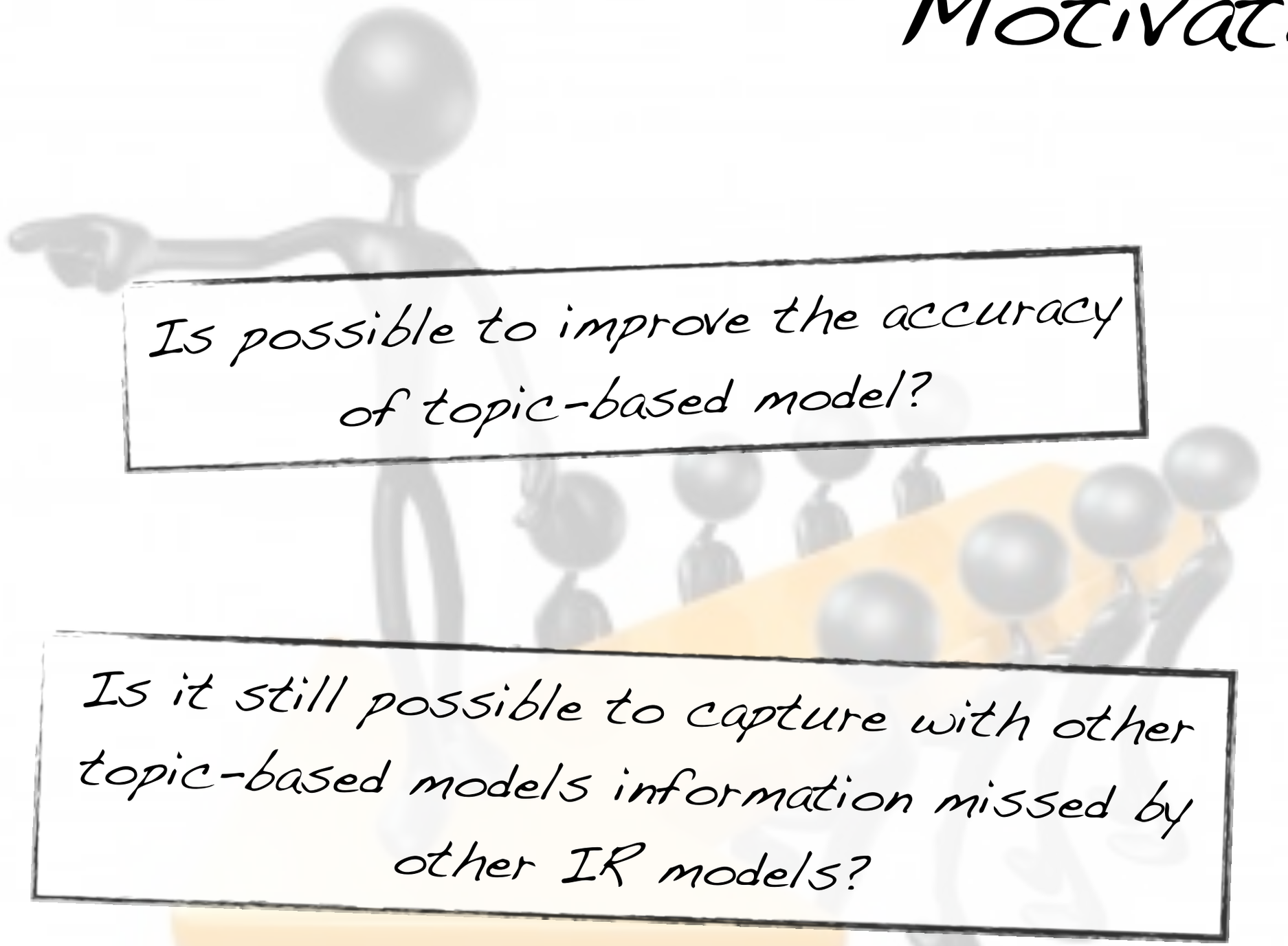
Which is the best IR method?

Is the ranking of these methods different?

LDA had lowest accuracy

LDA is orthogonal as compared to VSM, JS, and LSI

Motivation



Is possible to improve the accuracy of topic-based model?

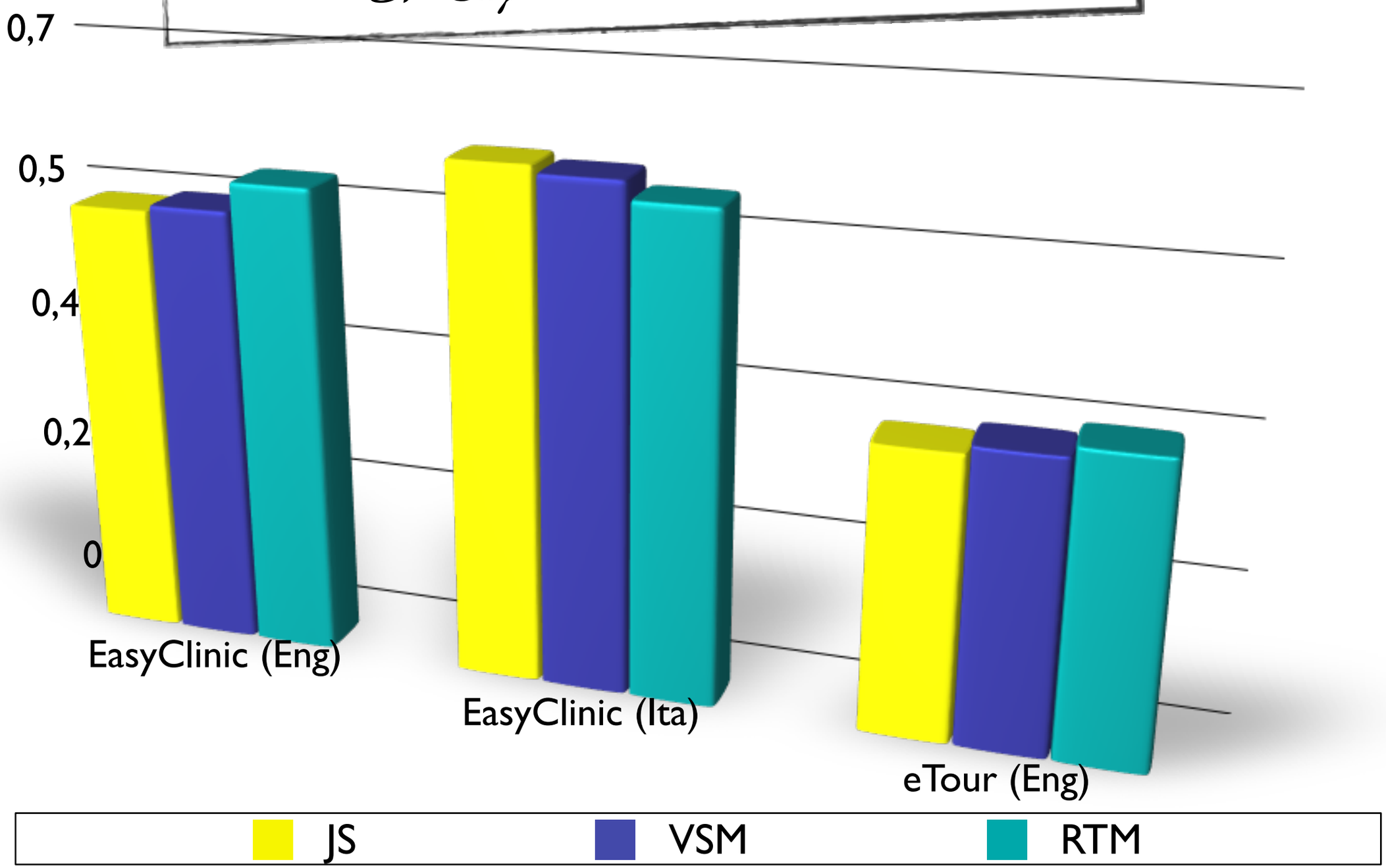
Is it still possible to capture with other topic-based models information missed by other IR models?

Relational Topic Model (RTM)

Is possible to improve the accuracy of topic-based model?

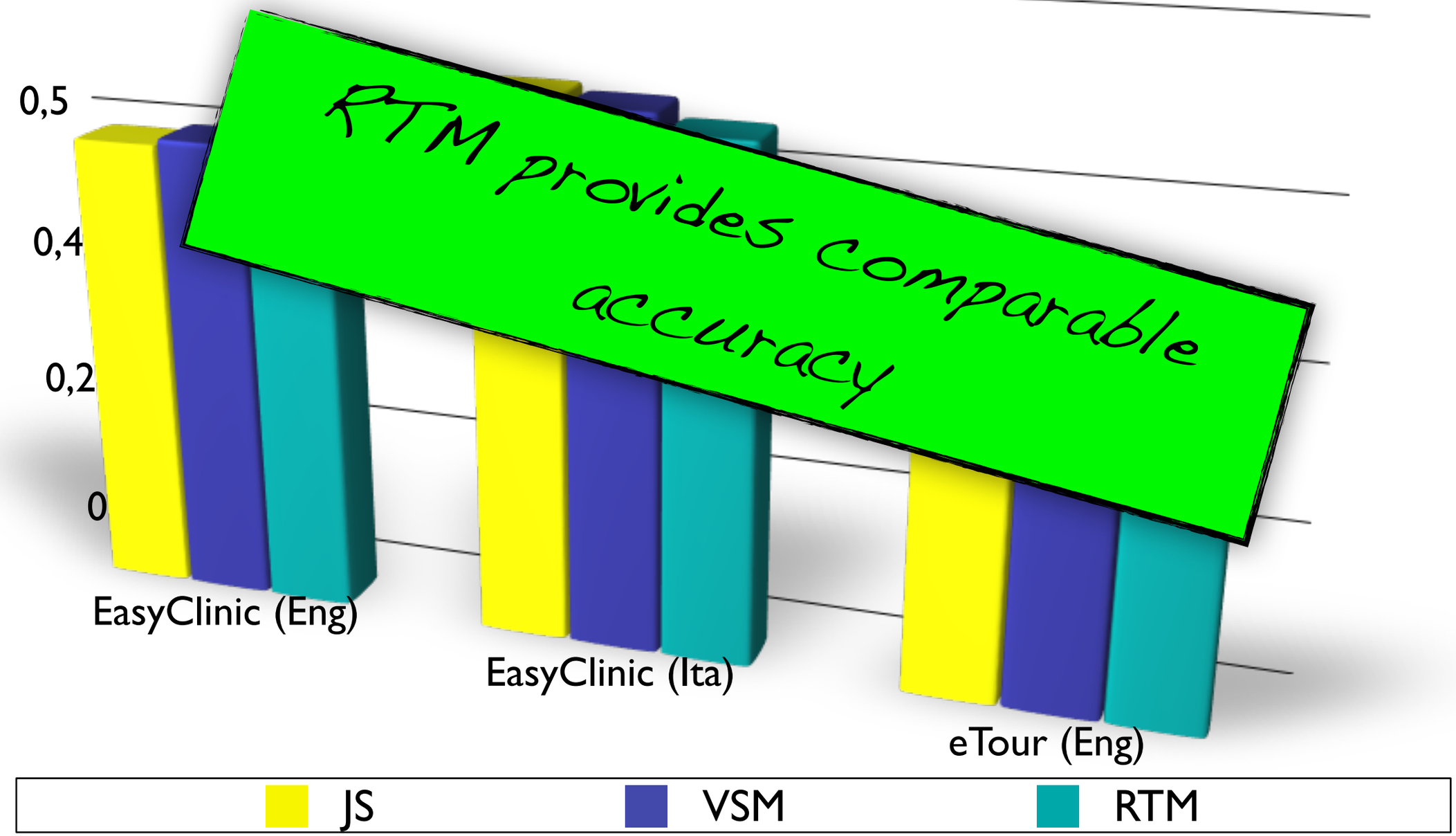
Relational Topic Model (RTM)

Is possible to improve the accuracy of topic-based model?



Relational Topic Model (RTM)

Is possible to improve the accuracy of topic-based model?



Orthogonality of Techniques

Is still possible to capture with other topic-based models information missed by other IR models?

Orthogonality of Techniques

	PC1	PC2	PC3
Proportio	68.51%	31.18%	0.31%
Cumulativ	68.51%	99.69%	100%
JS	0.99	0.11	0.07
VSM	0.99	0.08	-0.06
RTM	0.29	0.95	0.00

Is still possible to capture with other topic-based models information missed by other IR models?

Orthogonality of Techniques

	25	50	100
correctINT(JS,VSM)	91%	94%	85%
correctDIFF(JS,VSM)	4%	5%	7%
correctDIFF(VSM,JS)	4%	0%	7%
correctINT(JS,RTM)	23%	28%	35%
correctDIFF(JS,RTM)	41%	35%	22%
correctDIFF(RTM,JS)	35%	36%	41%
correctINT(VSM,RTM)	23%	29%	32%
correctDIFF(VSM,RTM)	41%	32%	24%
correctDIFF(RTM,VSM)	35%	38%	42%

Is still possible to capture with other topic-based models information missed by other IR models?

Orthogonality of Techniques

	25	50	100
correctINT(JS,VSM)	91%	94%	85%
correctDIFF(JS,VSM)	4%	5%	7%
correctDIFF(VSM,JS)	4%	0%	7%
correctINT(JS,RTM)	23%	28%	35%
correctDIFF(JS,RTM)	41%	35%	22%
correctDIFF(RTM,JS)	35%	36%	41%
correctINT(VSM,RTM)	23%	29%	32%
correctDIFF(VSM,RTM)	41%	32%	24%
correctDIFF(RTM,VSM)	35%	38%	42%

Is still possible to capture with other topic-based models information missed by other IR models?

Orthogonality of Techniques

	25	50	100
correctINT(JS,VSM)	91%	94%	85%
correctDIFF(JS,VSM)	4%	5%	7%
correctDIFF(VSM,JS)	4%	0%	7%
correctINT(JS,RTM)	23%	28%	35%
correctDIFF(JS,RTM)	41%	35%	22%
correctDIFF(RTM,JS)	35%	36%	41%
correctINT(VSM,RTM)	23%	29%	32%
correctDIFF(VSM,RTM)	41%	32%	24%
correctDIFF(RTM,VSM)	35%	38%	42%

Is still possible to capture with other topic-based models information missed by other IR models?

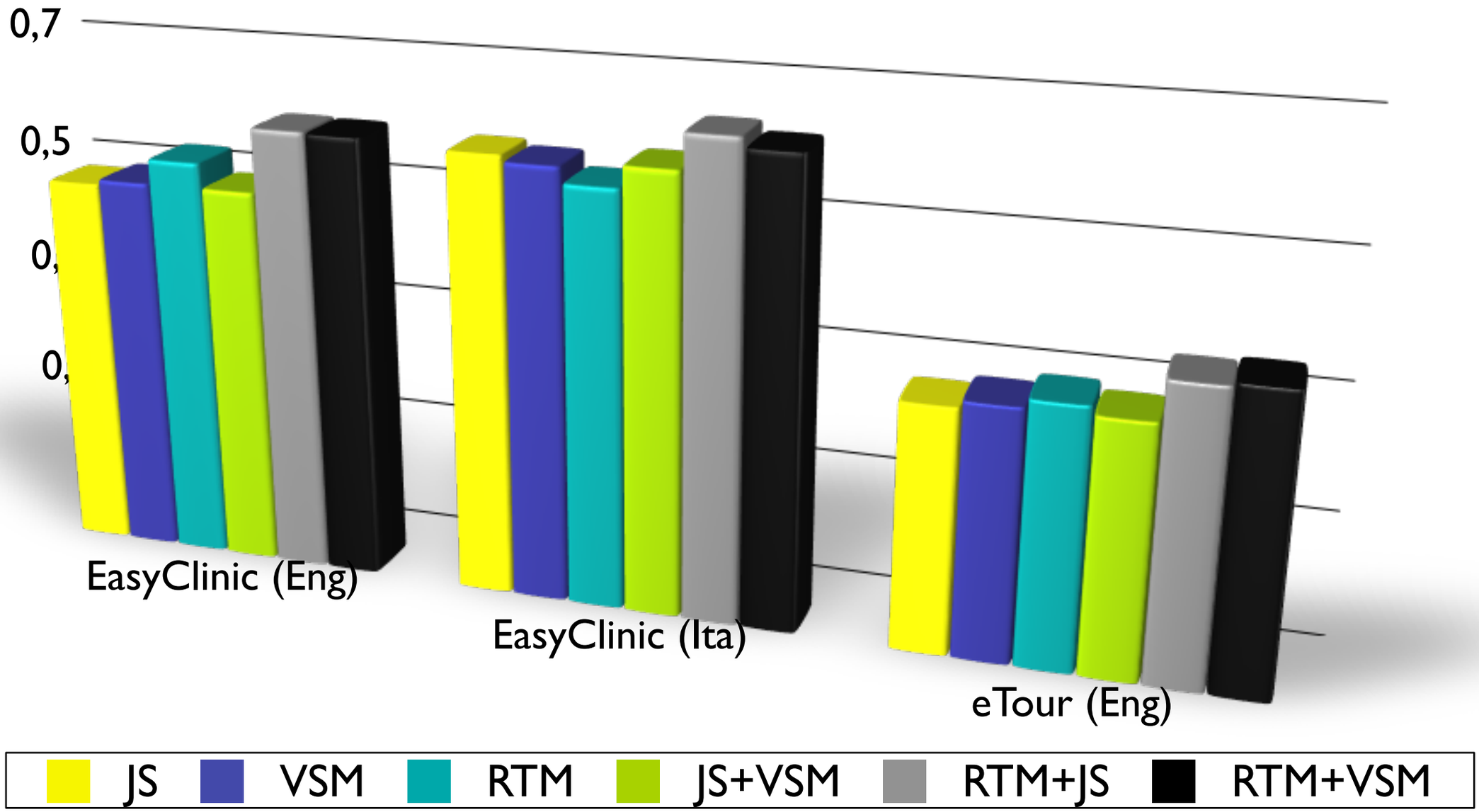
Orthogonality of Techniques

	25	50	100
correctINT(JS,VSM)	91%	94%	85%
correc	4%	5%	7%
corre		0%	7%
corr			35%
correctDIFF(JS,R			
correctDIFF(RTM,JS)			
correctINT(VSM,RTM)	23%	29%	%
correctDIFF(VSM,RTM)	41%	32%	24%
correctDIFF(RTM,VSM)	35%	38%	42%

RTM provides orthogonal results

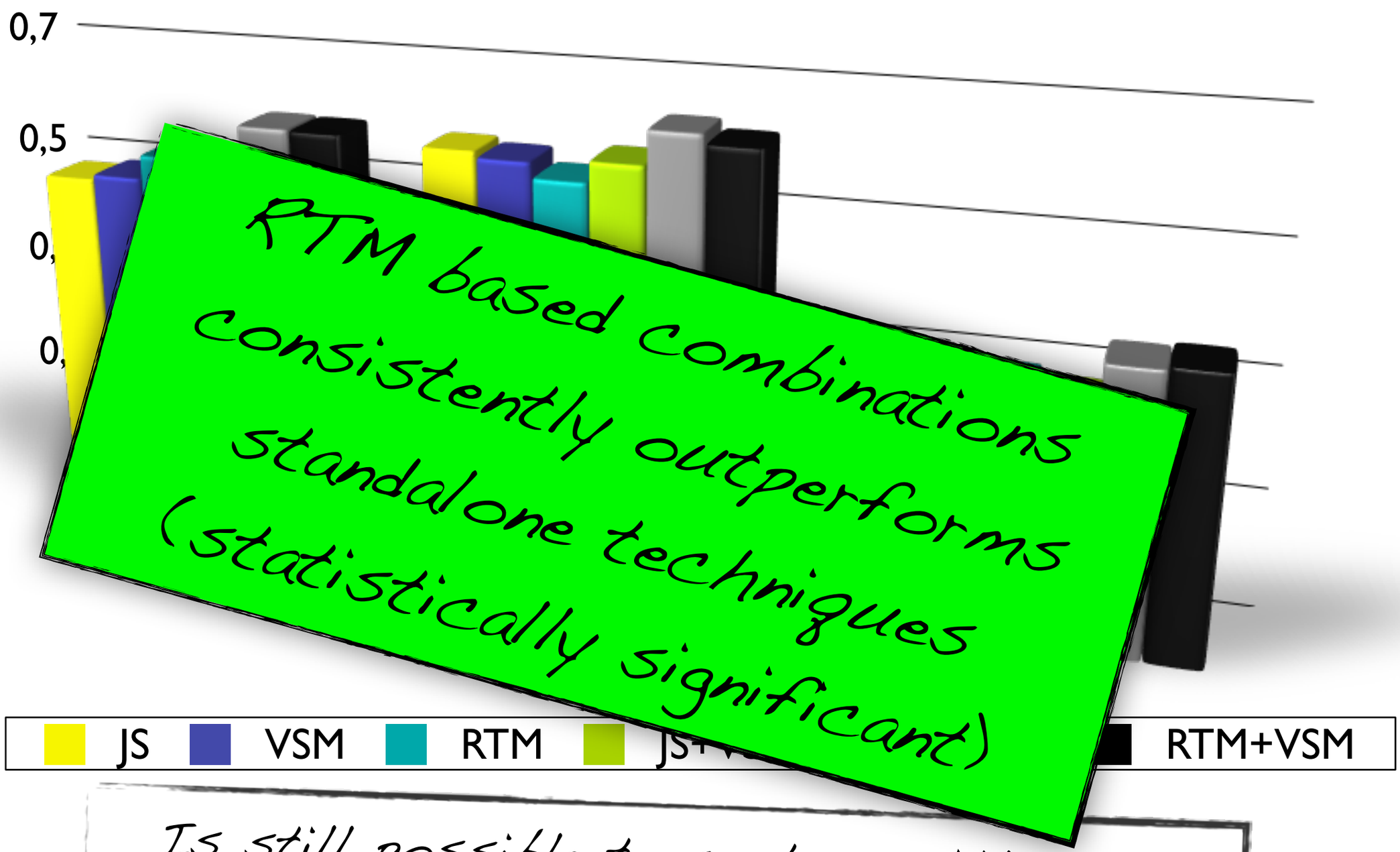
Is still possible to capture with other topic-based models information missed by other IR models?

Combining Orthogonal Techniques



Is still possible to capture with other topic-based models information missed by other IR models?

Combining Orthogonal Techniques



Is still possible to capture with other topic-based models information missed by other IR models?

Other Results

There is a statistically significant interaction between IR method and artifact type

There is a statistically significant interaction between IR method and language

Conclusions

RTM provides accuracy comparable to existing IR techniques

RTM is able to capture orthogonal information

RTM combined with other IR techniques provides improved accuracy for traceability recovery

Conclusions

Thank You!

Questions?