《On the relationship between bug reports and queries for text retrieval-based bug localization》	2020	ESE
《On Combining IR Methods to Improve Bug Localization》	2020	ICPC
《Industry-scale IR-based Bug Localization: A Perspective from Facebook》	2021	ICSE
《Improving Bug Localization by Mining Crash Reports: An Industrial Study》	2020	ICSME
《A Similarity Integration Method based Information Retrieval and Word Embedding in Bug Localization》	2020	QRS（水）

On Combining IR Methods to Improve Bug Localization

作者：Saket Khatiwada, Miroslav Tushev, and Anas Mahmoud

汇报人：陈冰婷　　　导师：邹卫琴

目录

Contents

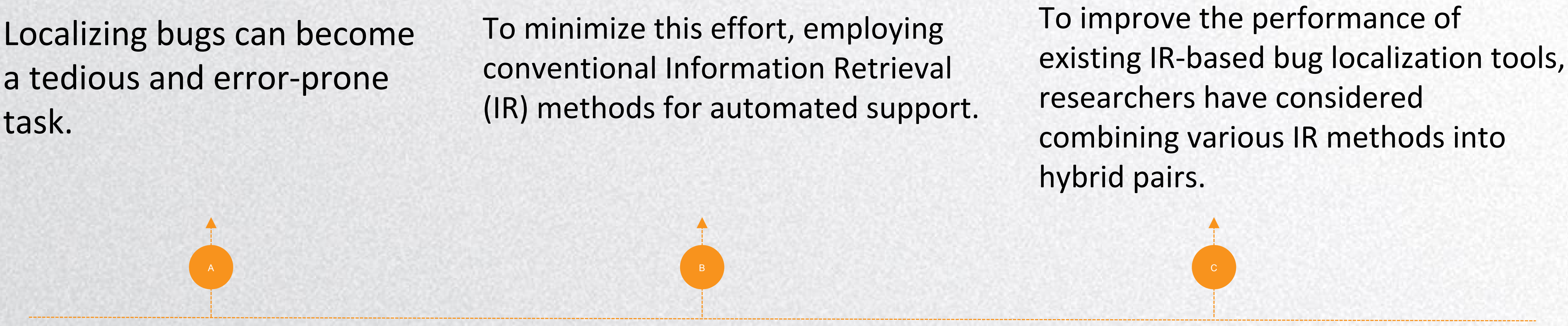
PART 01 Motivation

PART 02 Introduction

PART 03 Experiments

PART 04 Performance

Motivation



Localizing bugs can become a tedious and error-prone task.

To minimize this effort, employing conventional Information Retrieval (IR) methods for automated support.

To improve the performance of existing IR-based bug localization tools, researchers have considered combining various IR methods into hybrid pairs.

Introduction

V

VSM

A representative of string matching methods

L

LSI

A representative of semantically-enabled methods

J

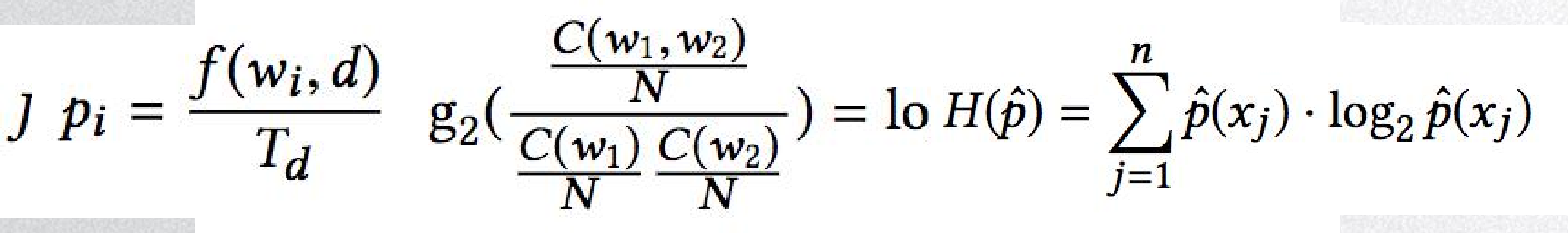
JSM

A representative of probabilisticmethods

P

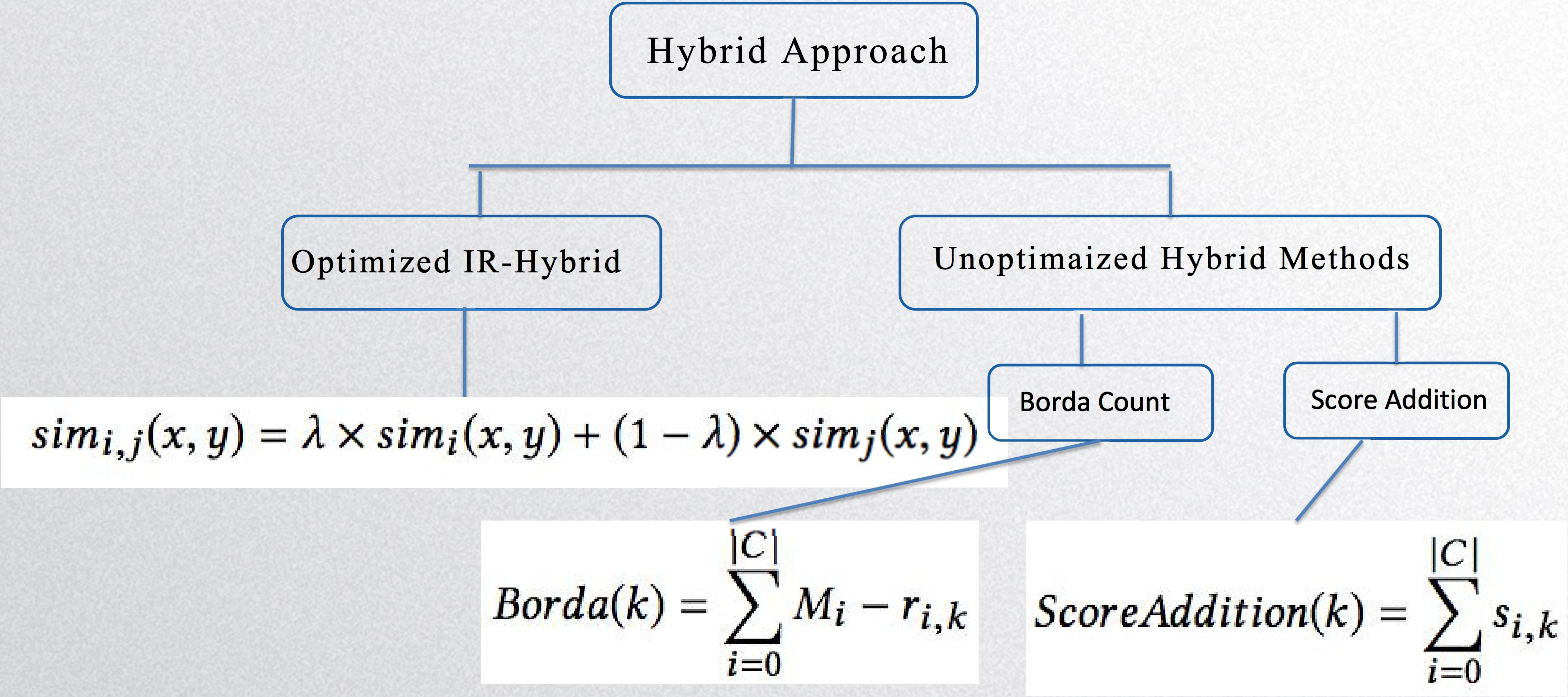
PMI

A representative of information-theoretic text retrieval methods.



$$J \quad p_i = \frac{f(w_i, d)}{T_d} \quad g_2\left(\frac{\frac{C(w_1, w_2)}{N}}{\frac{C(w_1)}{N} \frac{C(w_2)}{N}}\right) = \mathrm{lo} \; H(\hat{p}) = \sum_{j=1}^{n} \hat{p}(x_j) \cdot \log_2 \hat{p}(x_j)$$

Hybrid Approach



Hybrid Approach

Optimized IR-Hybrid

$$sim_{i,j}(x,y) = \lambda \times sim_i(x,y) + (1-\lambda) \times sim_j(x,y)$$

Unoptimaized Hybrid Methods

Borda Count

$$Borda(k) = \sum_{i=0}^{|C|} M_i - r_{i,k}$$

Score Addition

$$ScoreAddition(k) = \sum_{i=0}^{|C|} s_{i,k}$$

Experiment

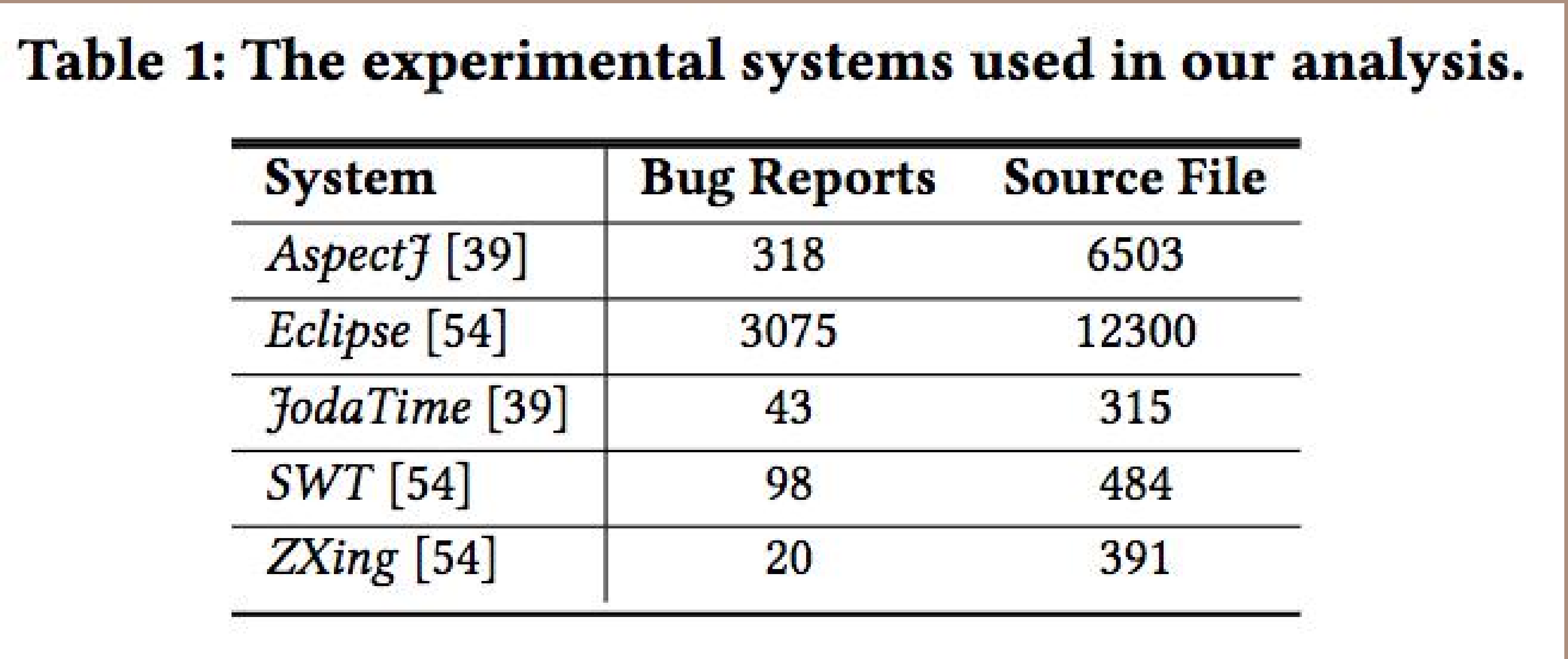


Table 1: The experimental systems used in our analysis.

System	Bug Reports	Source File
AspectJ [39]	318	6503
Eclipse [54]	3075	12300
JodaTime [39]	43	315
SWT [54]	98	484
ZXing [54]	20	391

Evaluation Measures

- MRR
- Top N

Implementation

Indexing

Extracting the textual component of code (code lexicon) embedded in identifier names and internal code comments.

StanfordNLP

Porter Stemmer

Reciprocal Rank (RR)

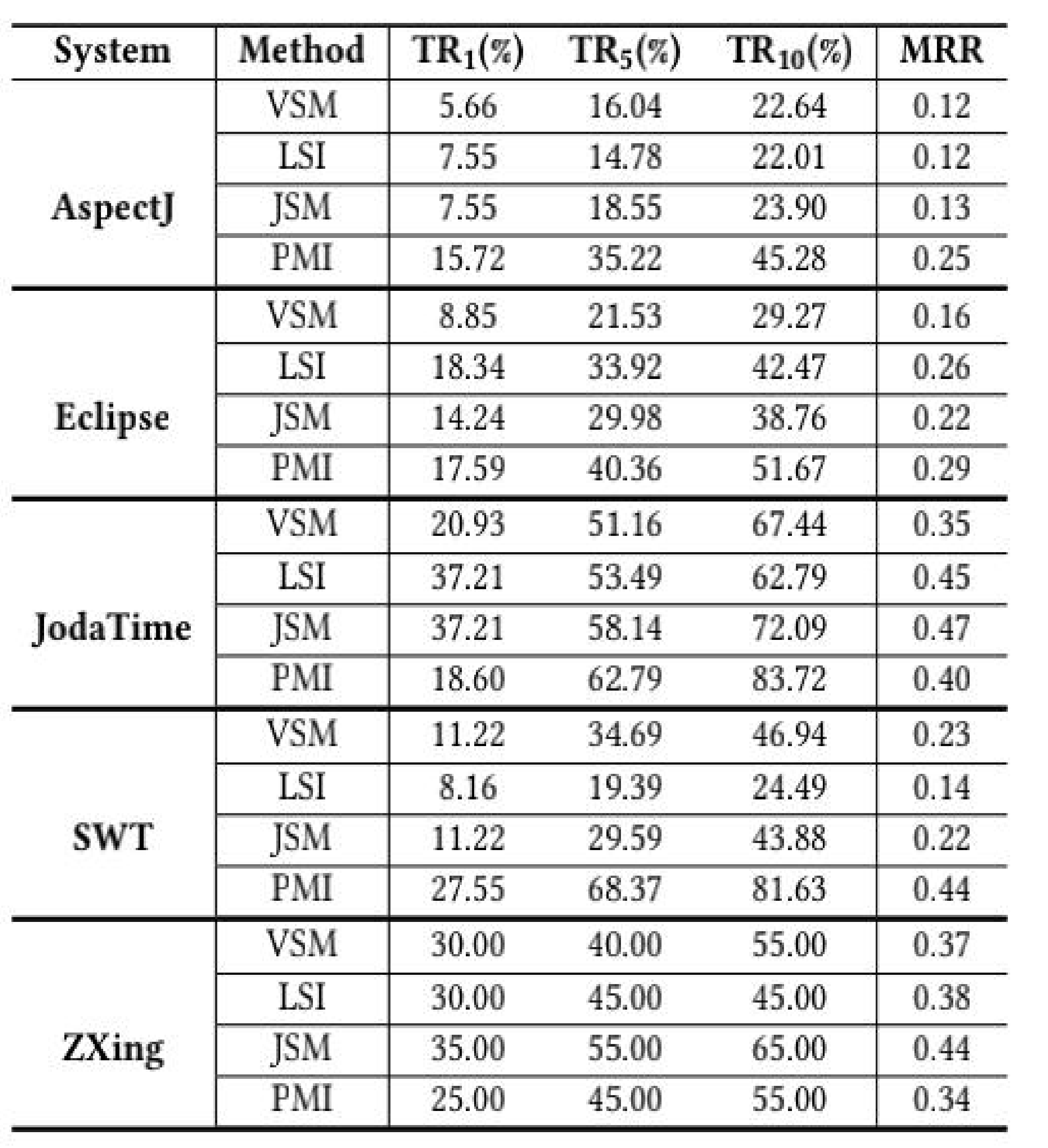
The k value which produced the highest RR was used

Bluebit Matrix Calculator

Analyzing

Specifically, for each query in each of our systems,we generated the LSI space for all k values in the set [50, 100, 150,200, ..., 900]. The performance in terms of reciprocal rank (RR) wasthen measured for each query at eachkvalue.

Individual Performance



System	Method	TR_1(%)	TR_5(%)	TR_{10}(%)	MRR
AspectJ	VSM	5.66	16.04	22.64	0.12
	LSI	7.55	14.78	22.01	0.12
	JSM	7.55	18.55	23.90	0.13
	PMI	15.72	35.22	45.28	0.25
Eclipse	VSM	8.85	21.53	29.27	0.16
	LSI	18.34	33.92	42.47	0.26
	JSM	14.24	29.98	38.76	0.22
	PMI	17.59	40.36	51.67	0.29
JodaTime	VSM	20.93	51.16	67.44	0.35
	LSI	37.21	53.49	62.79	0.45
	JSM	37.21	58.14	72.09	0.47
	PMI	18.60	62.79	83.72	0.40
SWT	VSM	11.22	34.69	46.94	0.23
	LSI	8.16	19.39	24.49	0.14
	JSM	11.22	29.59	43.88	0.22
	PMI	27.55	68.37	81.63	0.44
ZXing	VSM	30.00	40.00	55.00	0.37
	LSI	30.00	45.00	45.00	0.38
	JSM	35.00	55.00	65.00	0.44
	PMI	25.00	45.00	55.00	0.34

Table 2: The performance of the individual IR methods in terms of TR_1, TR_5, TR_{10}, and MRR

Is there any global optimal λ that can be used for combining IR methods in Eq.10?

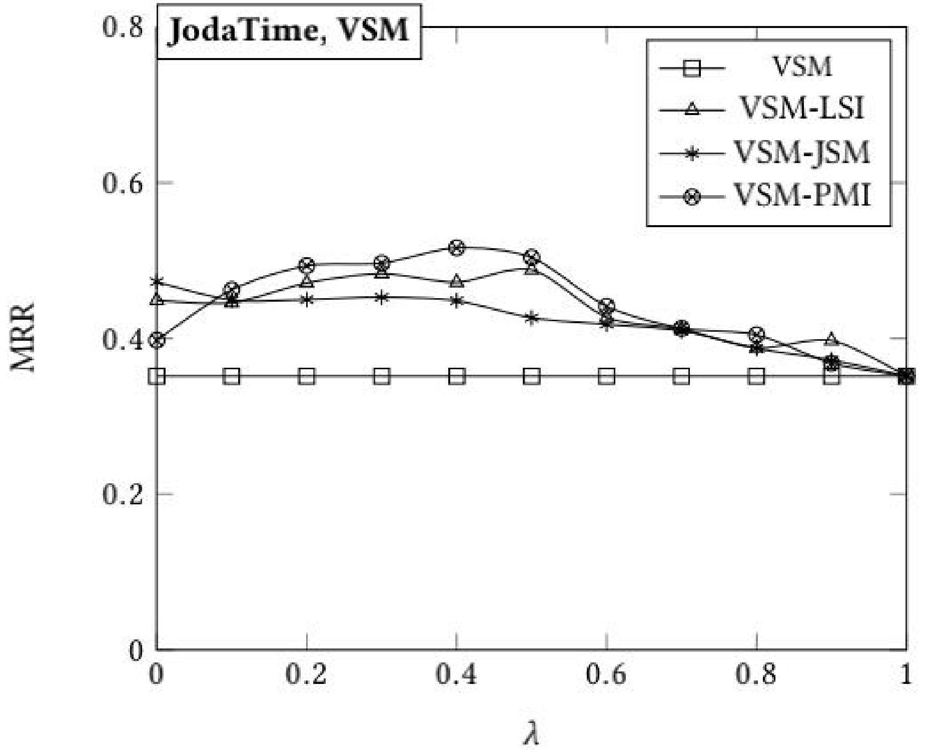
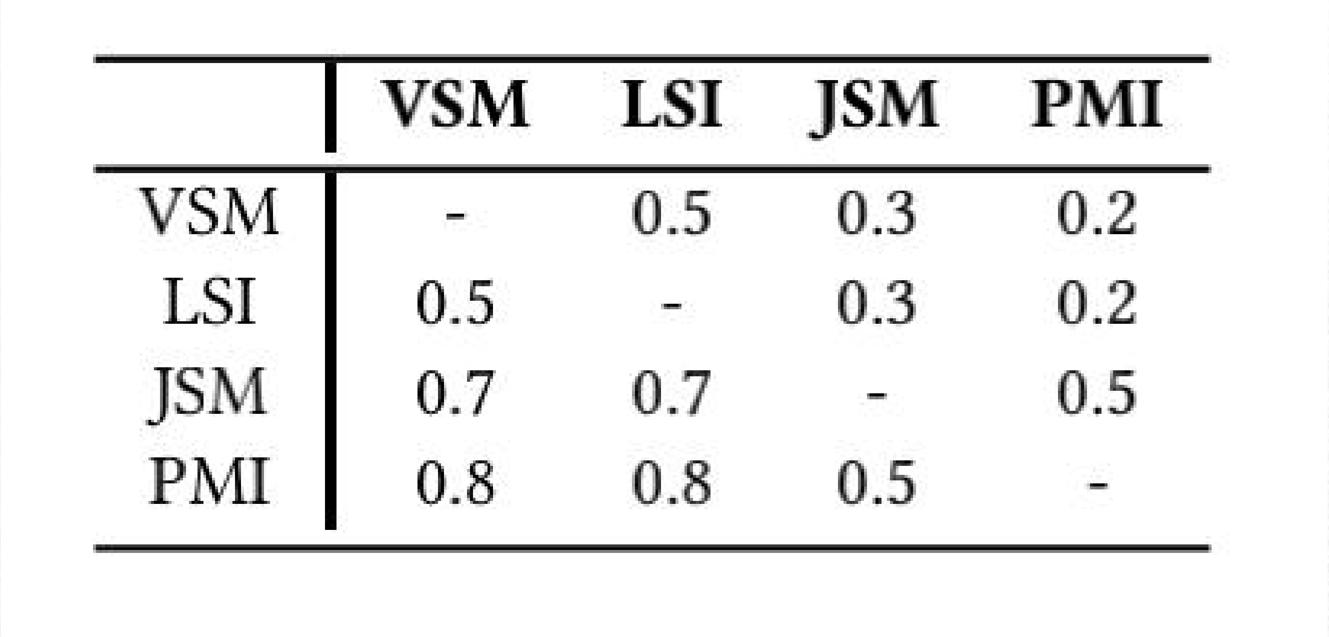


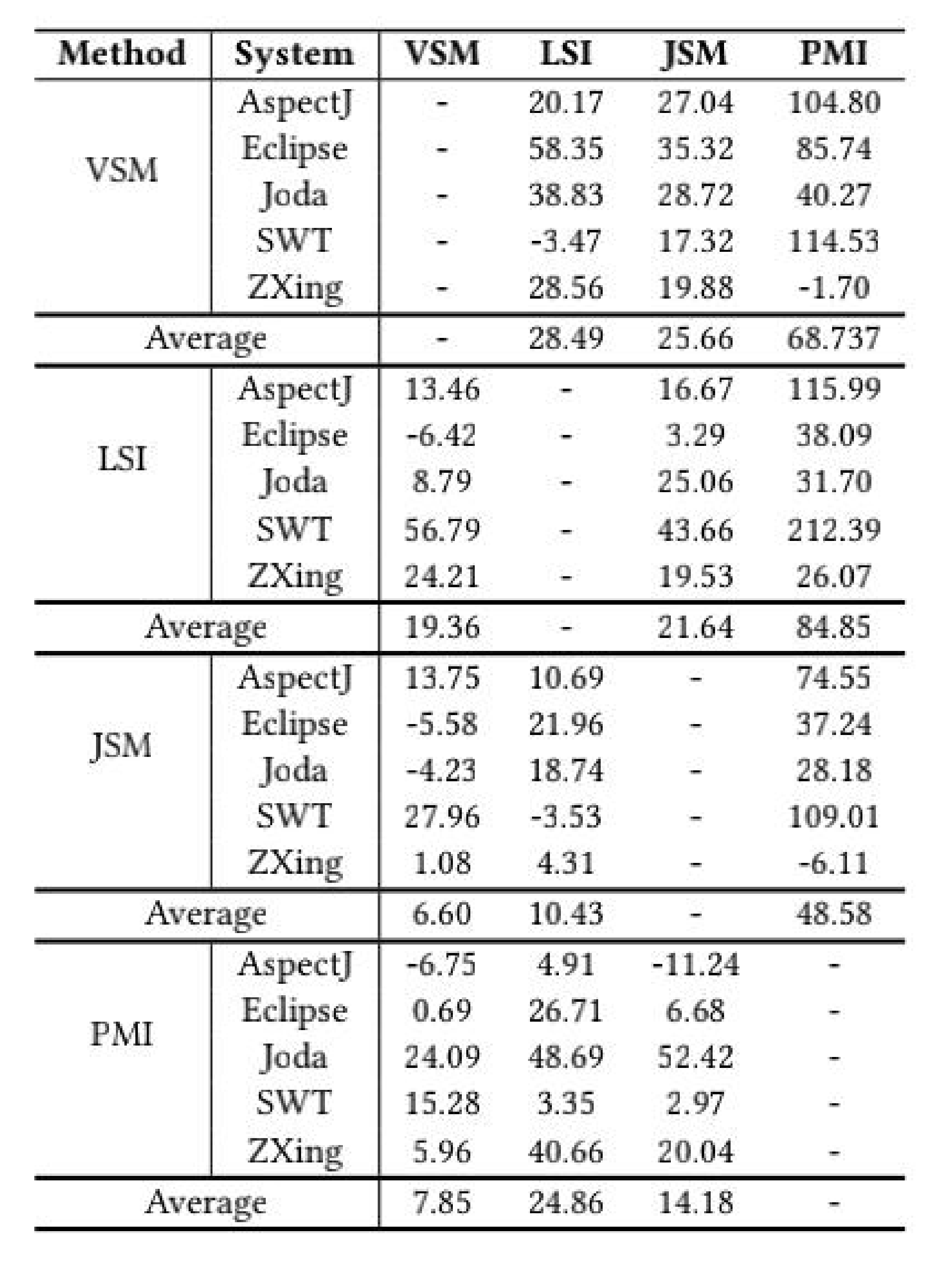
Figure 1: Optimizing λ for VSM in JodaTime.



	VSM	LSI	JSM	PMI
VSM	-	0.5	0.3	0.2
LSI	0.5	-	0.3	0.2
JSM	0.7	0.7	-	0.5
PMI	0.8	0.8	0.5	-

Table 4: Average near-optimal values of λ of the hybrid methods across all experimental systems (Optimized for best MRR).

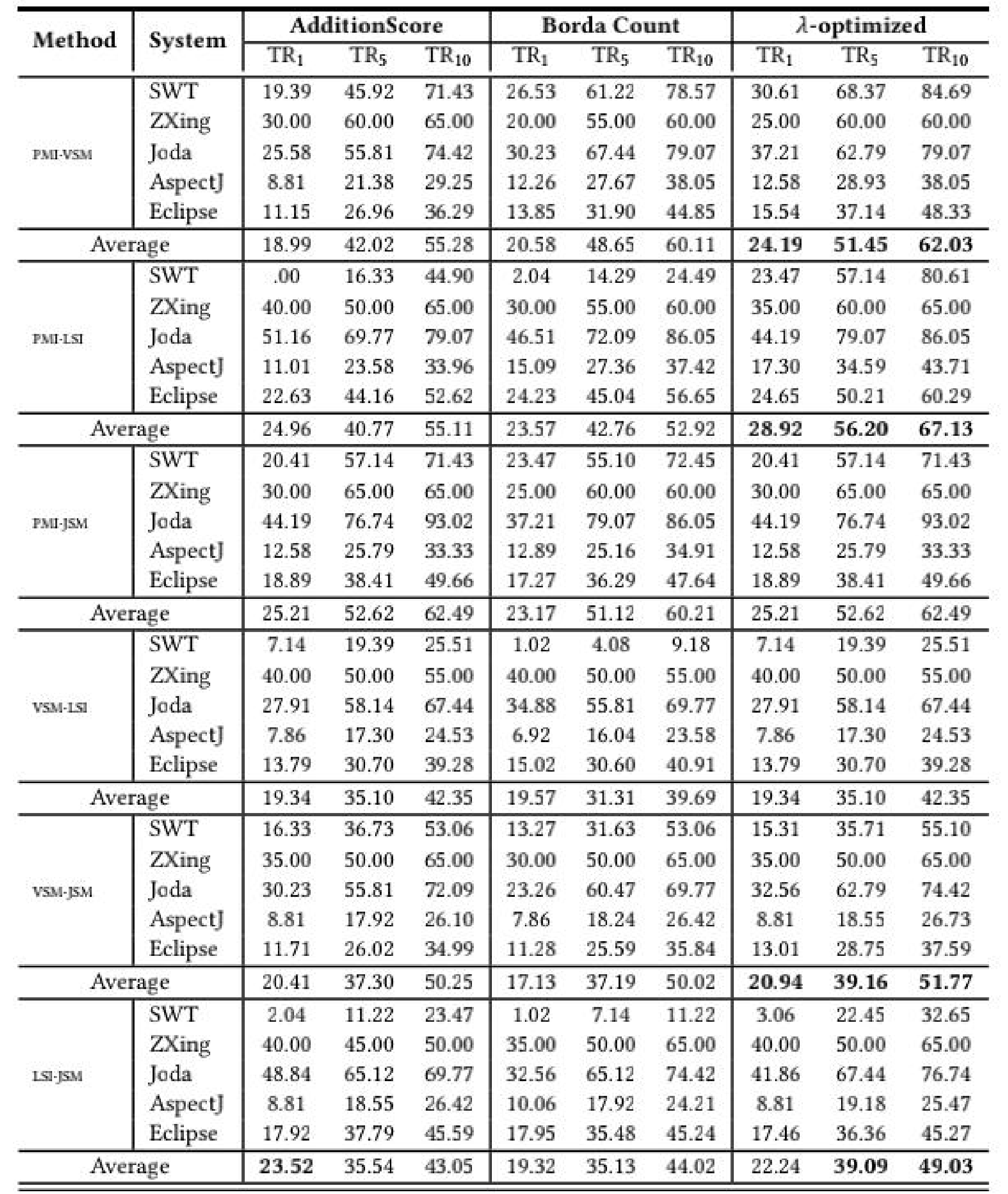
How effective is the hybrid approach in comparison to the individual IR methods?



Method	System	VSM	LSI	JSM	PMI
VSM	AspectJ	-	20.17	27.04	104.80
	Eclipse	-	58.35	35.32	85.74
	Joda	-	38.83	28.72	40.27
	SWT	-	-3.47	17.32	114.53
	ZXing	-	28.56	19.88	-1.70
Average		-	28.49	25.66	68.737
LSI	AspectJ	13.46	-	16.67	115.99
	Eclipse	-6.42	-	3.29	38.09
	Joda	8.79	-	25.06	31.70
	SWT	56.79	-	43.66	212.39
	ZXing	24.21	-	19.53	26.07
Average		19.36	-	21.64	84.85
JSM	AspectJ	13.75	10.69	-	74.55
	Eclipse	-5.58	21.96	-	37.24
	Joda	-4.23	18.74	-	28.18
	SWT	27.96	-3.53	-	109.01
	ZXing	1.08	4.31	-	-6.11
Average		6.60	10.43	-	48.58
PMI	AspectJ	-6.75	4.91	-11.24	-
	Eclipse	0.69	26.71	6.68	-
	Joda	24.09	48.69	52.42	-
	SWT	15.28	3.35	2.97	-
	ZXing	5.96	40.66	20.04	-
Average		7.85	24.86	14.18	-

Table 5: The performance gain (%MRR) of the hybrid methods in comparison to the individual IR methods.

How effective are the λ-optimized IR-hybrids in comparison to the unoptimized hybrids?



Method	System	AdditionScore			Borda Count			λ-optimized		
		TR_1	TR_5	TR_{10}	TR_1	TR_5	TR_{10}	TR_1	TR_5	TR_{10}
PMI-VSM	SWT	19.39	45.92	71.43	26.53	61.22	78.57	30.61	68.37	84.69
	ZXing	30.00	60.00	65.00	20.00	55.00	60.00	25.00	60.00	60.00
	Joda	25.58	55.81	74.42	30.23	67.44	79.07	37.21	62.79	79.07
	AspectJ	8.81	21.38	29.25	12.26	27.67	38.05	12.58	28.93	38.05
	Eclipse	11.15	26.96	36.29	13.85	31.90	44.85	15.54	37.14	48.33
Average		18.99	42.02	55.28	20.58	48.65	60.11	**24.19**	**51.45**	**62.03**
PMI-LSI	SWT	.00	16.33	44.90	2.04	14.29	24.49	23.47	57.14	80.61
	ZXing	40.00	50.00	65.00	30.00	55.00	60.00	35.00	60.00	65.00
	Joda	51.16	69.77	79.07	46.51	72.09	86.05	44.19	79.07	86.05
	AspectJ	11.01	23.58	33.96	15.09	27.36	37.42	17.30	34.59	43.71
	Eclipse	22.63	44.16	52.62	24.23	45.04	56.65	24.65	50.21	60.29
Average		24.96	40.77	55.11	23.57	42.76	52.92	**28.92**	**56.20**	**67.13**
PMI-JSM	SWT	20.41	57.14	71.43	23.47	55.10	72.45	20.41	57.14	71.43
	ZXing	30.00	65.00	65.00	25.00	60.00	60.00	30.00	65.00	65.00
	Joda	44.19	76.74	93.02	37.21	79.07	86.05	44.19	76.74	93.02
	AspectJ	12.58	25.79	33.33	12.89	25.16	34.91	12.58	25.79	33.33
	Eclipse	18.89	38.41	49.66	17.27	36.29	47.64	18.89	38.41	49.66
Average		25.21	52.62	62.49	23.17	51.12	60.21	25.21	52.62	62.49
VSM-LSI	SWT	7.14	19.39	25.51	1.02	4.08	9.18	7.14	19.39	25.51
	ZXing	40.00	50.00	55.00	40.00	50.00	55.00	40.00	50.00	55.00
	Joda	27.91	58.14	67.44	34.88	55.81	69.77	27.91	58.14	67.44
	AspectJ	7.86	17.30	24.53	6.92	16.04	23.58	7.86	17.30	24.53
	Eclipse	13.79	30.70	39.28	15.02	30.60	40.91	13.79	30.70	39.28
Average		19.34	35.10	42.35	19.57	31.31	39.69	19.34	35.10	42.35
VSM-JSM	SWT	16.33	36.73	53.06	13.27	31.63	53.06	15.31	35.71	55.10
	ZXing	35.00	50.00	65.00	30.00	50.00	65.00	35.00	50.00	65.00
	Joda	30.23	55.81	72.09	23.26	60.47	69.77	32.56	62.79	74.42
	AspectJ	8.81	17.92	26.10	7.86	18.24	26.42	8.81	18.55	26.73
	Eclipse	11.71	26.02	34.99	11.28	25.59	35.84	13.01	28.75	37.59
Average		20.41	37.30	50.25	17.13	37.19	50.02	**20.94**	**39.16**	**51.77**
LSI-JSM	SWT	2.04	11.22	23.47	1.02	7.14	11.22	3.06	22.45	32.65
	ZXing	40.00	45.00	50.00	35.00	50.00	65.00	40.00	50.00	65.00
	Joda	48.84	65.12	69.77	32.56	65.12	74.42	41.86	67.44	76.74
	AspectJ	8.81	18.55	26.42	10.06	17.92	24.21	8.81	19.18	25.47
	Eclipse	17.92	37.79	45.59	17.95	35.48	45.24	17.46	36.36	45.27
Average		**23.52**	35.54	43.05	19.32	35.13	44.02	22.24	**39.09**	**49.03**

Table 8: TR_1, TR_5, and TR_{10} for Score Addition, Borda Count, and the λ-optimized approach for all combinations of IR-methods used in our experiment.

How does the performance of individual IR methodsaffect the performance of their hybrid pairs?

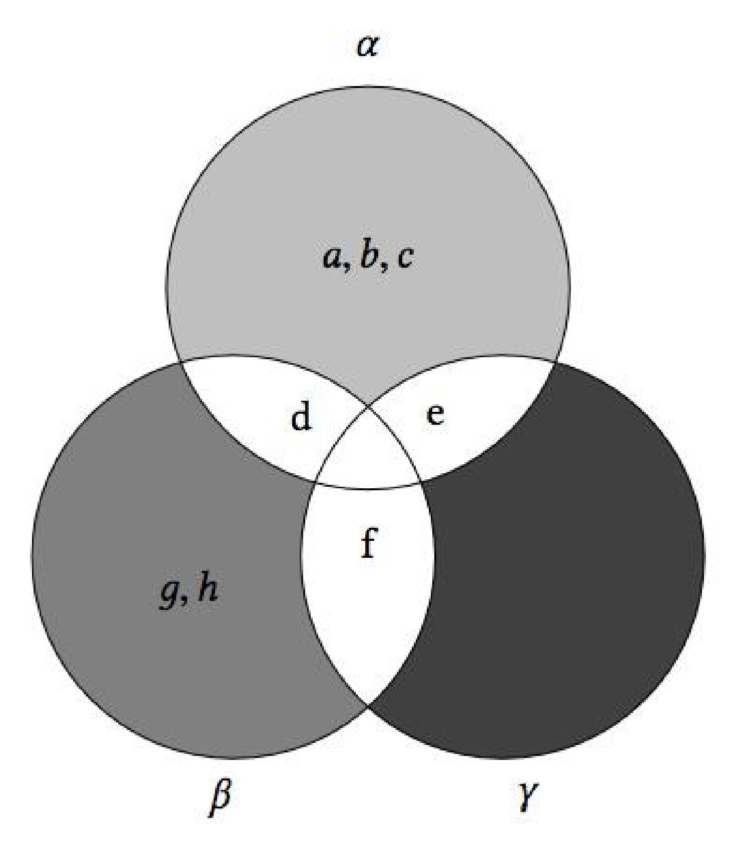
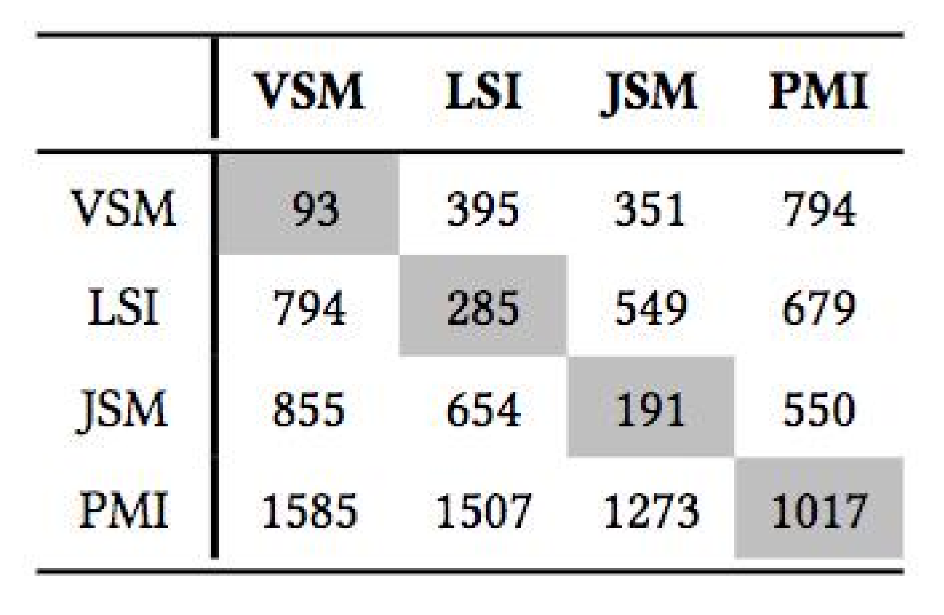


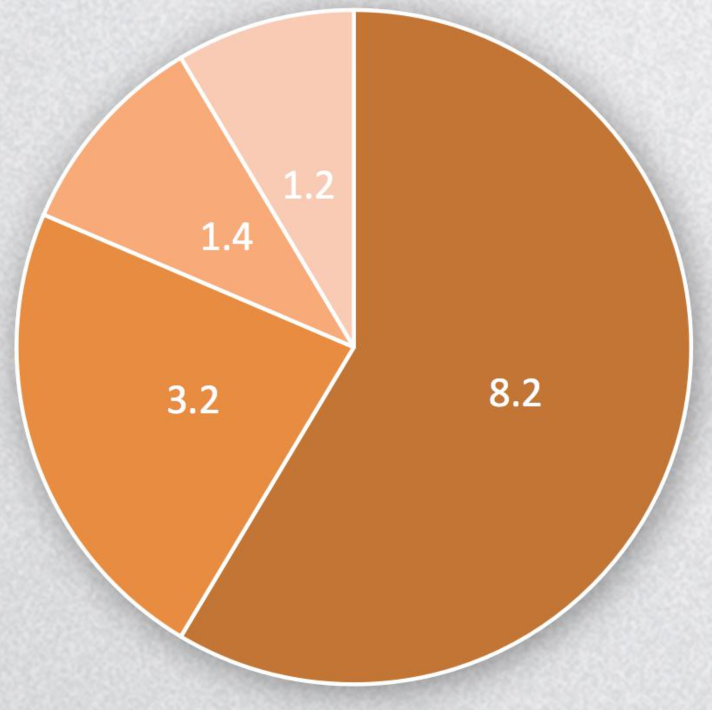
Figure 2: Example: An illustration of the calculation of the number of unique artifacts retrieved by three IR methods α, β, and γ.



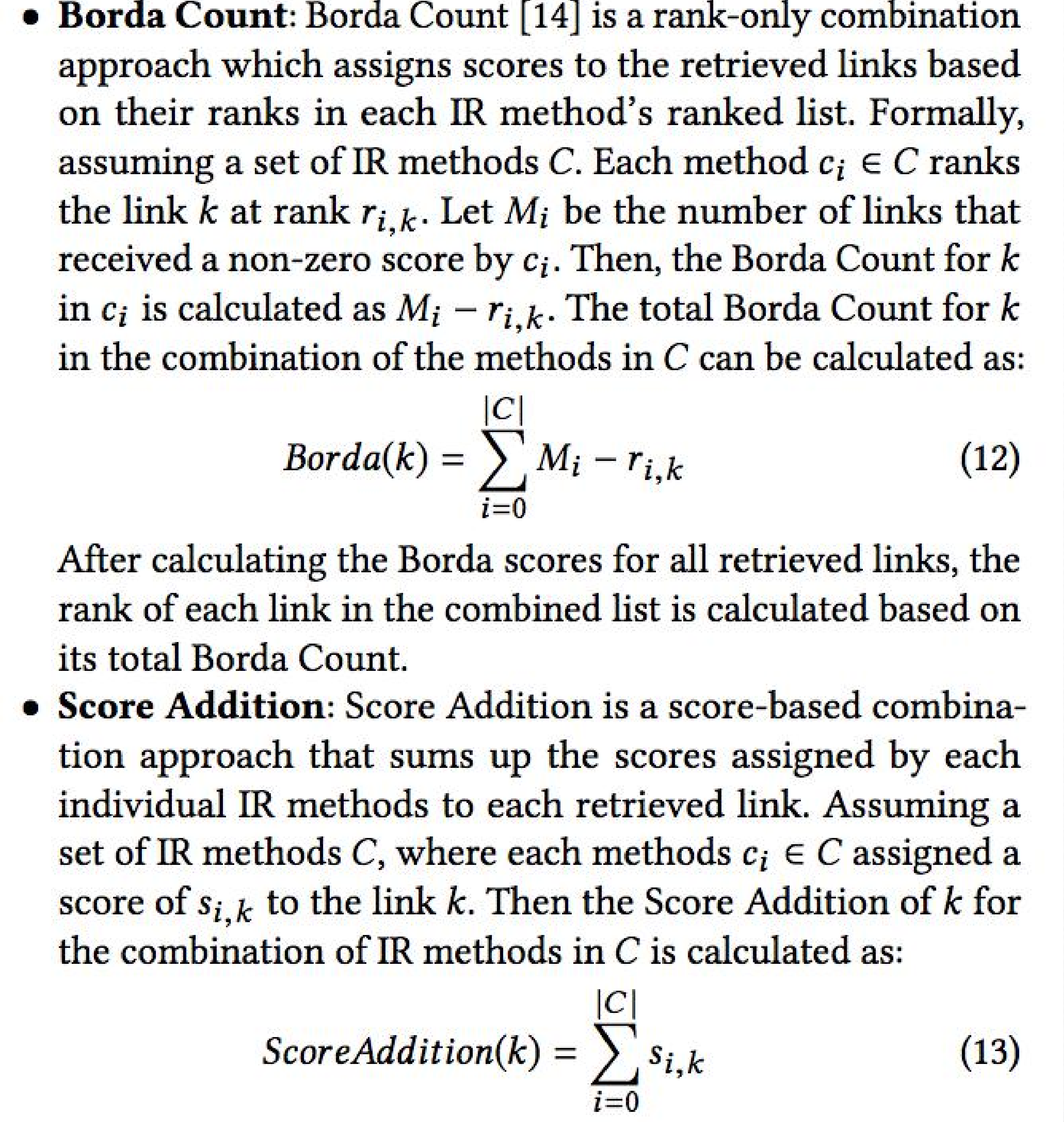
	VSM	LSI	JSM	PMI
VSM	93	395	351	794
LSI	794	285	549	679
JSM	855	654	191	550
PMI	1585	1507	1273	1017

Table 10: Each row shows the number of unique relevant artifacts (true positives) retrieved by each IR method in comparison to the other methods. The diagonal shows the number of relevant artifacts unique to the method (were not retrieved by any other method).

Results



- Combining different methods almost always resulted in improvement over all performance indicators.
- The amount of improvement was highly dependent on theperformance of individual IR methods.



- **Borda Count**: Borda Count [14] is a rank-only combination approach which assigns scores to the retrieved links based on their ranks in each IR method's ranked list. Formally, assuming a set of IR methods C. Each method $c_i \in C$ ranks the link k at rank $r_{i,k}$. Let M_i be the number of links that received a non-zero score by c_i. Then, the Borda Count for k in c_i is calculated as $M_i - r_{i,k}$. The total Borda Count for k in the combination of the methods in C can be calculated as:

  $$Borda(k) = \sum_{i=0}^{|C|} M_i - r_{i,k} \tag{12}$$

  After calculating the Borda scores for all retrieved links, the rank of each link in the combined list is calculated based on its total Borda Count.
- **Score Addition**: Score Addition is a score-based combination approach that sums up the scores assigned by each individual IR methods to each retrieved link. Assuming a set of IR methods C, where each methods $c_i \in C$ assigned a score of $s_{i,k}$ to the link k. Then the Score Addition of k for the combination of IR methods in C is calculated as:

  $$ScoreAddition(k) = \sum_{i=0}^{|C|} s_{i,k} \tag{13}$$

THANK YOU FOR YOUR LISTENING.

谢谢您的聆听