A Machine Learning-based Impact Analysis Tool and its Improvement Using Co-occurrence Relationships

Teppei Kawabata, Tsuyoshi Nakajima   Shuichi Tokumoto, Ryota Tsukamoto, Kazuko Takahashi

Shibaura Institute of Technology,   Information Technology R&D Center, Mitsubishi Electric Corporation

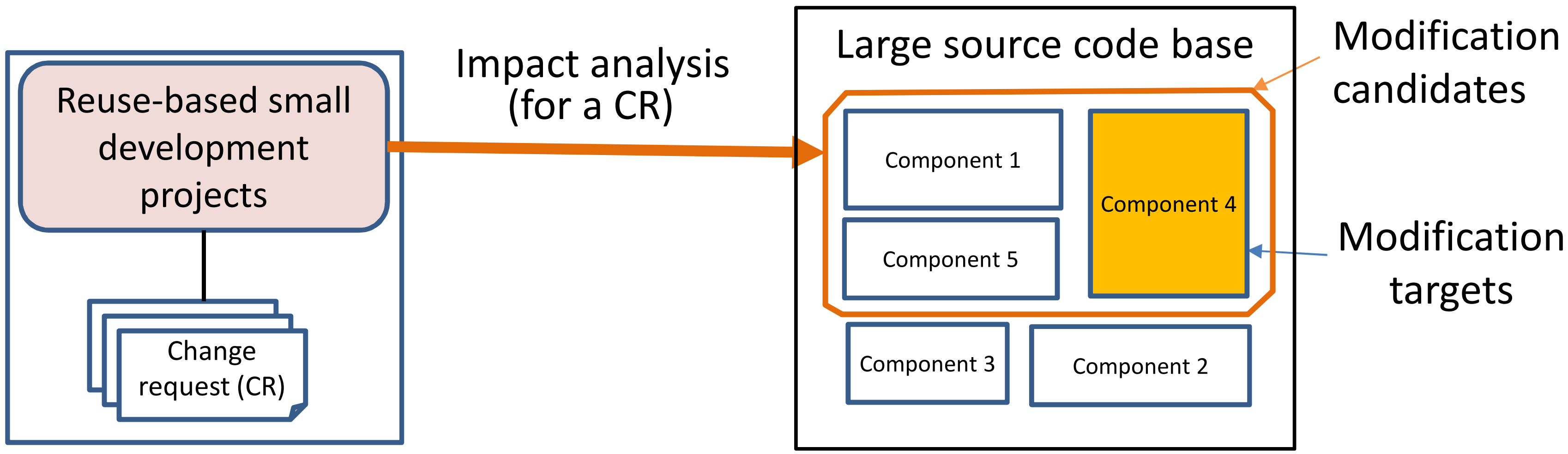
Table of Contents

中島研
Software Engineering Lab

1. Conventional impact analysis methods and their problems
2. Proposed impact analysis method using machine learning
3. Four proposed algorithms in machine learning considering multilabel classification
4. For a comparative evaluation of the above four algorithms

Background: Importance of impact analysis

Software change impact analysis plays an important role in controlling software evolution in the maintenance of continuous software development.

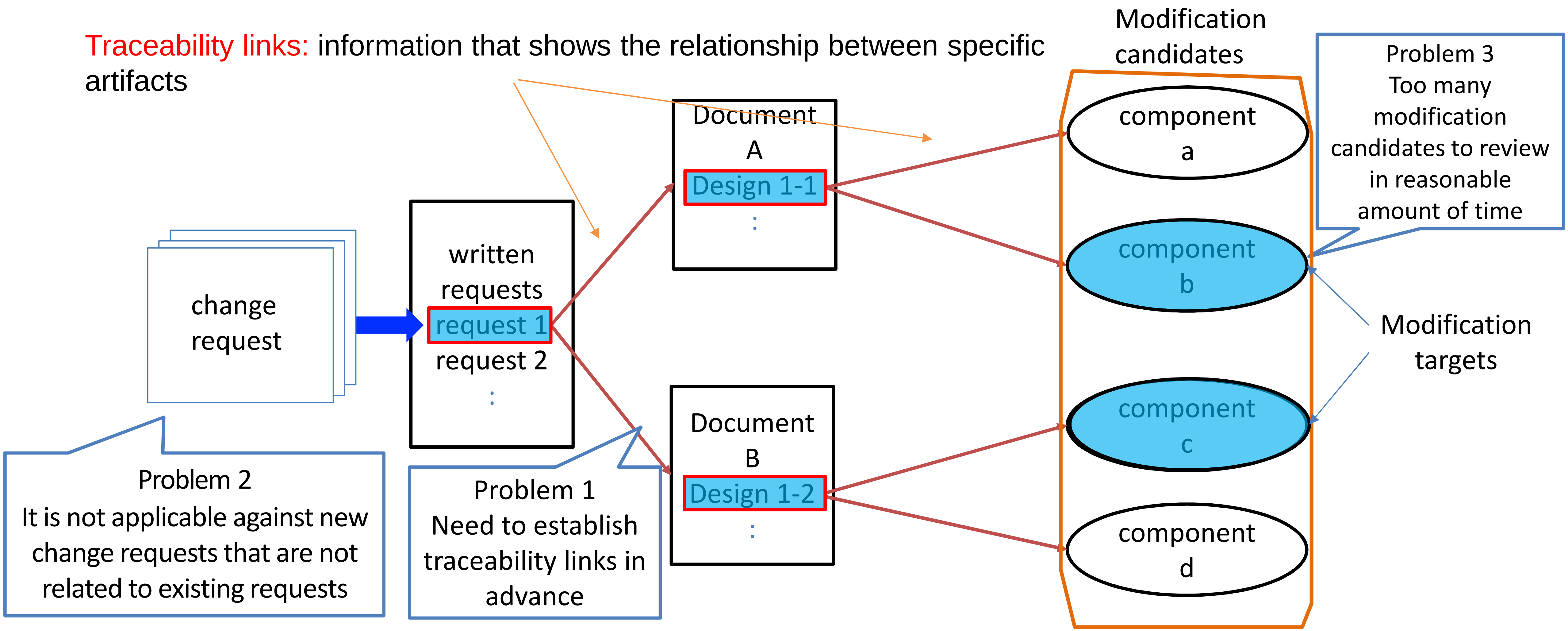


- It is important to improve the accuracy and efficiency to obtain modification candidates.
  This is because it is difficult to automate determining whether a modification candidate is really a modification target or not, requiring a lot of efforts.
- However, the problem is that it depends on the amount of developer's knowledge about the source code base.

Conventional method: Impact analysis with traceability

Traceability: established linkage between multiple deliverables in the development process[1]

Traceability links: information that shows the relationship between specific artifacts



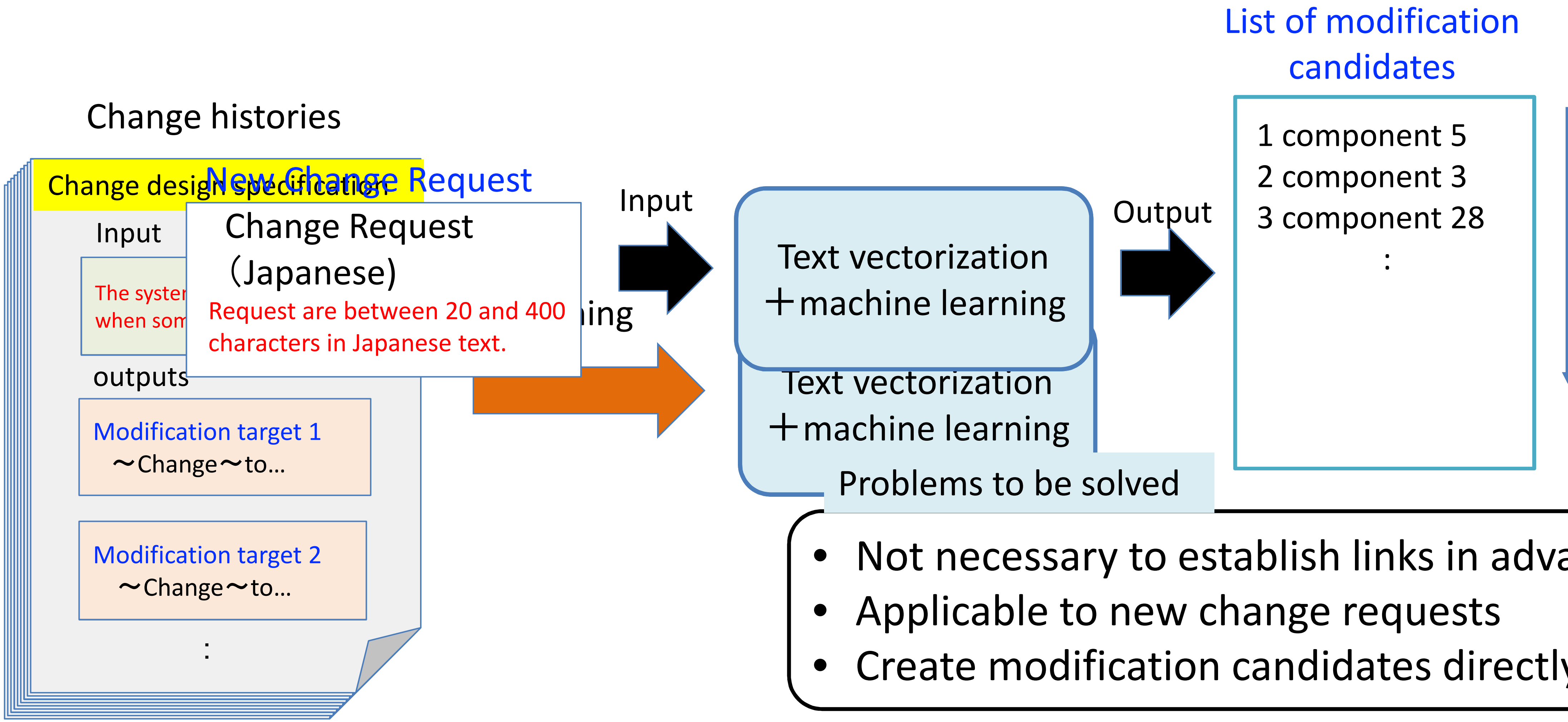
[1]Udagawa Y.,et al. Traceability in Information System Development Standards: A Case Study and Its Future. IPSJ,2010,51.2.:150-158

Proposed method: Learning from change histories.

Software Engineering Lab

Our method is:

- To learn from a large number of change histories from past projects, and
- To create modification candidates from a change request.

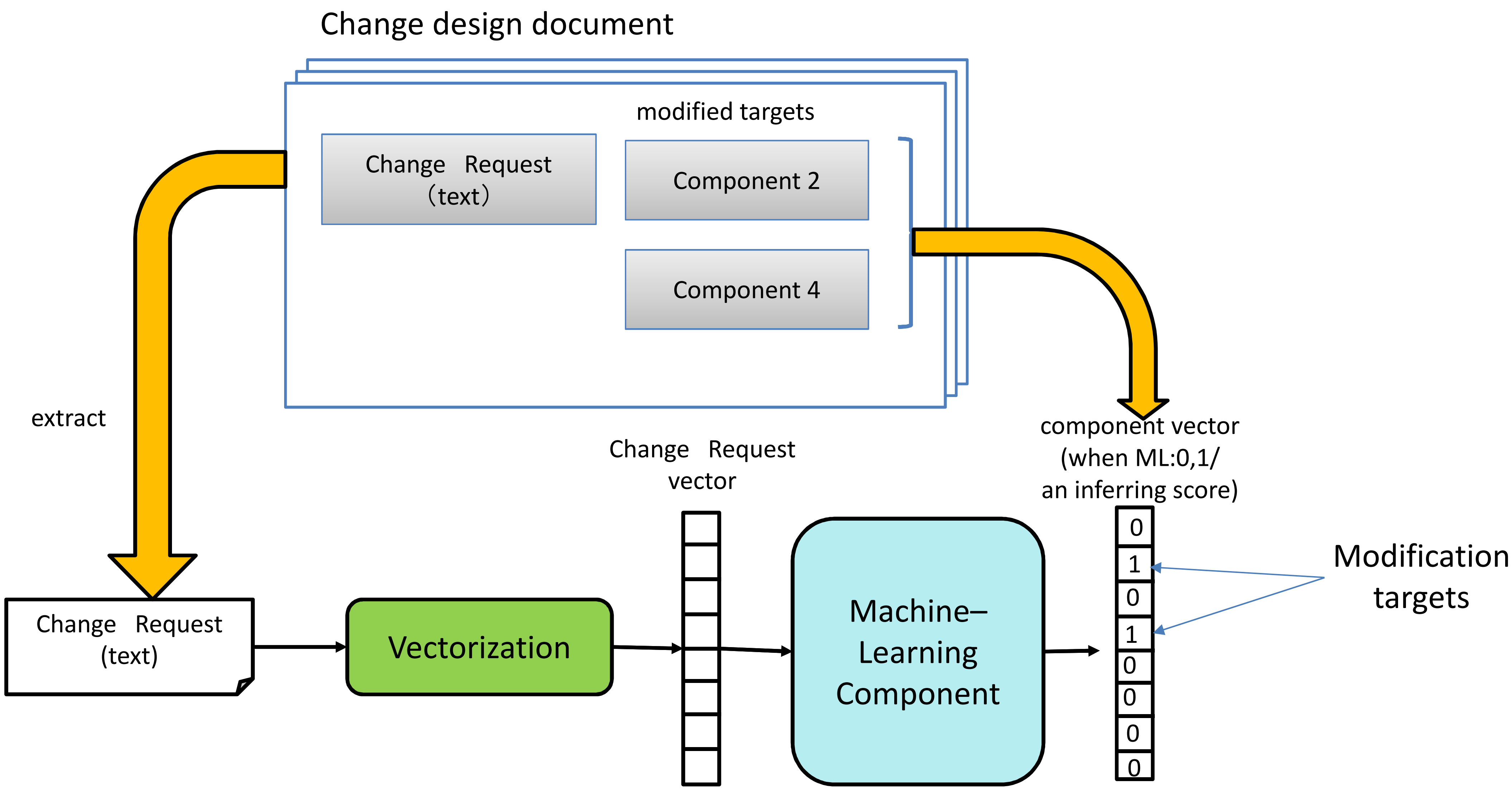


Problems to be solved

- Not necessary to establish links in advance
- Applicable to new change requests
- Create modification candidates directly

Proposed method: composition of the algorithm

中島研
Software Engineering Lab



Proposed method: How to implement sentence vectorization

中島研
Software Engineering Lab

Vectorizing steps — Possible choices

1 Word extraction
- Full morphological selection
- Extract nouns only
- Selection by developer (Weighting)

2 Word vectorization
- word2vec

3 Vector association
- simple average
- weighted average
- doc2vec

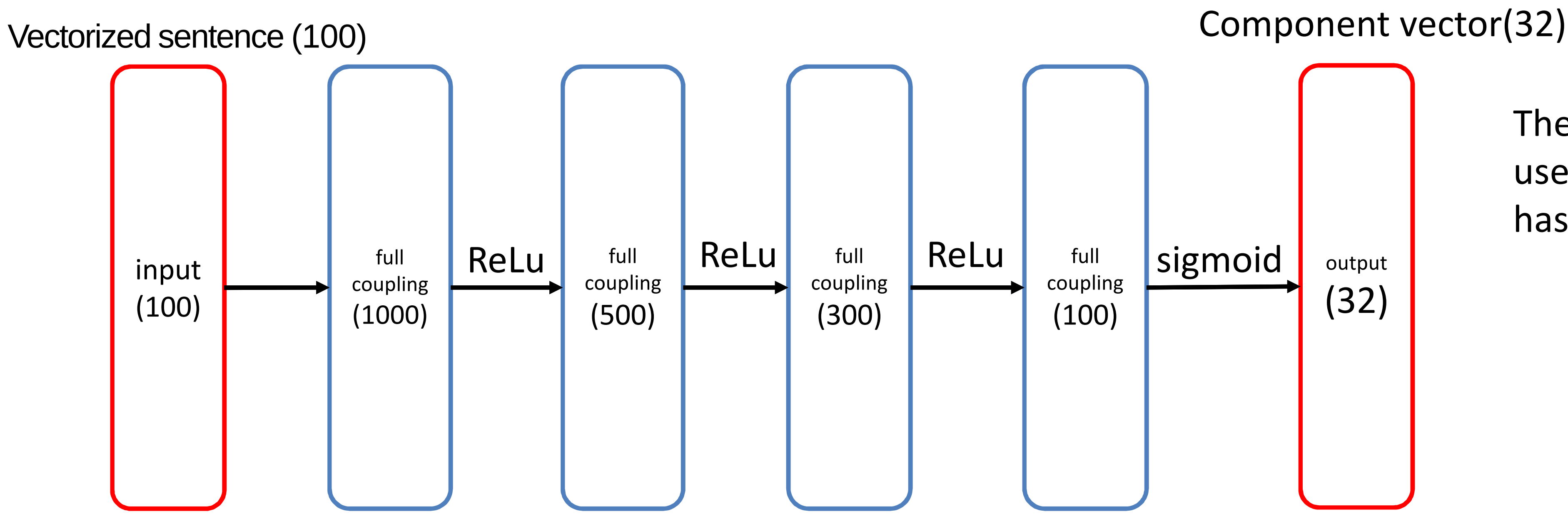
Three implementations were evaluated

	1. Word extraction	2. Word Vectorization	3. Vector association
Implementation 1	noun only	word2vec	simple average
Implementation 2	All	word2vec	doc2vec
Implementation 3	noun only	word2vec	doc2vec

Previous study: Neural Network as the machine learning component

Software Engineering Lab

Configuration of the NN



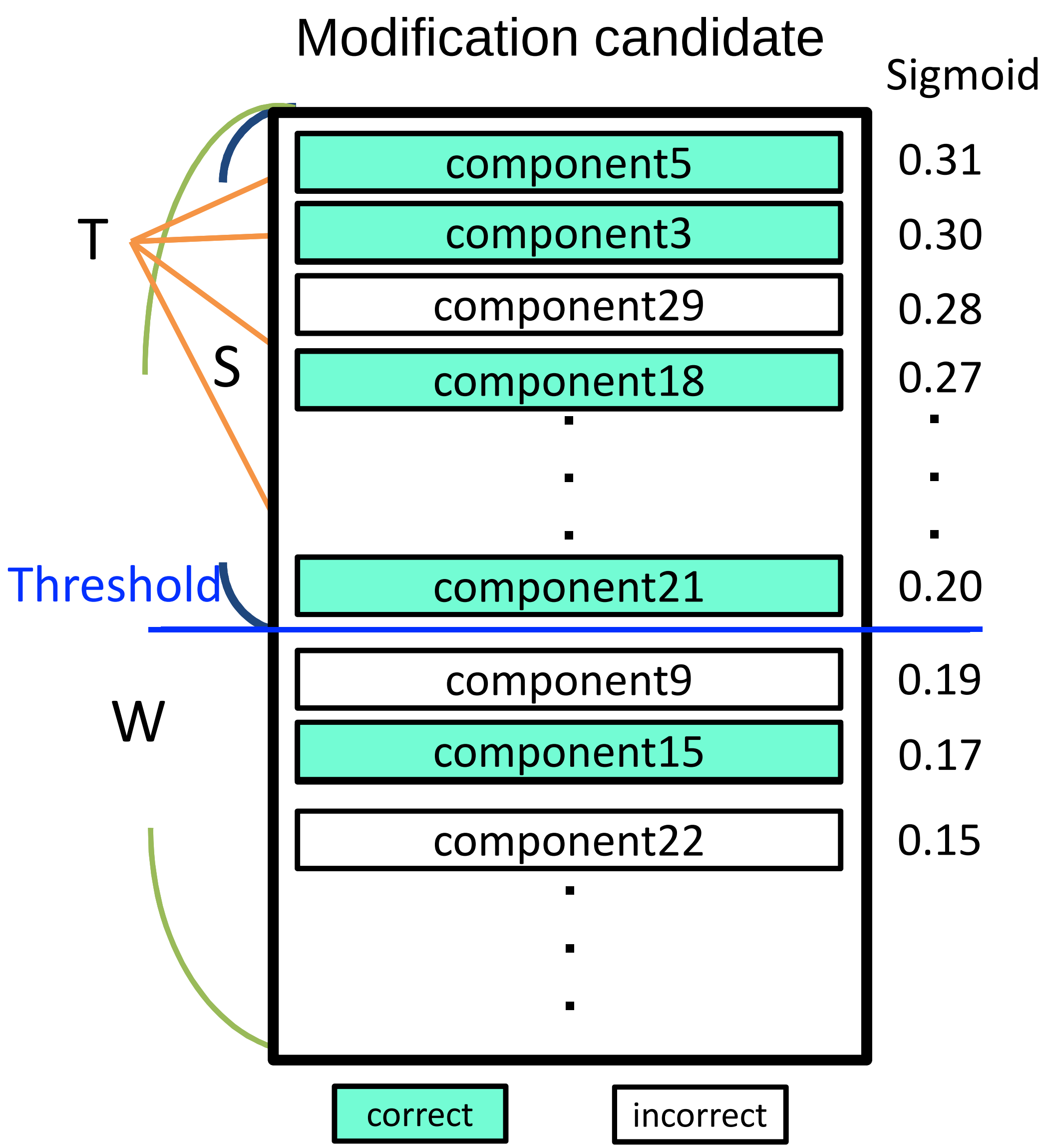
The source code base used in our experiments has 32 components

Hyper Parameters

- Number of studies performed: 50
- Batch size: 50
- Learning rate: 0.1
- Loss function: binary cross-entropy error
- Weight parameter update method: SGD

[2] Y. Iwasaki, Proposal of a system that recommends candidate program changes from requirement text by learning past change, 2020

Evaluation Methods and the results of the previous study



We defined three indexes for the given threshold of Sigmoid value.

A) Candidate Range ratio

$$A = \frac{S}{W}$$

B) Accuracy in the candidate range

$$B = \frac{(S \cap T)}{S}$$

C) Missing rate

$$C = \frac{((W - S) \cap T)}{T}$$

The results of the previous study.

Threshold	A)Accuracy in the candidate range	B)Percentage of correct answer	C)Missing rate
0.06	30.0%	35.0%	23.0%

Missing modification targets has serious consequences.

Our idea to reduce missing rate

Hypothesis

A specific change pattern may cause modification of the same combination of components

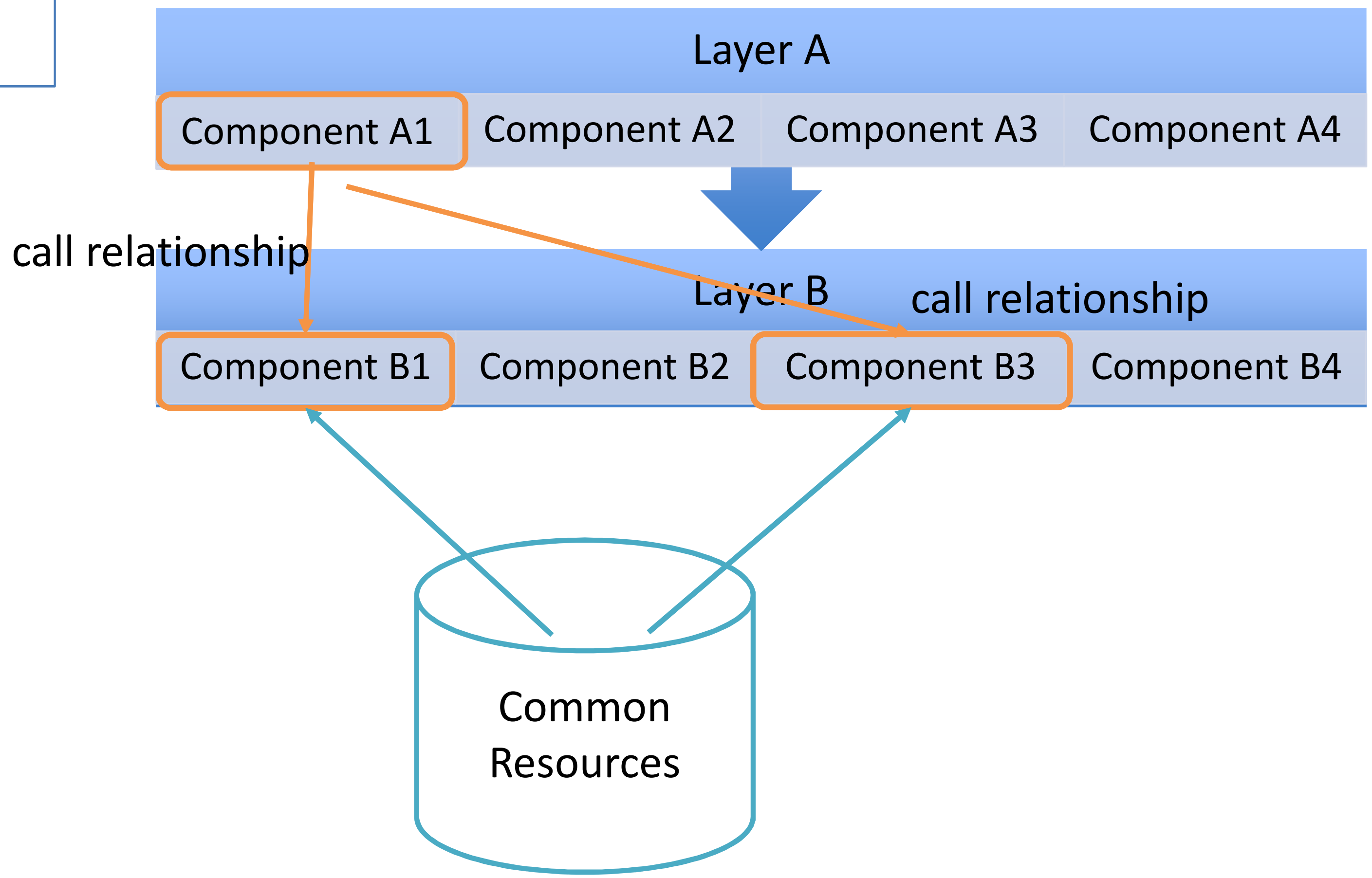
Rationale

From the architectural point of view, some components may use common resources, or some call relationships exists between layers.

Idea for improvement

Adopting multi-label classifiers that model the co-occurrence relationship

Dependencies arising from architecture



The four algorithms implementation to be evaluated

- Previous study
  - ➢Neural Network(NN)

- Basic Methods for Handling Multilabel Classification
  - ➢Binary Relevance（BR）method

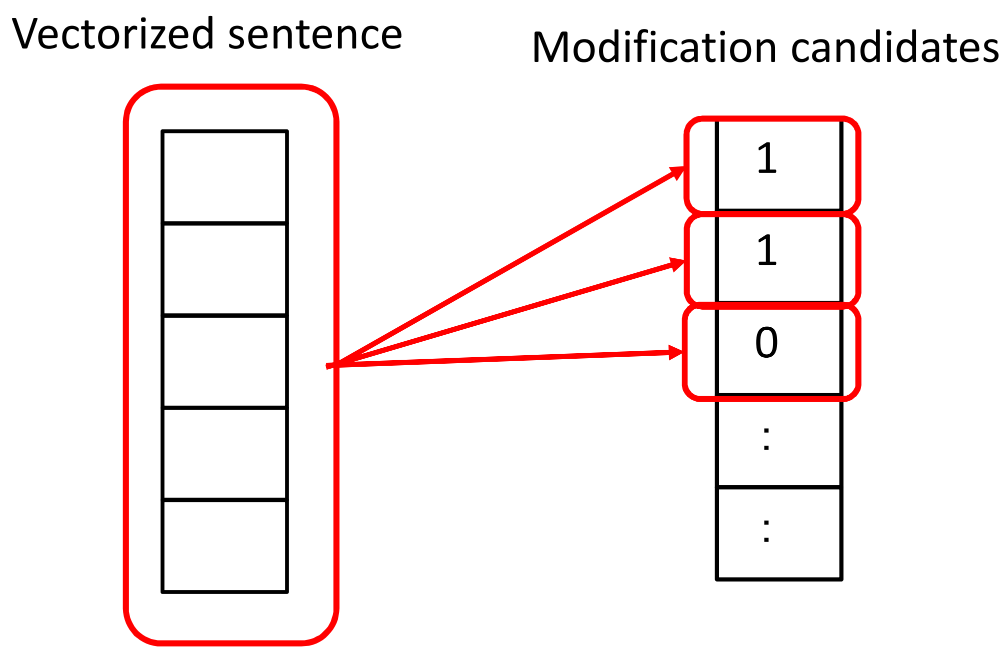
- Methods modeling co-occurrence relationships
  - ➢Label Powerset（LP）method
  - ➢Random k-Labelsets（RAkEL）methods

Basic Methods for Handling Multilabel Classification

Software Engineering Lab

Binary Relevance（BR）method

- Binary Relevance (BR) is a multilabel classification method, which learns a binary model for each label independently of the rest.
- This method does not model the co-occurrence relationships.



The four methods evaluated in this paper

- Previous study
  - Neural Network(NN)

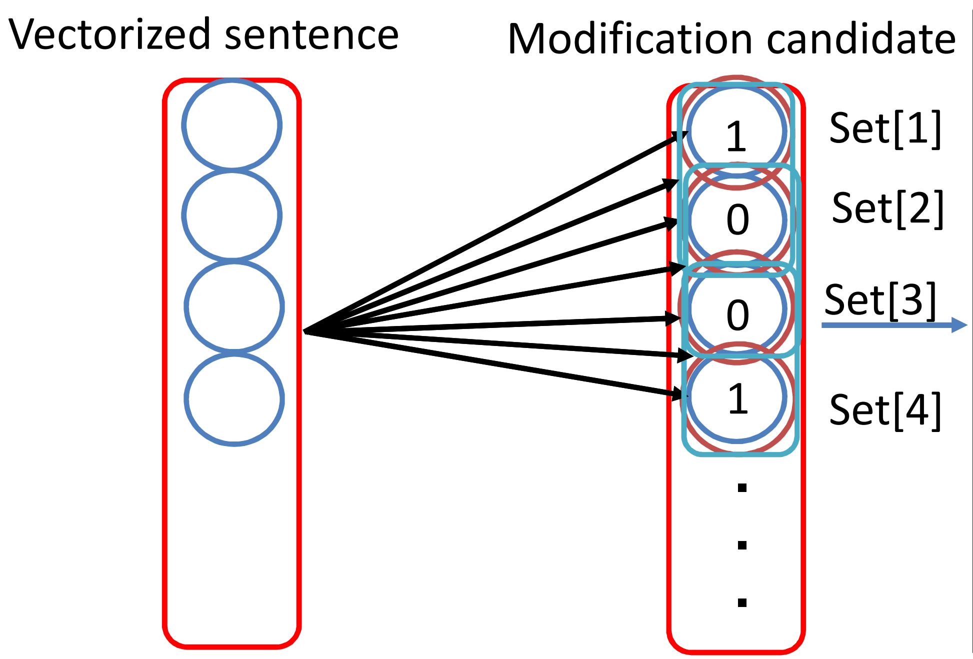
- Basic Methods for Handling Multilabel Classification
  - Binary Relevance（BR）method

- Methods using co-occurrence relationships
  - Label Powerset（LP）method
  - Random k-Labelsets（RAkEL）methods

中島研
Software Engineering Lab

Label Powerset（LP）method

- LP is a multilabel classification method that models the co-occurrence relationship, considering all distinct combinations of labels as a different class and conducting a single-label classification for each.



class	Probability of occurrence of class	Estimation Results for each label in the class				
		label1	label2	・・・	label31	label32
set[1,2]	0.1	1	0	・・・	-	-
set[31,32]	0.2	-	-	・・・	1	0
:	:	:	:	:	:	:
set[1,2..,32]	0.3	1	0	・・・	0	1
Total evaluation		0.1*1 + 0.3*1	0	・・・	0.2*1	0.3*1

Disadvantage large amount of calculation and over-learning

The four methods evaluated in this paper

中島研
Software Engineering Lab

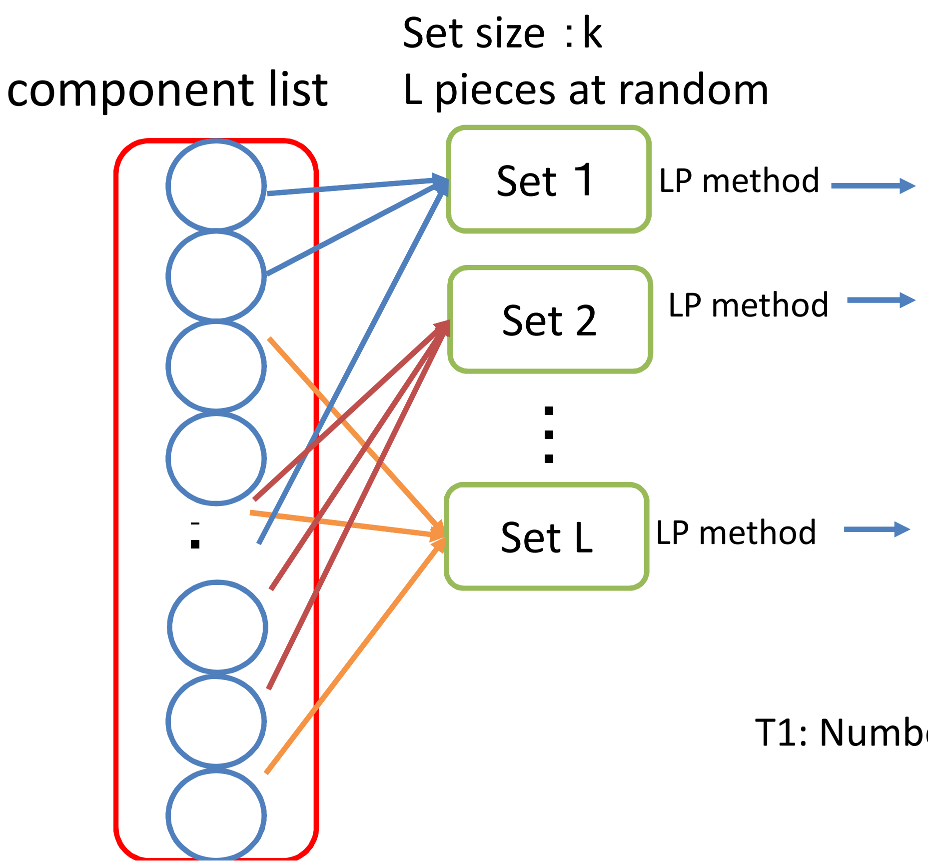
- Previous study
  - Neural Network(NN)

- Basic Methods for Handling Multilabel Classification
  - Binary Relevance（BR）method

- Methods using co-occurrence relationships
  - Label Powerset（LP）method
  - Random k-Labelsets（RAkEL）methods

Random k-Labelsets（RAkEL）methods

- RAkEL is a multilabel classification method that models the co-occurrence relationship, breaking the initial set of labels into a number of small random subsets, called labelsets and employing LP to train a corresponding classifier.



Class	Estimation Results for each Label				
	label1	label2	・・・	label31	label32
set[1,31]	1	-	・・・	0	-
set[2,31]	-	0	・・・	1	-
:	:	:	:	:	:
set[1,2,..32]	0	0	・・・	-	1
Total evaluation	T_1/M_1	T_2/M_2	・・・	T_{31}/M_{31}	T_{32}/M_{32}

T1: Number of cells whose estimated result is 1， Mi: Number of cells with estimated results

Experiment

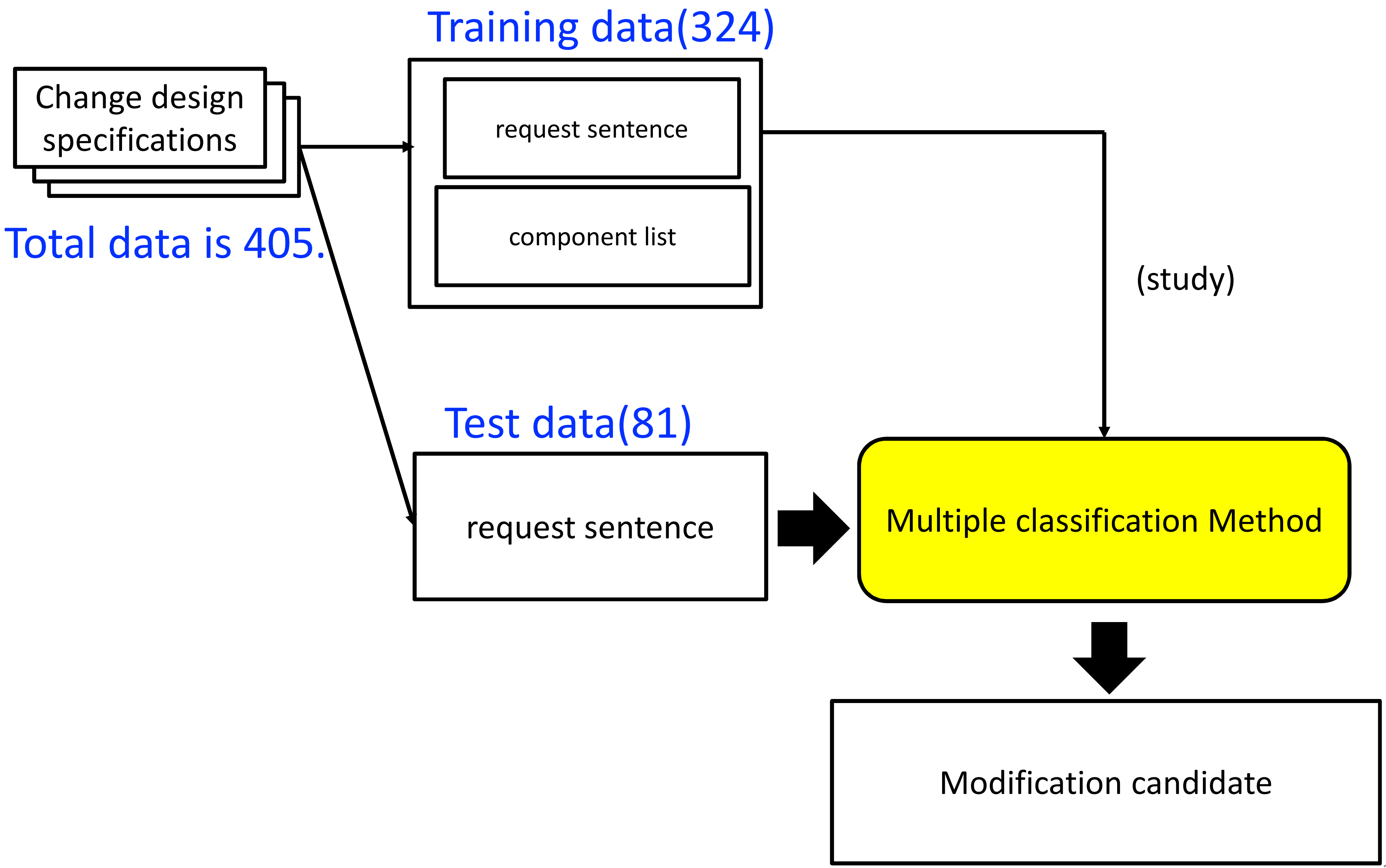
Purpose of experiment

To investigate whether the LP and RAkEL methods, which model the co-occurrence relationship, improve accuracy or not.

The four methods were evaluated using the same field data.

Multi-label classification method	Classifier
M1:Neural Network(NN)	
M2:BR method	SVM
M3:LP method	SVM
M4:RAkEL method	SVM

Data used in the experiments



Results of the experiment

Software Engineering Lab

The threshold was set so that the candidate range ratio is around 30 percent.

Method	Candidate Range ratio	Accuracy in the candidate range	Missing rate
M1:NN	30.00%(0.06)	18.00%	23.00%
M2:BR+SVM	29.10%(0.06)	19.10%	17.10%
M3:LP+SVM	29.70%(0.06)	22.20%	23.00%
M4:RAkEL+SVM	29.50%(0.07)	24.50%	15.60%

Improved 5.9% → ①

M3 is 5.9% worse than M2→ ②

Improved 1.5% → ③

Result

① M2 is more accurate than M1→ SVM is an excellent classifier

② M3 is less accurate than M2 → Small number of data could have caused overlearning.

③ M4 is the most accurate one.

RAkEL provides the best results, meaning to model the co-occurrence relationship has a good effect to reduce missing rate. However their missing rates are not at enough level for

Summary and Future Issues

Software Engineering Lab

Summary

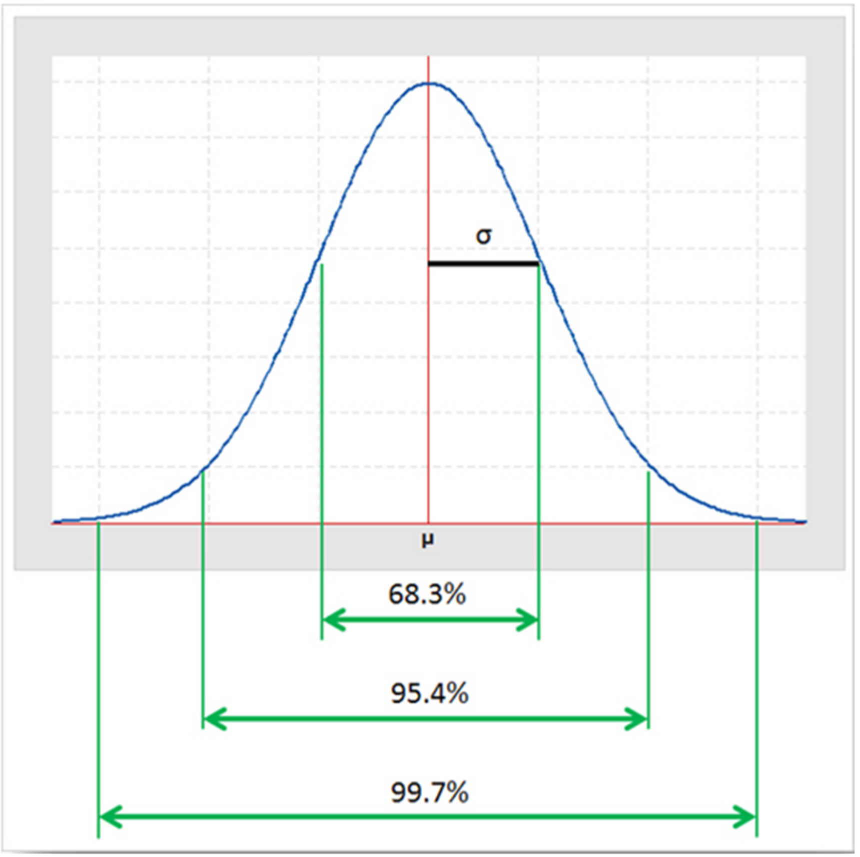
- We proposed an impact analysis method that learn change histories to directly create modification candidates.
- To improve the previous study, which use NN as the machine-learning component, we proposed a multi-label classification method considering the co-occurrence relationship
- The effectiveness of this method was confirmed by an experiment using BR, LP, and RAkEL methods.

Future Issues

- Application of an improved algorithm for the RAkEL method
- Validation by using the other data set (from OSS)

Supplementary data: Reasons for determining target values

Utilizes standard deviation (σ), a value often used in quality control

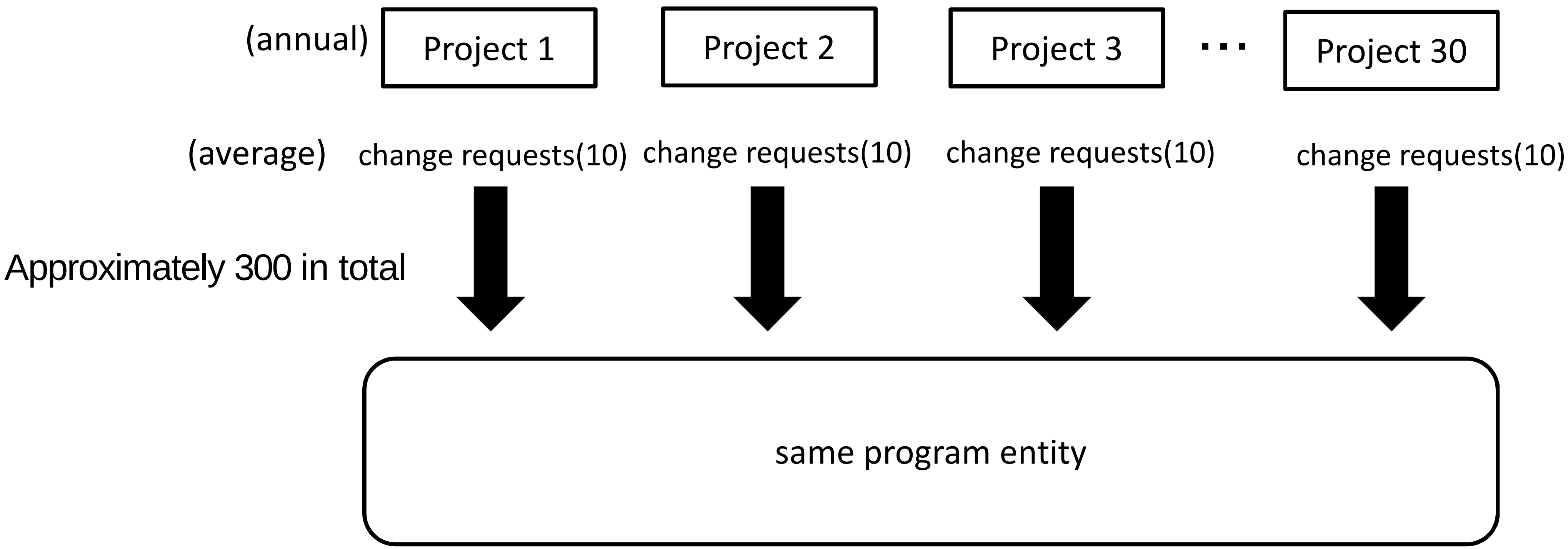


z-distribution diagram

- ±σ (σ interval): 68.3%
- ±2σ (2σ interval): 95.4%
- ±3σ (3σ interval): 99.7%

Supplementary material: Target projects used for the study

中島研
Software Engineering Lab



- Each project modifies the program matrix for multiple change requests
- Create a change design document for each change request

Improved machine learning implementation methods.

Software Engineering Lab

Apply and evaluate machine learning methods that consider co-occurrence relationships to reduce the hazard rate that has been the subject of previous research.

research goal

Performance targets, taking into account the extent to which this is possible in terms of actual audits：
Candidate Range ratio ≦ 30% and, Missing rate ≦5%