5 Papers

Title	Publication source
Combining Deep Learning with Information Retrieval to Localize Buggy Files for Bug Reports (N)	ASE
Modeling bug report quality	ASE
Fault localization with nearest neighbor queries	ASE
Lightweight fault-localization using multiple coverage types	ICSE
The Probabilistic Program Dependence Graph and Its Application to Fault Diagnosis	TSE

Combining Deep Learning with Information Retrieval to Localize Buggy Files for Bug Reports (N)

汇报人：王昭丹

CONTENTS

01. INTRODUCTION

02. COMBINING MODEL —— HyLoc

03. DNN-BASED RELEVANCY ESTIMATION

04. EMPIRICAL EVALUATION

INTRODUCTION

statistics of **program** analysis information

- semantics of the program and/or its execution information
- with **test cases**

content of a given **bug report**

- information retrieval (IR) / machine learning (ML)
- extracting important features
- from the given bug report and source files

Key Limitation

lexical mismatch between natural language texts in bug reports and technical terms in source code

HyLoc

rVSM

DNN

HyLoc —— Key Design Ideas

Bridge lexical gap

Using DNN to bridge the lexical gap.

Two DNNs

Feature combination

Using another DNN for feature combination.

Dimension reduction

Using another DNN to perform dimension reduction for feature vectors.

1

2

3

Three types of features

- textual similarity feature computed from rVSM
- relevancy feature from the DNNs
- **metadata features** —— bug-fixing recency score for a file

HyLoc Model Architecture

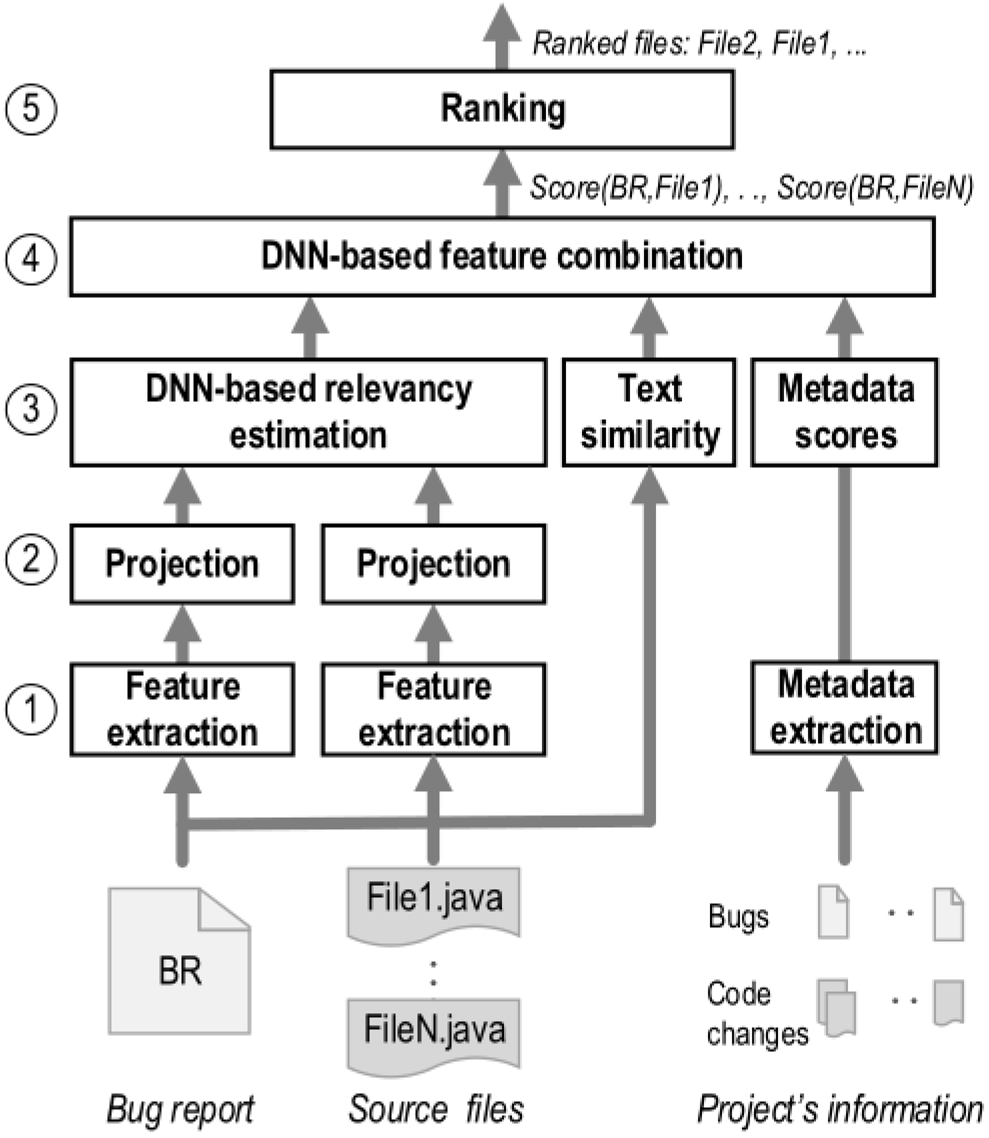
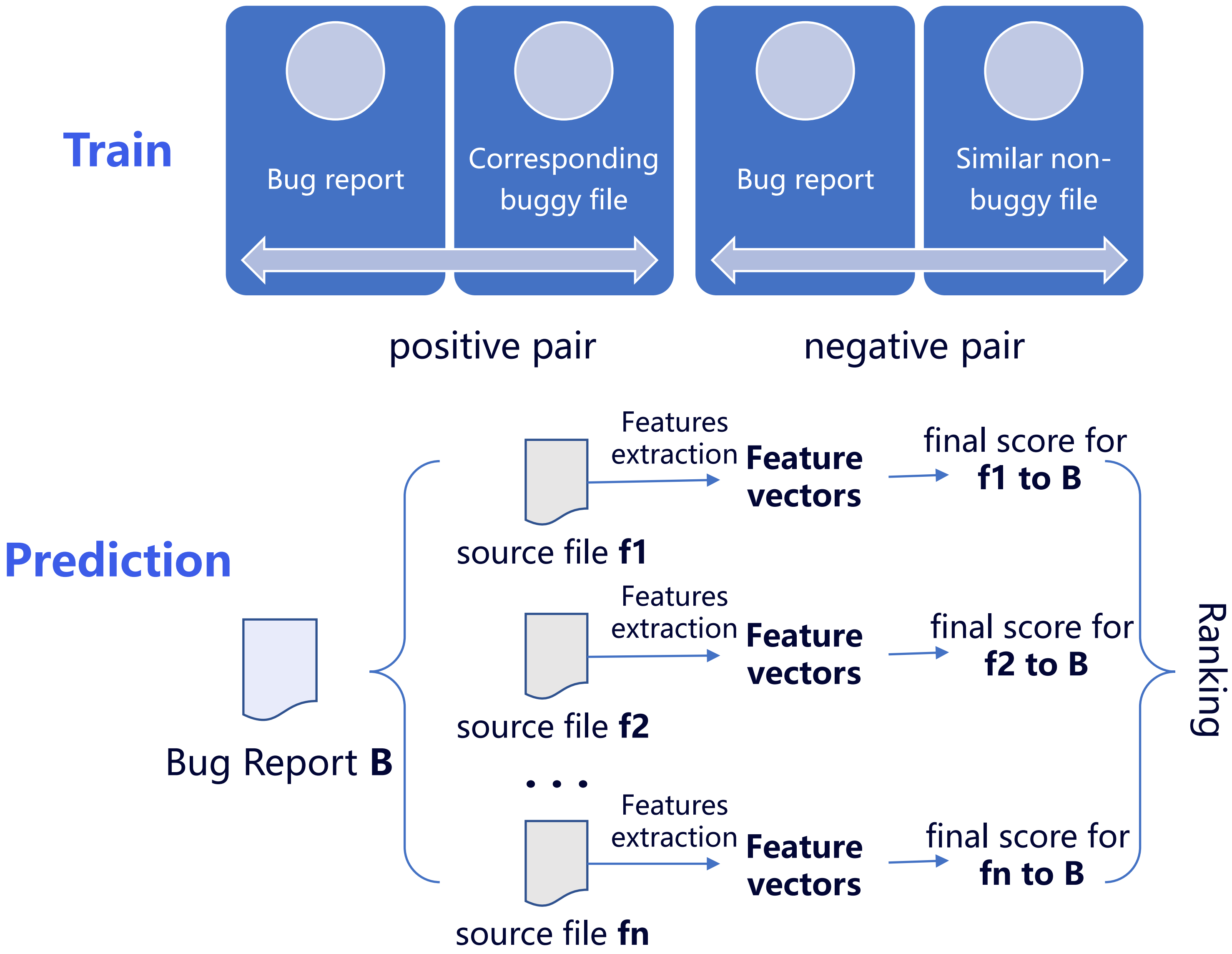


Fig. 1: HyLoc Bug Localization Model



DNN-BASED RELEVANCY ESTIMATION

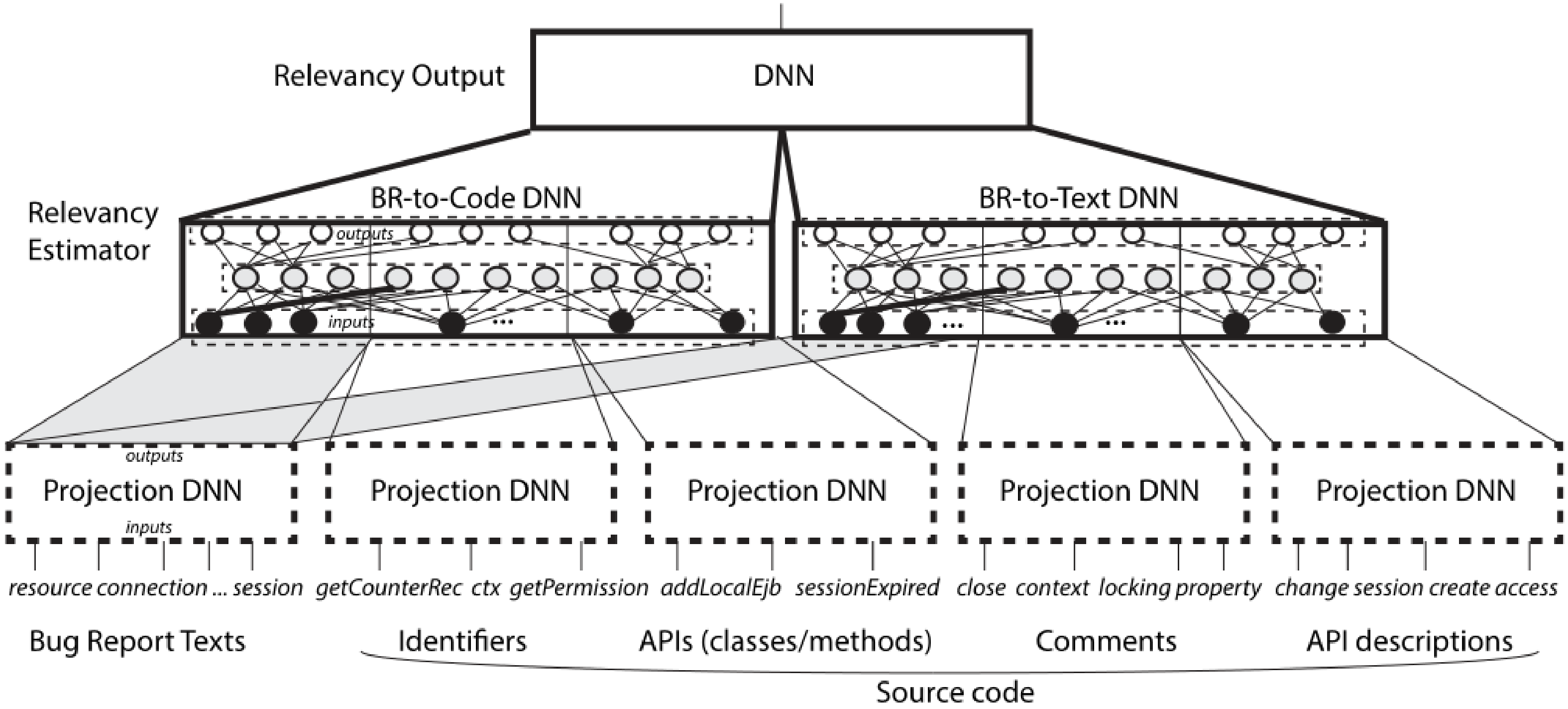


Fig. 2: DNNs for Projection and Relevancy Estimation

Two DNNs

- **Bug Report-to-Code DNN**

  learn the relations for the text features from bug reports and the source code features

- **Bug Report-to-Text DNN**

  learn the relations between the textual features in the bug reports and those in comments and API descriptions

EMPIRICAL EVALUATION

Experimental Setting

- **Benchmark Dataset:** provided by Ye et al.
- **For training:** text similarity measure; top 300 similar files negative samples
- **For prediction:** compute the scores and rank all the files in a project.

Three **metrics** for evaluation

- **Top-ranked accuracy**
- **Mean Average Precision (MAP)**
- **Mean inverse Rank (MRR)**

Sorted the bug reports chronologically by their report **timestamps**.

Divided the bug reports into **10 folds** with equal sizes (fold1 - oldest and fold10 - newest).

Trained a model on **foldi** and **tested** it on **foldi+1**.

EMPIRICAL EVALUATION

Impacts of Components and Parameters on Accuracy

a) Accuracy with Different Components

- DNN by itself does not give high accuracy
- DNN+rVSM > both DNN and rVSM individually

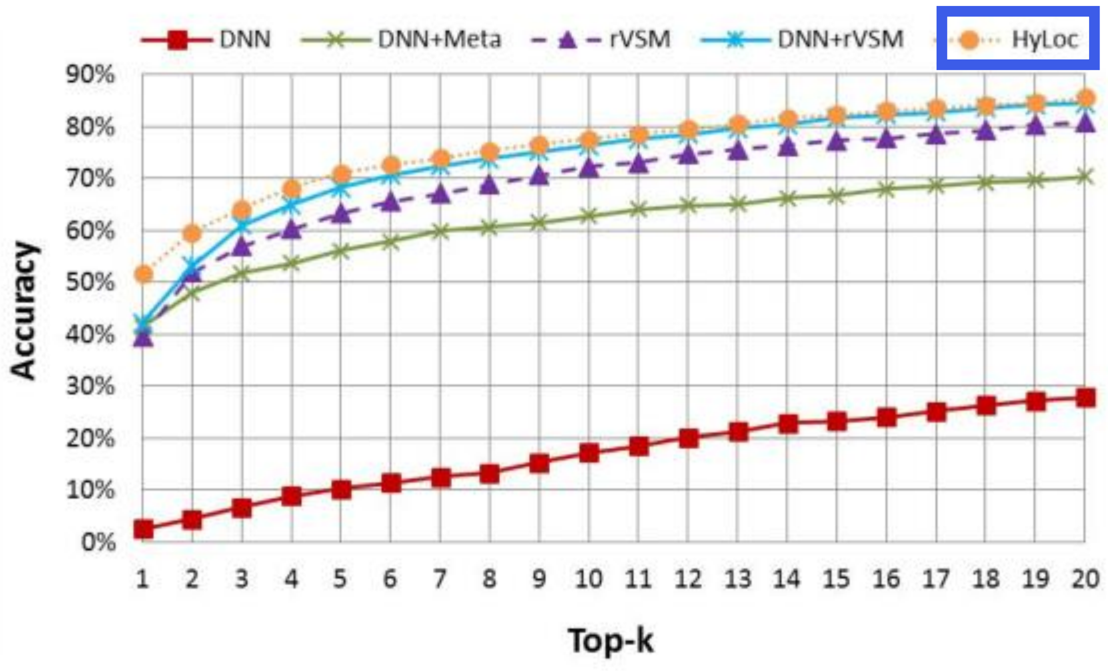


Fig. 3: Top-*k* Accuracy with Different Components

DNN, rVSM vs. DNN+rVSM Models

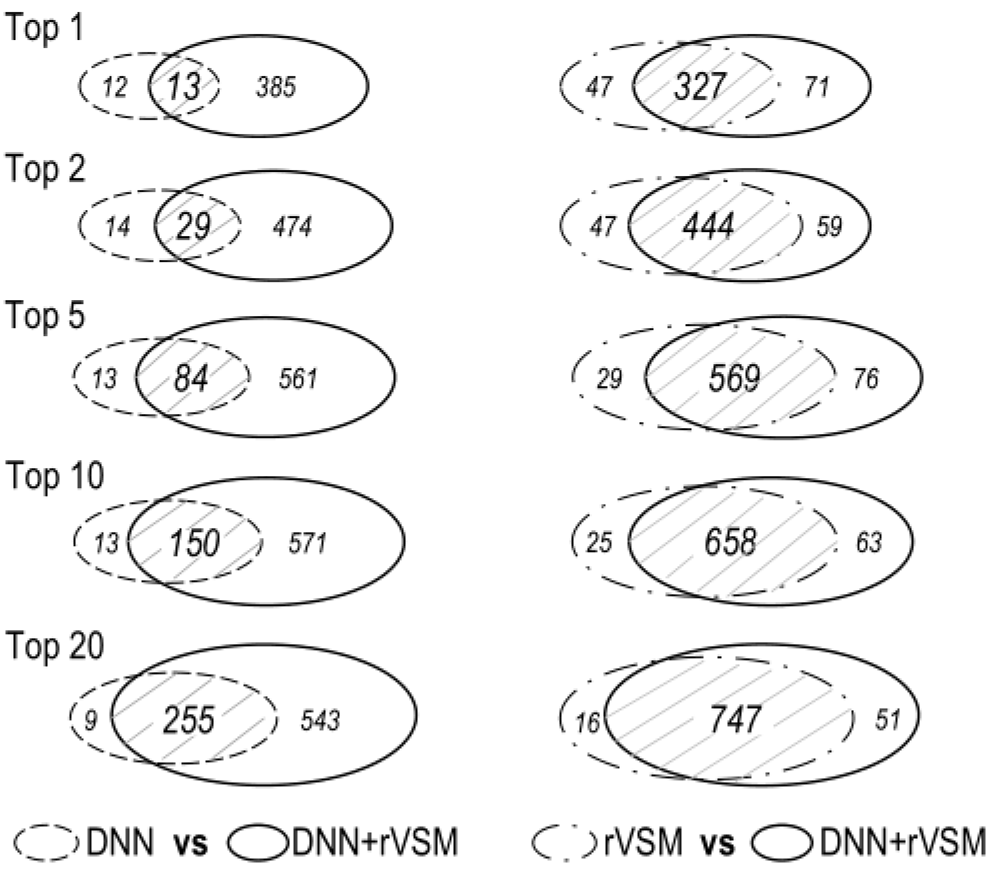


Fig. 4: Venn Diagrams for Correct Results of Approaches

EMPIRICAL EVALUATION

b) Examples of Linking Terms in Two Spaces

1. Each of the textual tokens in all bug reports as the **input**.
2. **Paired** it with each of the code tokens in source files and the textual tokens in comments/documentations.
3. **DNN** to compute the score **output** for each of such pairs.
4. Produced **the list of "relevant" tokens** in source files ranked by the relevancy score between two tokens.

c) Feature combination comparison

- compare feature combination by DNN and by learn-to-rank approach
- built another experimental model — — LRCombine
- HyLoc higher than LRCombine **22%** at top-1 accuracy, at top-5 is **10%**

non-linear combination better linear one

TABLE II: Linking Terms in Reports and Terms/Tokens in Files

BR	Token 1	Token 2	Token 3	Token 4
context	authorization	ctx	envCtx	asyncContext
resource	virtualClasspath	changeSessID	setSecureClass	addLocalEjb
writer	globalCacheSize	charset	index	charsWritten
read	headerLength	InternalBuffer	readBytes	dir

EMPIRICAL EVALUATION

Accuracy Comparison

Compare HyLoc to the stateof-the-art approaches:

- Naive Bayes (NB) approach —— ML
- LR (learn-to-rank) approach —— hybrid
- BugLocator —— IR-based

TABLE III: Accuracy Comparison

System	Model	1	2	3	4	5	10	15	20	MRR	MAP
TomCat	HyLoc	51.6	59.6	64.1	68.3	71.0	77.6	82.2	85.6	0.60	0.52
	LR	46.2	54.2	59.8	62.3	66.5	74.7	80.1	82.1	0.55	0.49
	BL	35.5	48.7	52.9	58.7	61.8	71.1	77.3	80.2	0.48	0.43
	NB	5.2	6.9	8.3	8.8	9.0	11.9	14.5	16.6	0.08	0.07
AspectJ	HyLoc	40.9	51.8	58.6	61.7	65.7	75.9	79.9	82.7	0.52	0.32
	LR	20.2	32.1	38.5	41.3	45.5	61.1	68.2	70.9	0.33	0.25
	BL	20.1	30.5	40.1	43.3	47.7	57.0	62.1	67.6	0.32	0.22
	NB	4.2	8.0	11.3	11.7	16.0	21.1	26.3	28.2	0.10	0.07
Birt	HyLoc	19.1	24.8	29.7	33.5	36.0	44.6	51.1	55.4	0.28	0.20
	LR	12.4	18.1	22.5	25.1	27.9	37.3	42.4	46.0	0.20	0.15
	BL	11.1	16.2	20.0	22.4	24.9	32.1	37.0	40.6	0.18	0.14
	NB	2.9	4.7	6.5	7.9	8.7	13.8	15.9	17.6	0.06	0.05
Eclipse	HyLoc	40.3	49.8	55.7	60.4	63.5	72.5	77.3	80.3	0.51	0.41
	LR	36.5	47.0	52.0	58.0	60.1	70.7	75.3	79.1	0.47	0.40
	BL	26.5	34.9	40.3	44.8	49.3	60.1	67.3	70.2	0.37	0.31
	NB	3.8	6.1	8.3	9.6	10.6	14.7	16.8	18.3	0.07	0.06
JDT	HyLoc	33.3	44.1	51.0	55.6	59.0	68.5	73.4	76.7	0.45	0.34
	LR	30.0	40.3	48.2	51.1	55.2	68.1	72.4	77.6	0.42	0.34
	BL	19.1	26.2	31.6	37.4	40.2	51.2	57.7	61.3	0.30	0.23
	NB	6.6	9.7	11.8	13.6	15.0	20.0	22.9	25.2	0.11	0.08
SWT	HyLoc	31.0	42.8	51.9	57.6	62.1	74.2	80.3	84.3	0.45	0.37
	LR	28.3	39.4	47.9	52.7	58.2	70.0	76.8	80.0	0.41	0.36
	BL	19.3	24.5	30.0	34.4	38.3	51.1	58.5	64.4	0.28	0.25
	NB	7.4	11.8	14.9	17.0	19.0	26.9	31.3	35.3	0.14	0.11

EMPIRICAL EVALUATION

Time Efficiency

Training time is large for a solution involving one thread to run DNN.

- Parallel computing infrastructures for DNNs
- Incremental training techniques

Predicting time is reasonable.

TABLE IV: Training and Predicting Time in Minutes

System	Training for one fold		Predicting for one report	
	Max	Average	Max	Average
Tomcat	70	65	1.5	1.0
AspectJ	70	70	4.1	2.4
Birt	98	84	4.5	3.1
Eclipse	120	90	3.8	2.1
JDT	122	94	4.8	3.3
SWT	95	83	2.4	1.8

学 习 进 展 & 暑 期 计 划

感谢您的聆听

汇报人：王昭丹