

WETSoDA 2017

Framework Information Based Java Software Architecture Recovery

Xiacong Li, Li Zhang, Ning Ge

Beihang University, Beijing, China

2017-12-04

OUTLINE

Introduction

Problem Statement

Our Solution

Experiment & Result

Conclusion & Future Works



INTRODUCTION



INTRODUCTION

- Program comprehension is a major cost factor
- Lack of documents or the documents are out of date
- Architecture recovery methods are used
- Hierarchical clustering algorithm is used to divide source files into different modules
- The more features two entities have in common, the more likely they are in the same module

HIERARCHICAL CLUSTERING ALGORITHMS

- Examples
 - Single Linkage Algorithm(SLA)
 - Complete Linkage Algorithm(CLA)
 - Weighted Combined Algorithm(WCA)
 - scaLable InforMation BOttleneck(LIMBO)
- We want to improve the accuracy using framework information
 - Spring
 - Struts
 - Hibernate
 - MyBatis

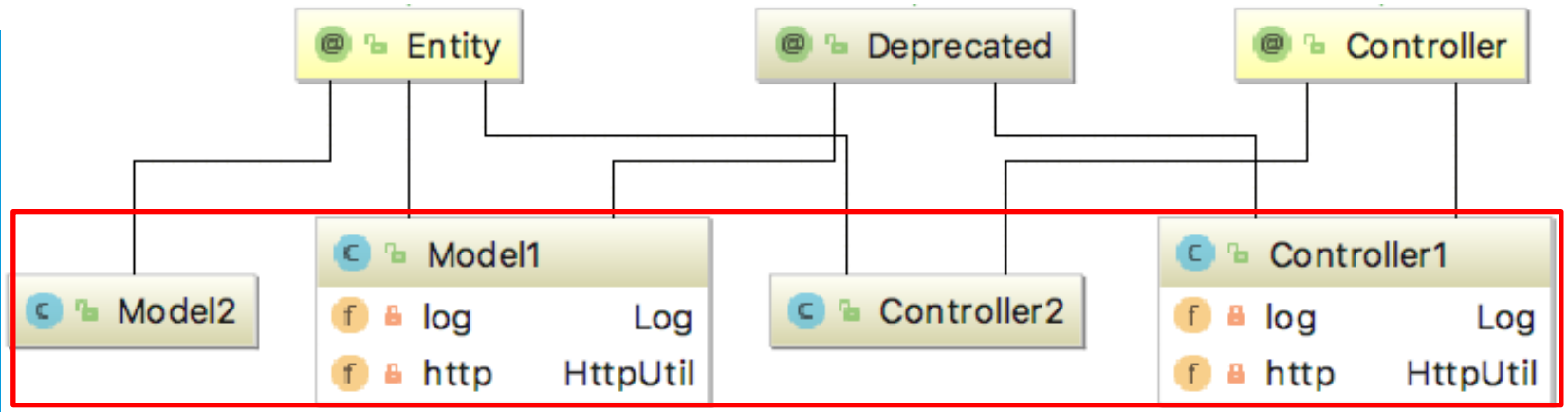
HIERARCHICAL CLUSTERING ALGORITHMS

- Feature selection
- Similarity calculation
- Entities mergence
- Repetition

Use the Spring MVC
as an example

PROBLEM STATEMENT

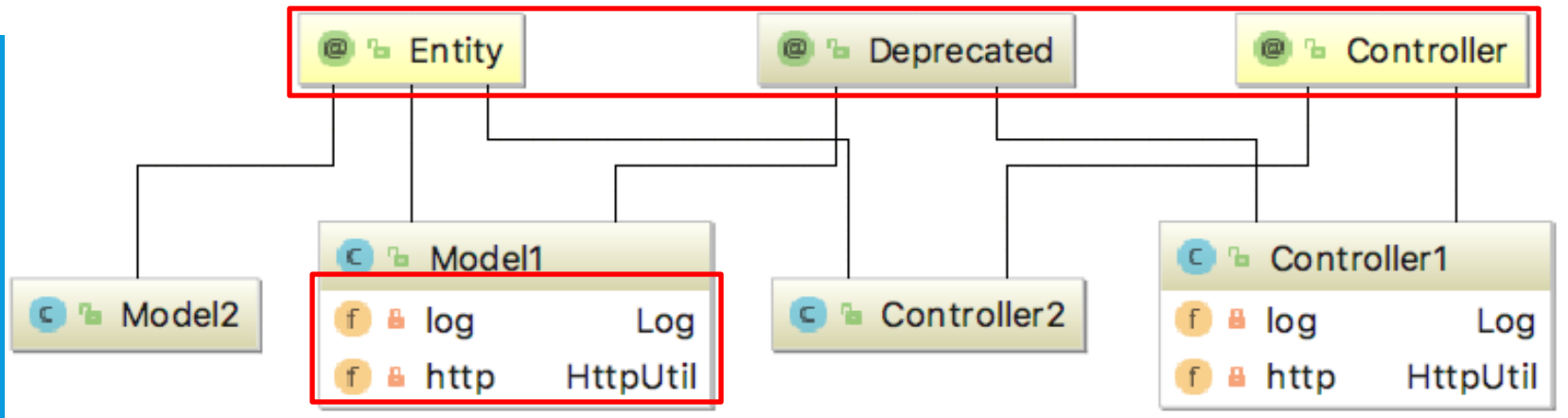
PROBLEM STATEMENT



Log HttpUtil @Entity @Controller @Deprecated

	Log	HttpUtil	@Entity	@Controller	@Deprecated
Model1	1	1	1	0	1
Model2	0	0	1	0	0
Controller1	1	1	0	1	1
Controller2	0	0	0	1	0

PROBLEM STATEMENT



	Log	HttpUtil	@Entity	@Controller	@Deprecated
--	-----	----------	---------	-------------	-------------

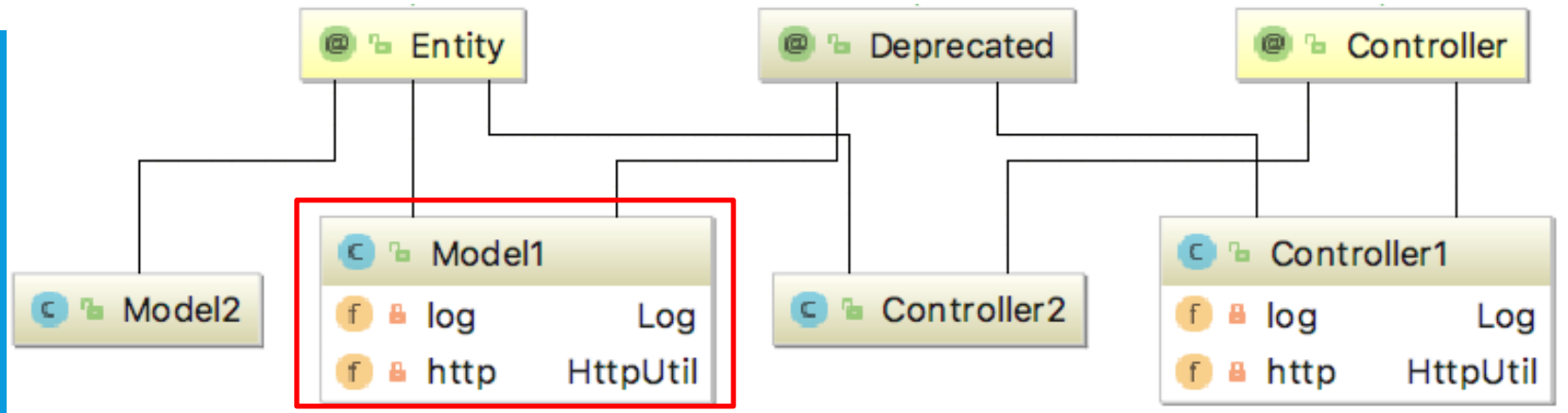
Model1	1	1	1	0	1
--------	---	---	---	---	---

Model2	0	0	1	0	0
--------	---	---	---	---	---

Controller1	1	1	0	1	1
-------------	---	---	---	---	---

Controller2	0	0	0	1	0
-------------	---	---	---	---	---

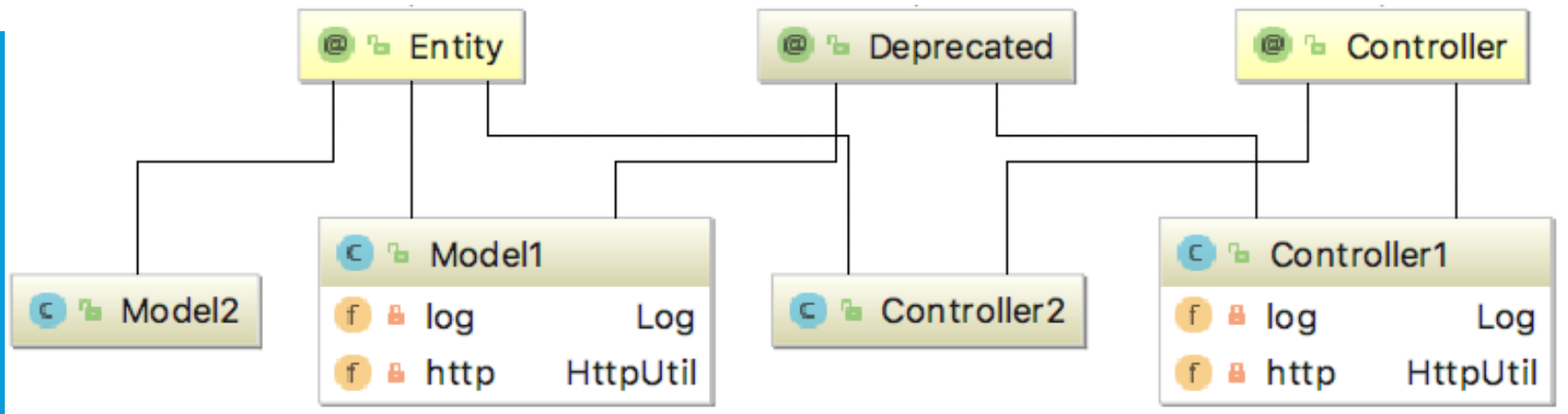
PROBLEM STATEMENT



Log HttpUtil @Entity @Controller @Deprecated

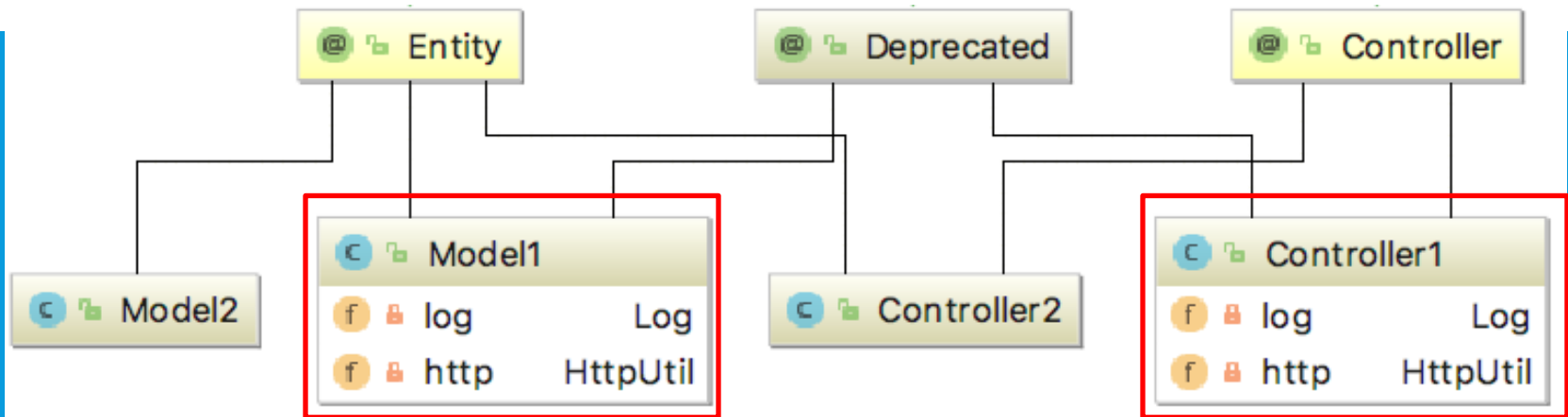
	Log	HttpUtil	@Entity	@Controller	@Deprecated
Model1	1	1	1	0	1
Model2	0	0	1	0	0
Controller1	1	1	0	1	1
Controller2	0	0	0	1	0

PROBLEM STATEMENT



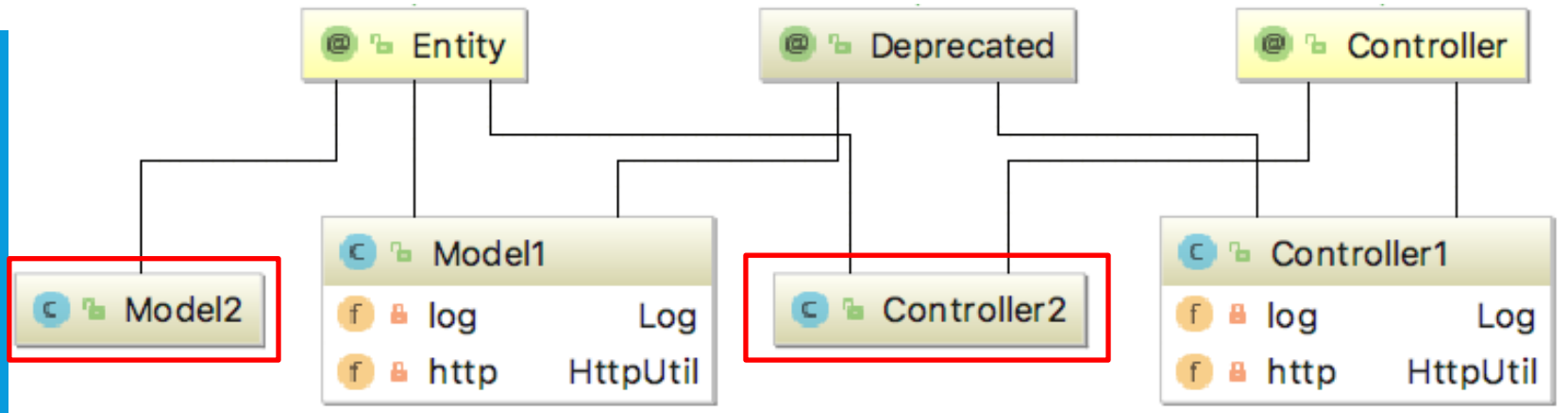
	Model1	Model2	Controller1	Controller2
Model1	-	0.75	0.5	1.25
Model2	-	-	1.25	0.5
Controller1	-	-	-	0.75
Controller2	-	-	-	-

PROBLEM STATEMENT



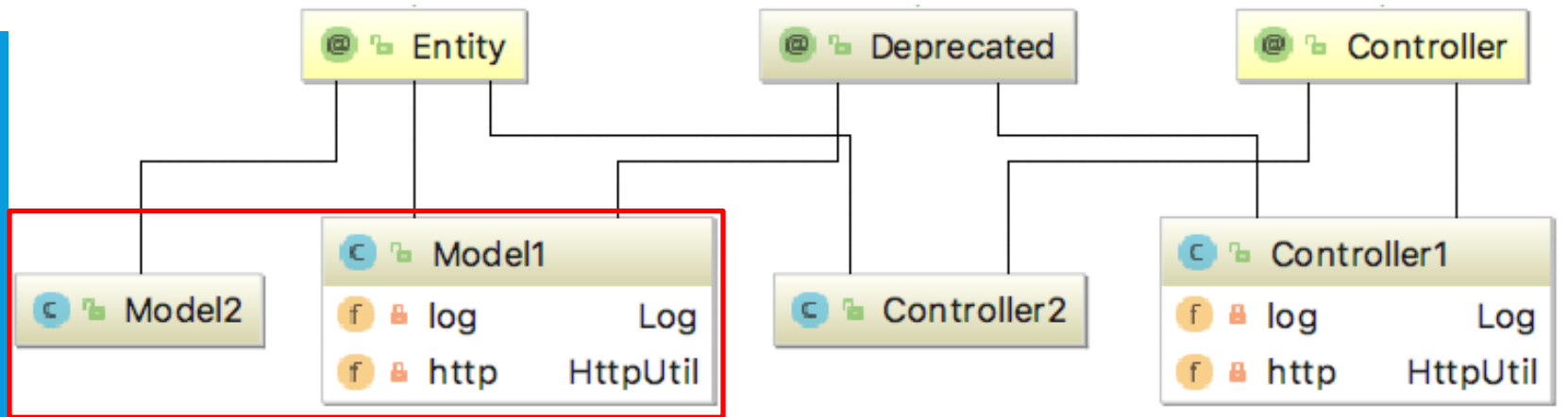
- Module1
 - Model1
 - Controller1
- Module2
 - Model2
 - Controller2

PROBLEM STATEMENT



- Module1
 - Model1
 - Controller1
- Module2
 - Model2
 - Controller2

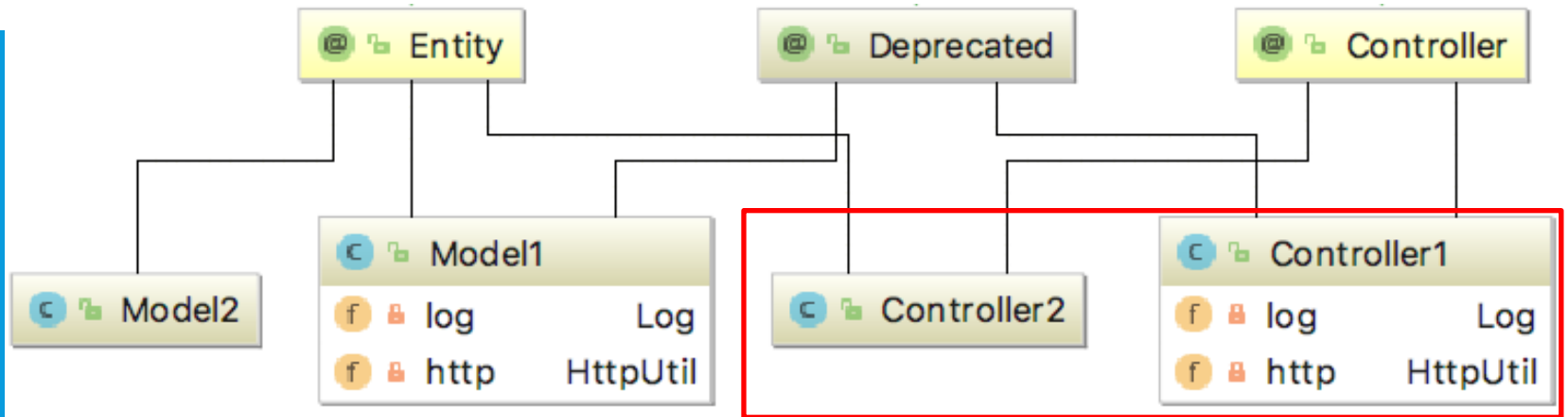
PROBLEM STATEMENT



- Module1
 - Model1
 - Controller1
- Module2
 - Model2
 - Controller2

- Model layer
 - Model1
 - Model2
- Controller layer
 - Controller1
 - Controller2

PROBLEM STATEMENT



- Module1
 - Model1
 - Controller1
- Module2
 - Model2
 - Controller2

- Model layer
 - Model1
 - Model2

- Controller layer
 - Controller1
 - Controller2

Use framework
information

OUR SOLUTION

FRAMEWORK INFORMATION

- Frameworks
 - Spring
 - Struts
 - Hibernate
 - MyBatis
- Annotations : like @Entity
- Interfaces
- Super class
- Class name key words

FRAMEWORK INFORMATION

- Original Feature Vector

$$v_i = (v_i^1, v_i^2, \dots, v_i^n)$$

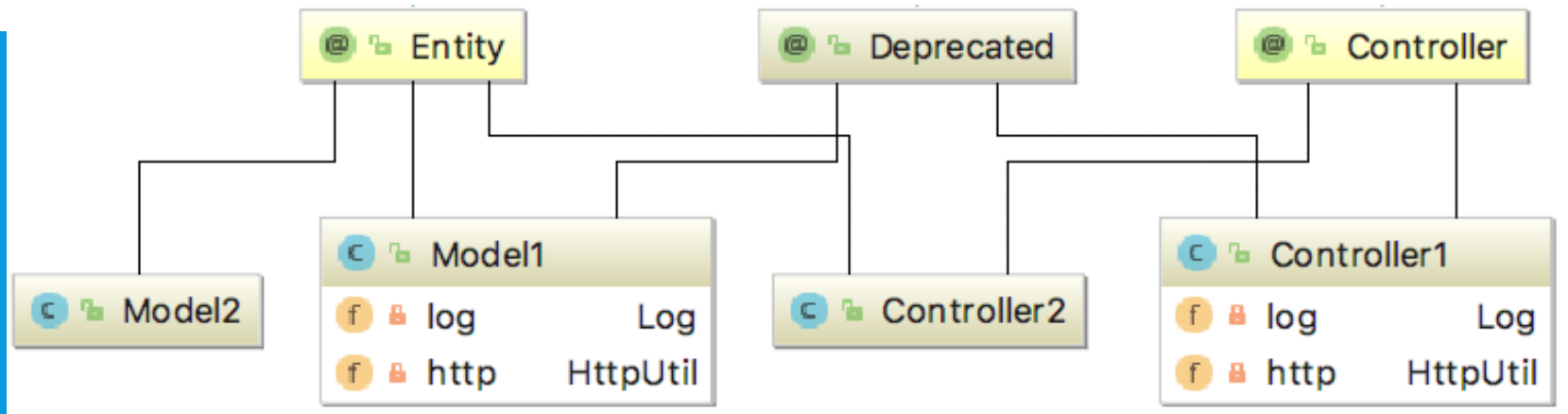
- Framework Feature Weight

$$w_s = (w_s^1, w_s^2, \dots, w_s^n)$$

- Weighted Feature Vector

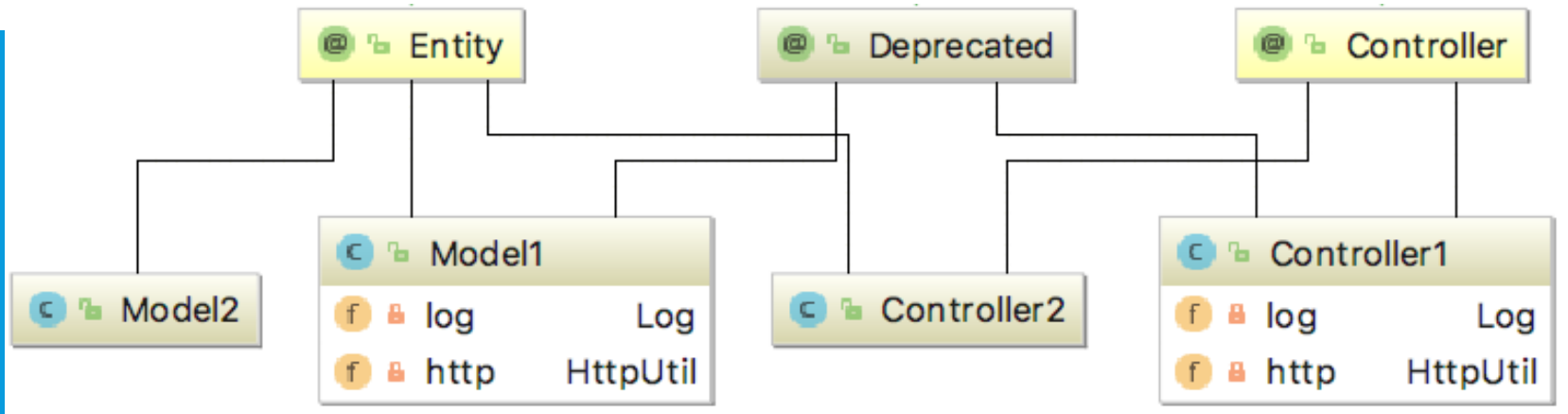
$$wv_i = (w_s^1 v_i^1, w_s^2 v_i^2, \dots, w_s^n v_i^n)$$

USE FRAMEWORK INFORMATION



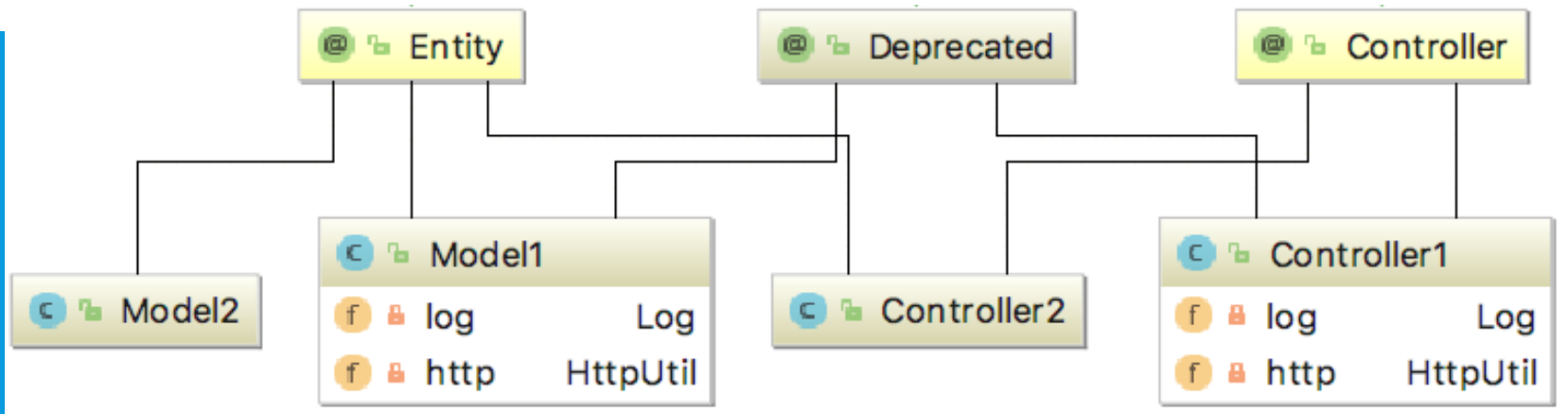
	Log	HttpUtil	@Entity	@Controller	@Deprecated
Model1	1	1	1	0	1
Model2	0	0	1	0	0
Controller1	1	1	0	1	1
Controller2	0	0	0	1	0

USE FRAMEWORK INFORMATION



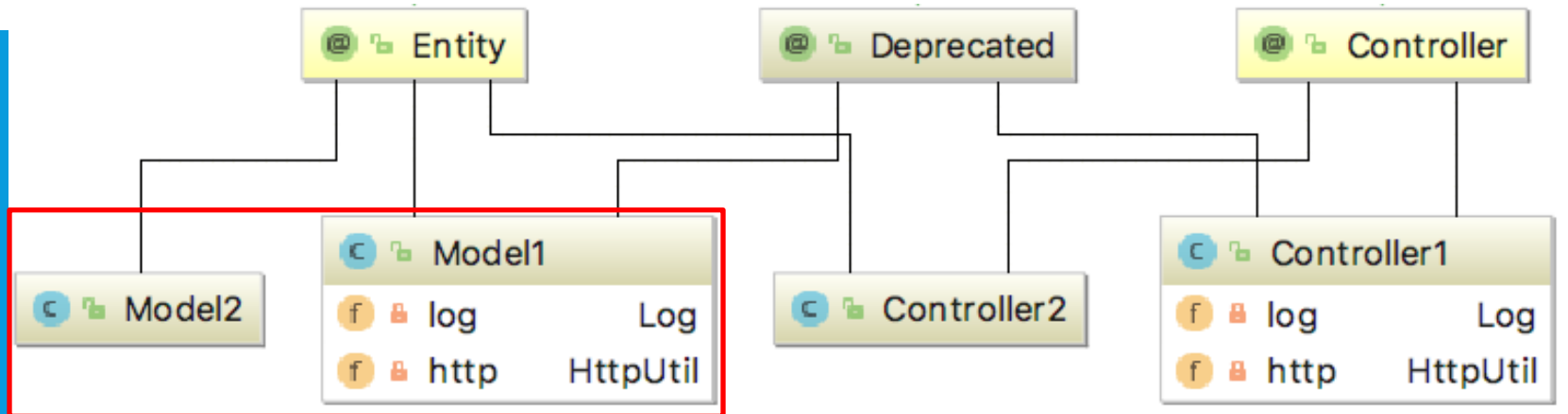
	Log	HttpUtil	@Entity	@Controller	@Deprecated
Model1	1	1	4	0	1
Model2	0	0	4	0	0
Controller1	1	1	0	4	1
Controller2	0	0	0	4	0

USE FRAMEWORK INFORMATION



	Model1	Model2	Controller1	Controller2
Model1	-	0.75	2.00	2.75
Model2	-	-	2.75	2.00
Controller1	-	-	-	0.75
Controller2	-	-	-	-

PROBLEM STATEMENT



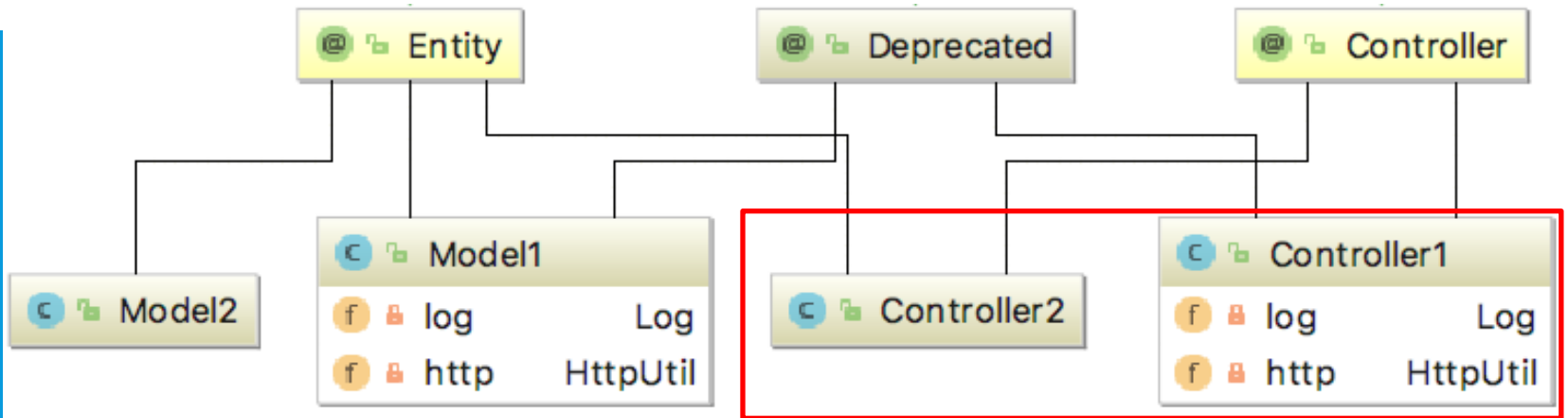
- Model layer

- Model1
- Model2

- Controller layer

- Controller1
- Controller2

PROBLEM STATEMENT



- Model layer

- Model1
- Model2

- Controller layer

- Controller1
- Controller2



EXPERIMENT AND RESULT



EXPERIMENT DESIGN

- Select projects using the Spring Framework on GitHub
- Select framework specific features
- Run the algorithm with different feature weights
- Compare results with original architecture

SELECTED PROJECTS

Project	Modules	File Number	LOC
act	6	39	5158
cms	5	40	4264
gen	5	18	2005
oa	4	16	1669
sys	8	37	5025
SpringBlog	9	45	2100

SELECTED FEATURES

- Usage of @Controller Annotation
- Usage of @RestController Annotation
- Usage of @Service Annotation
- Usage of @Entity Annotation
- Implementation of Serializable Interface
- Implementation of JpaRepository Interface

ACCURACY CALCULATION

- Initial architecture

$$M_{init} = \{M_{init}^1, M_{init}^2, \dots, M_{init}^n\}$$

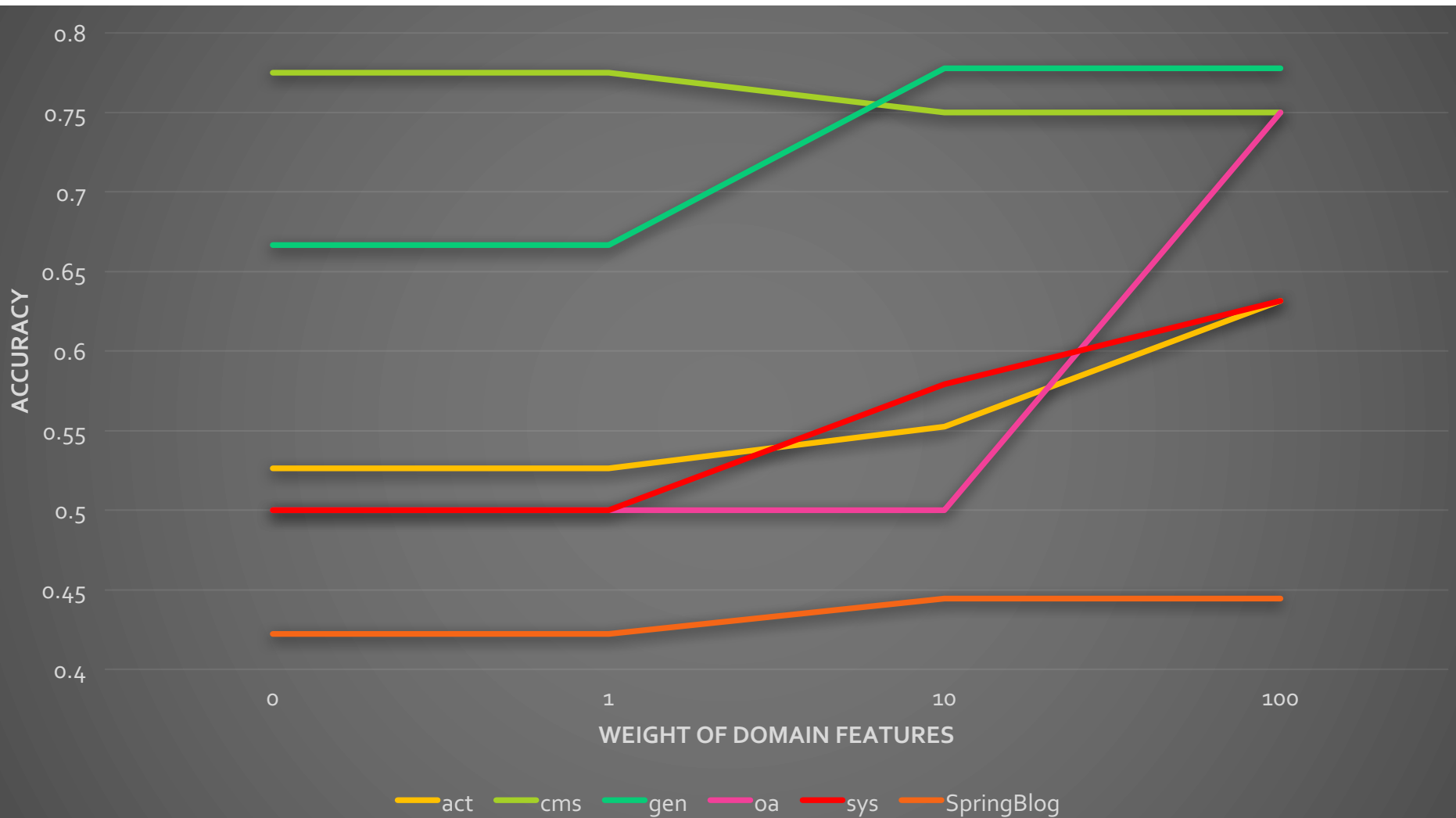
- Recovered architecture

$$M_{new} = \{M_{new}^1, M_{new}^2, \dots, M_{new}^n\}$$

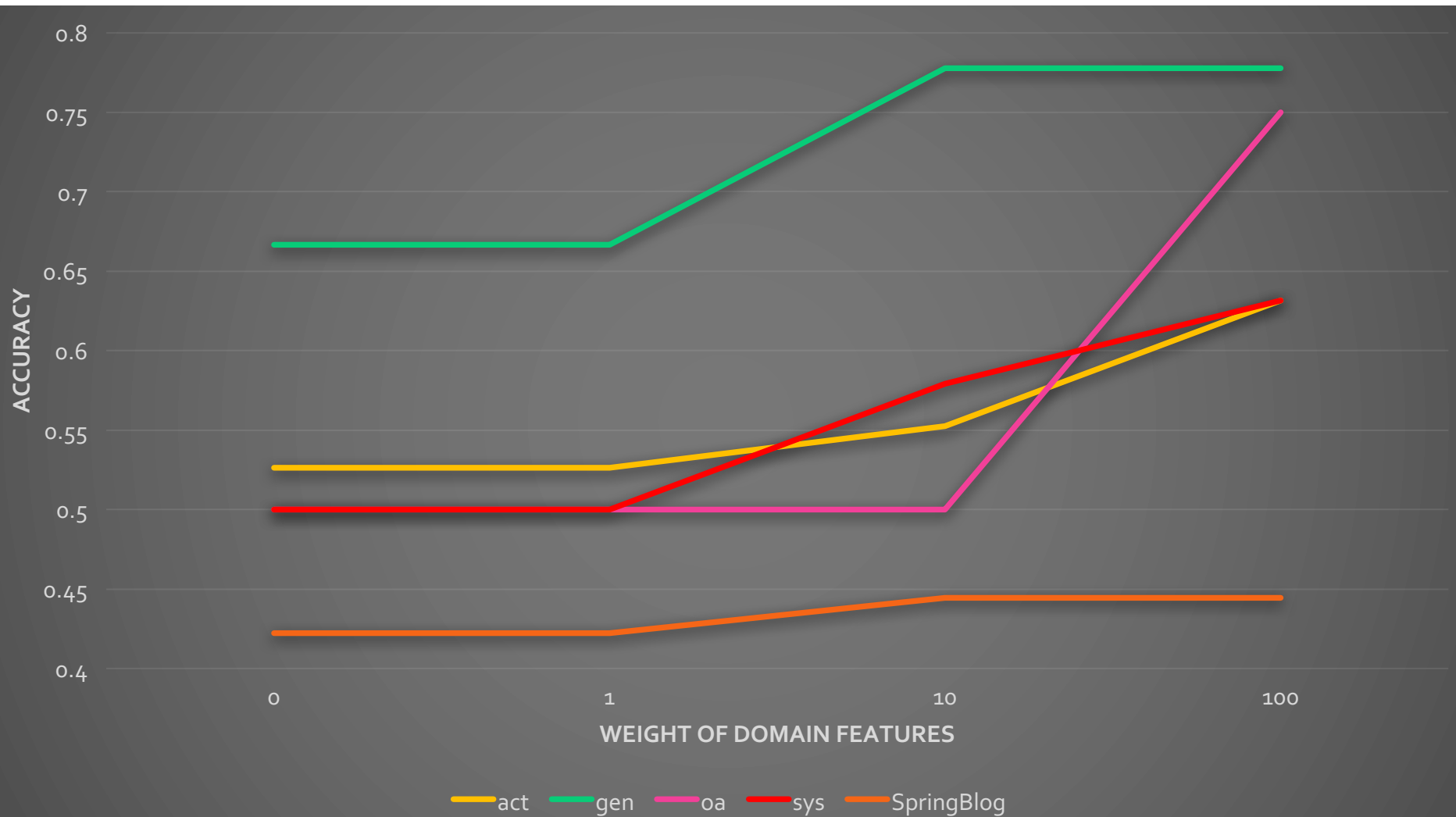
- Accuracy

$$\text{Similarity}[M_{init}, M_{new}] = \frac{\sum_{i=0}^n |M_{init}^i \cap M_{new}^i|}{\sum_{i=0}^n |M_{init}^i|}$$

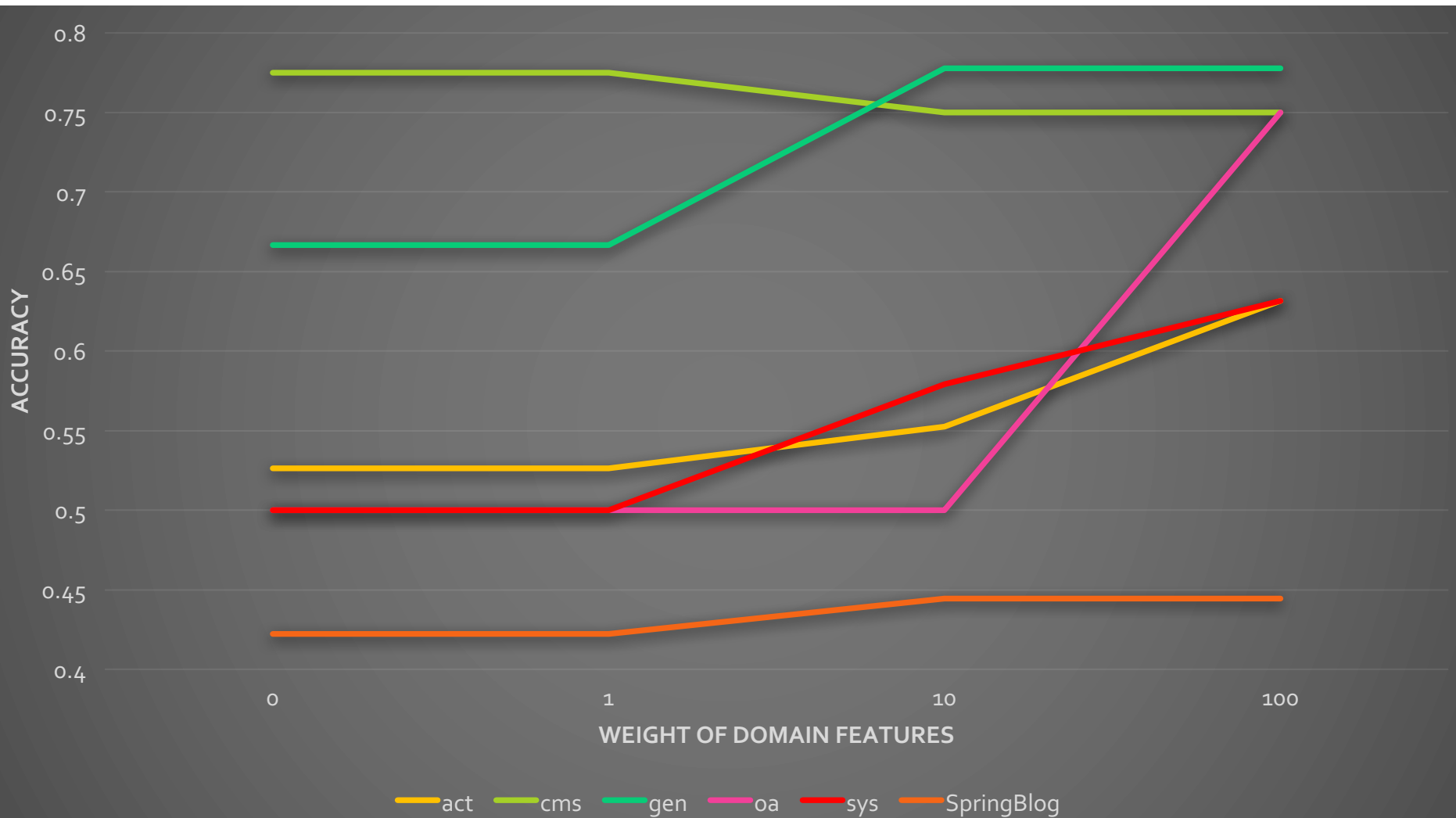
RESULT



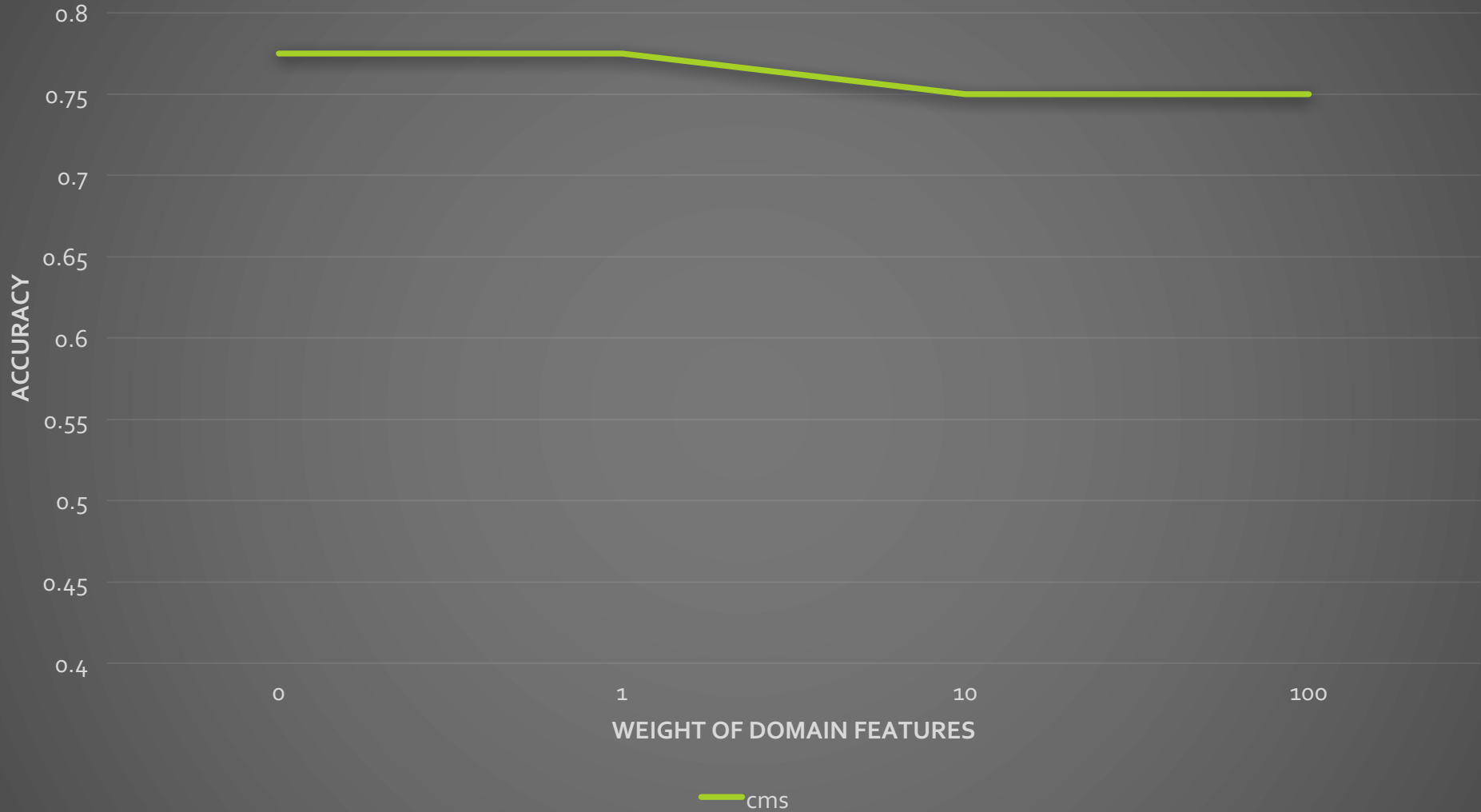
RESULT



RESULT



RESULT





CONCLUSION AND FUTURE WORKS



CONCLUSION & FUTURE WORKS

- Framework information can be used to improve architecture recovery algorithm
- Future works
 - More features should be used
 - Recommend the weight using machine learning
 - Apply the method to other systems
 - Large-scale system should be tested

Q&A