Enhancing Traceability Link Recovery with Unlabeled Data

Jianfei Zhu∗, Guanping Xiao∗†⋆, Zheng Zheng‡, Yulei Sui §

∗College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

†State Key Laboratory of Novel Software Technology, Nanjing University, China

‡School of Automation Science and Electrical Engineering, Beihang University, China

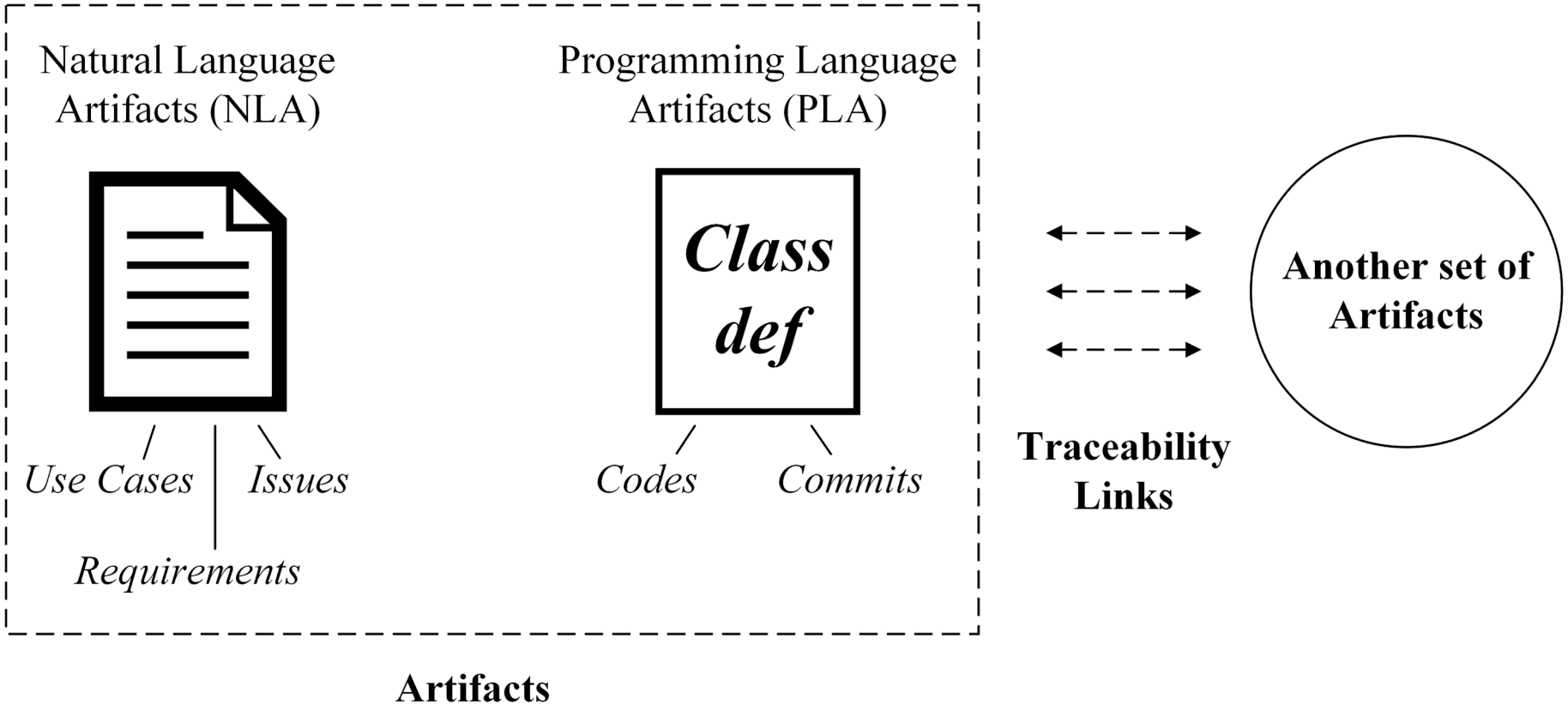
§ School of Computer Science, University of Technology Sydney, Australia

Charlotte, North Carolina, USA (Online)

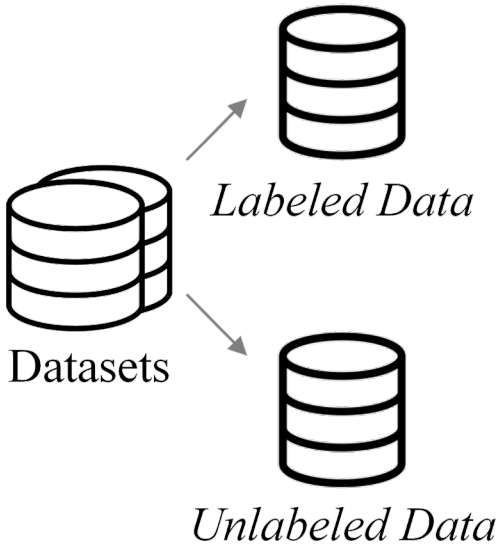
2022-11-01

1. Traceability Link Recovery

Traceability link recovery (TLR) is a software engineering task that recovers links between different types of software artifacts.



1. Traceability Link Recovery (Cont.)



Training

Evaluating

high performance

TLR methods based on classification models,
i.e. machine learning or deep learning

difficult to obtain and small quantity

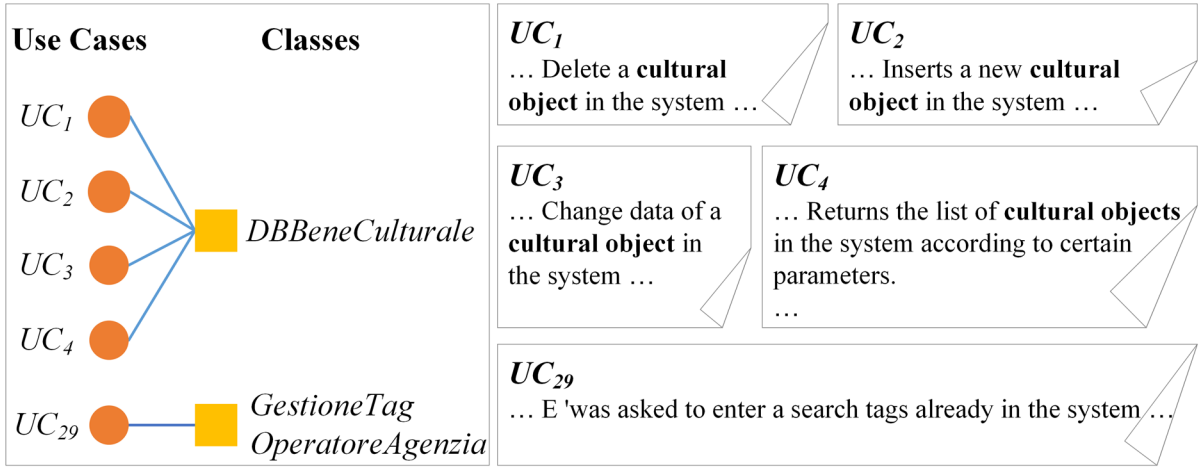
Labeled Data

readily available and large quantity

Unlabeled Data

2. Motivating Example

This is a real-world traceability link example in the eTOUR project. There is a link between use case 1 to 4 and the class DBBeneCulturale and there is no link between use case 29 and the class DBBeneCulturale.

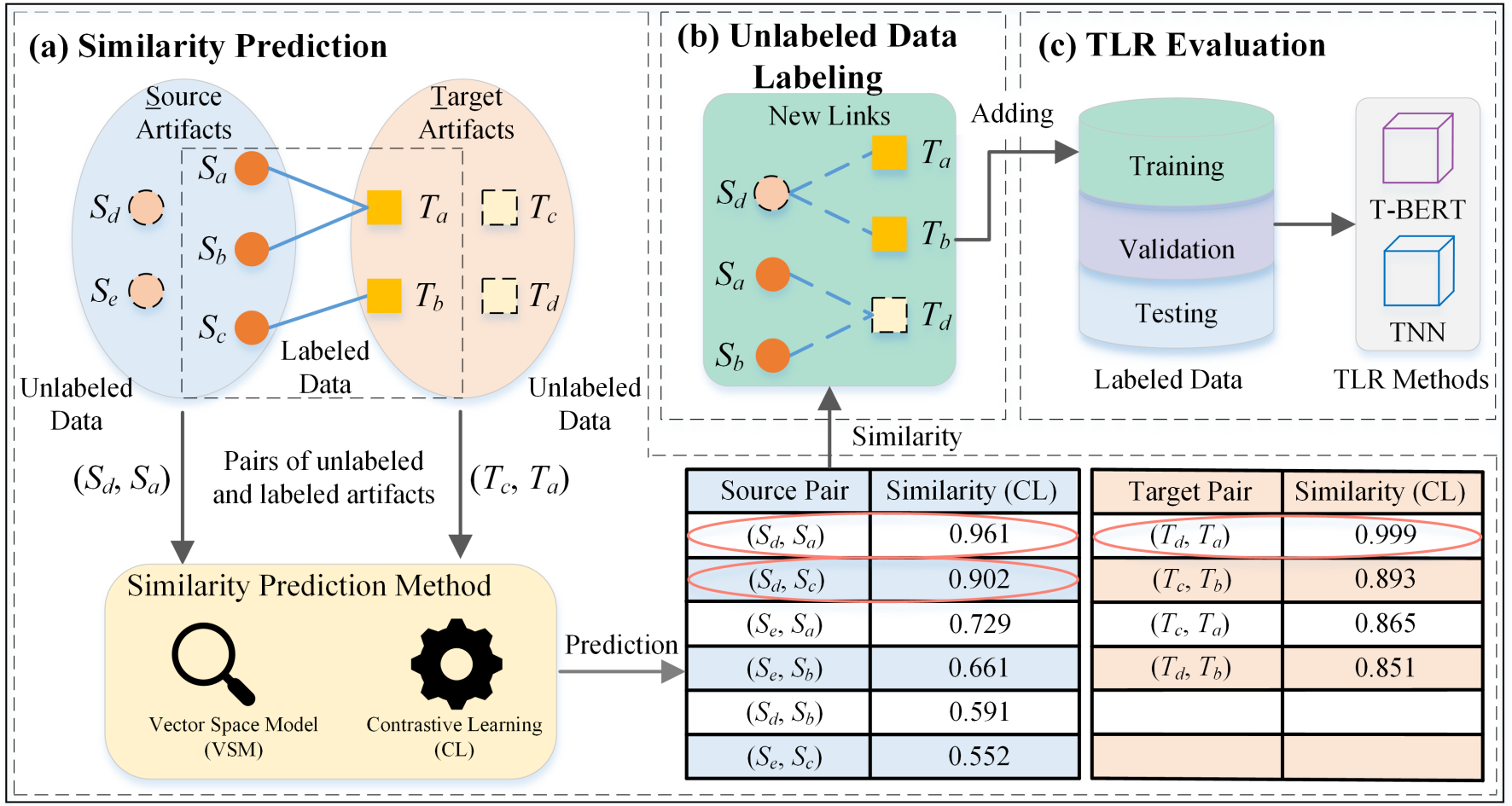


The following table is the result of calculating text similarity between use cases by VSM. High similarity between use cases linked to the same target class.

Has Common Target Artifact		No Common Target Artifact	
Use Case	**Similarity**	**Use Case**	**Similarity**
(UC_1, UC_3)	0.646	(UC_4, UC_{29})	0.162
(UC_2, UC_3)	0.556	(UC_3, UC_{29})	0.142
(UC_1, UC_2)	0.419	(UC_1, UC_{29})	0.138
(UC_3, UC_4)	0.347	(UC_2, UC_{29})	0.123
(UC_2, UC_4)	0.328		
(UC_1, UC_4)	0.257		

3. Our Solution

The TraceFUN framework mainly consists of three parts: similarity prediction, unlabeled data labeling and TLR evaluation.

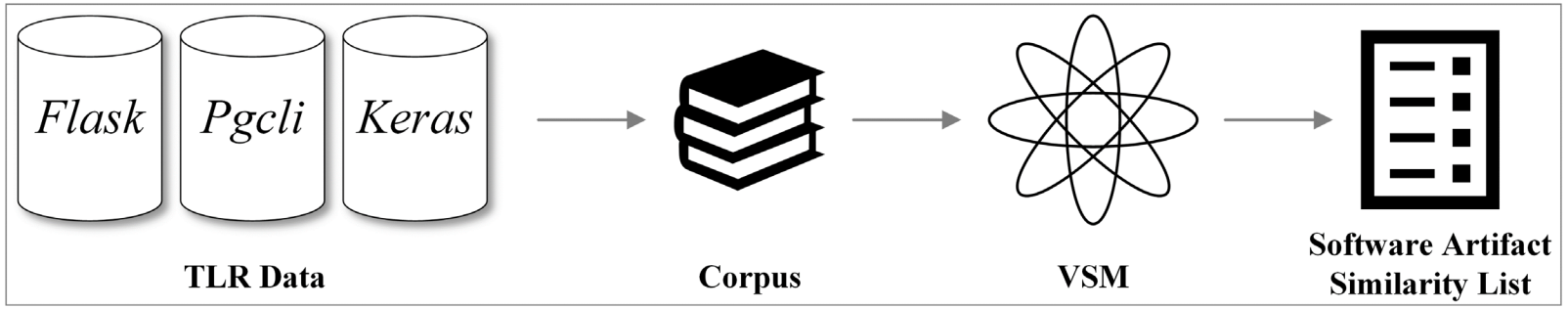


Overview of TraceFUN

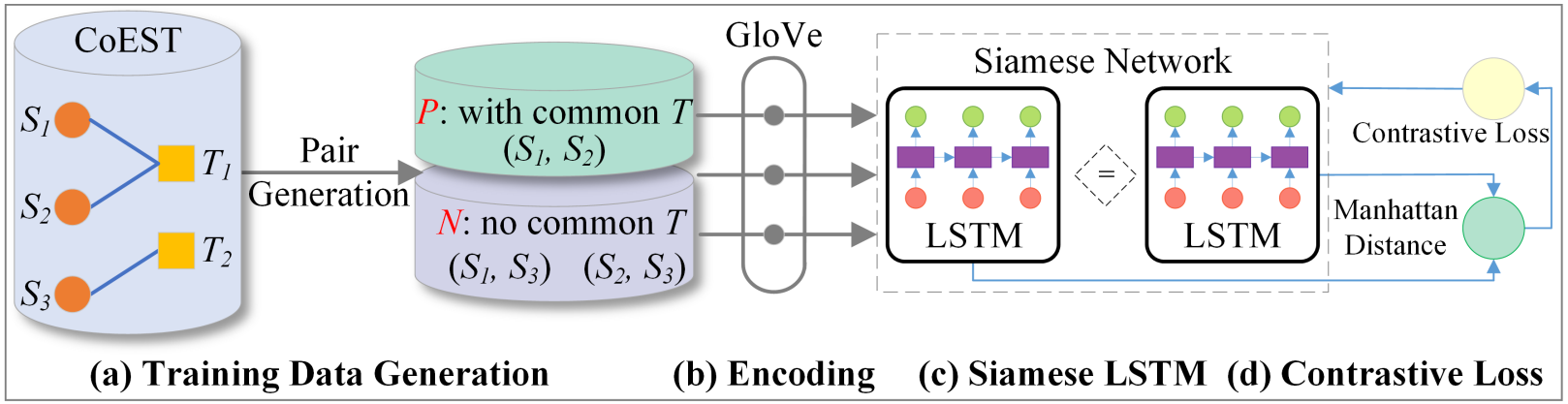
4. Our TraceFUN Approach

(1) Similarity Prediction Method

Vector Space Model (VSM)



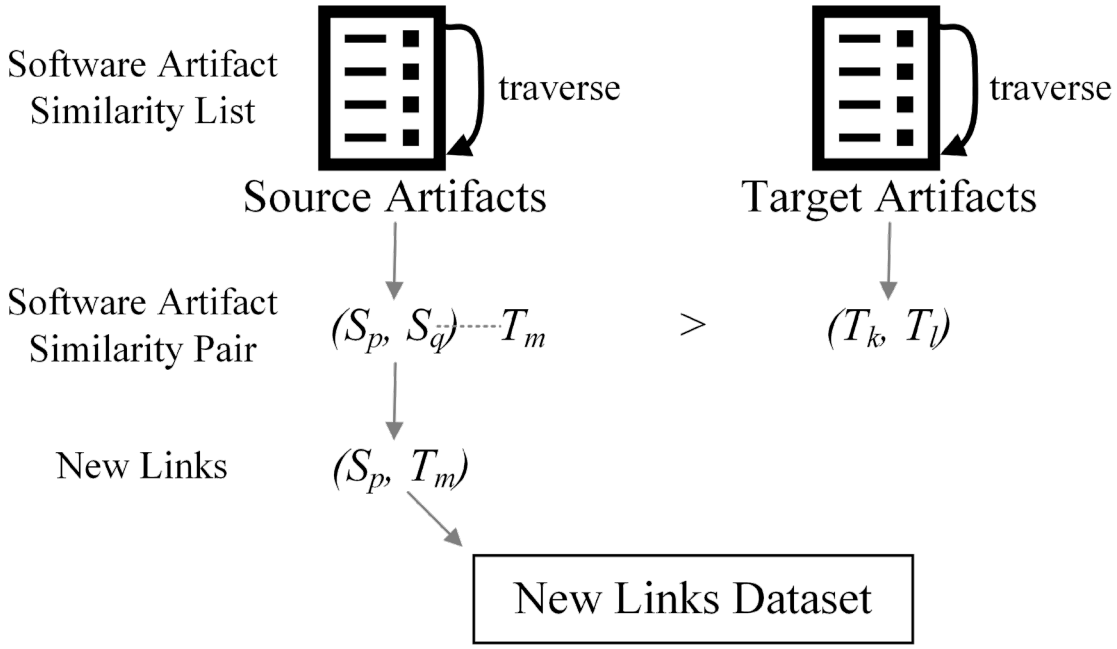
Contrastive Learning (CL)

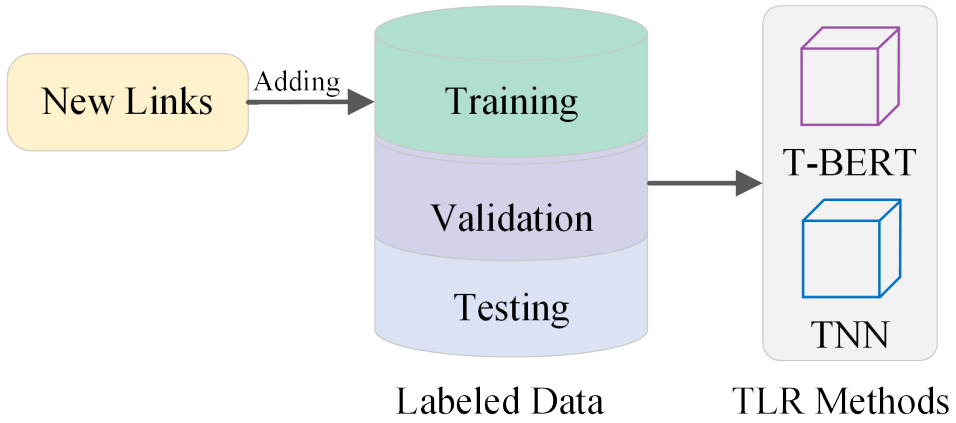


4. Our TraceFUN Approach (Cont.)

(2) Unlabeled Data Labeling

(3) TLR Evaluation





5. EXPERIMENT SETUP

- **Dataset for CL Training:** 15 datasets provided by CoEST [46].
- **Dataset for TraceFUN Evaluation:** issue-commit links are collected from [2], includes Flask, Pgcli, and Keras.
- **Settings for TLR Methods:**
  - TNN [26], ICSE 2017
  - T-BERT [27], ICSE 2021
- **Evaluation Metrics:**
  - F1-Score
  - F2-Score
  - Mean Average Precision (MAP)
- **Experiment Environments:**
  - Python 3.8, TensorFlow 2.6.0 and Keras
  - Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz
  - NVIDIA GeForce RTX 2080Ti GPU

6. Research Questions

RQ1: Can TraceFUN improve TLR performance?

RQ2: What's the impact of different similarity prediction methods used in TraceFUN on TLR performance?

RQ3: What's the impact of different sizes of newly labeled links generated by TraceFUN on TLR performance?

7. Results-RQ1

RQ1: Can TraceFUN improve TLR performance?

F1-Score of T-BERT with 50% new data trained by TraceFUN

Project	Original	VSM	CL	Random
Falsk	0.644	0.718 (12%)	0.713 (11%)	0.576 (-11%)
Pgcli	0.737	0.765 (4%)	0.769 (4%)	0.638 (-13%)
Keras	0.945	0.910 (-4%)	0.920 (-3%)	0.882 (-7%)

F1-Score of TNN with 50% new data trained by TraceFUN

Project	Original	VSM	CL	Random
Falsk	0.033	0.149 (351%)	0.164 (396%)	0.022 (-35%)
Pgcli	0.036	0.148 (311%)	0.181 (403%)	0.039 (8%)
Keras	0.032	0.066 (106%)	0.042 (33%)	0.028 (-13%)

Answer: TraceFUN can significantly improve TLR performance. TraceFUN is able to capture the semantically similar relationships between unlabeled and labeled artifacts, thereby generating effective newly labeled links for TLR model training.

Note that the displayed results are partially selected from our paper.
For full results, please refer to our paper.

7. Results-RQ2

RQ2: What's the impact of different similarity prediction methods used in TraceFUN on TLR performance?

F1-Score of T-BERT trained by TraceFUN

Project	Original	5%		20%		50%		80%		110%	
		VSM	CL	VSM	CL	VSM	CL	VSM	CL	VSM	CL
Flask	0.644	0.667	0.654	0.682	0.671	0.718	0.713	0.759	0.757	0.778	0.763
Pgcli	0.737	0.737	0.734	0.767	0.757	0.765	0.769	0.810	0.714	0.817	0.697
Keras	0.945	0.939	0.940	0.936	0.938	0.910	0.920	0.895	0.910	0.866	0.889

F1-Score of TNN trained by TraceFUN

Project	Original	5%		20%		50%		80%		110%	
		VSM	CL	VSM	CL	VSM	CL	VSM	CL	VSM	CL
Flask	0.033	0.031	0.035	0.063	0.064	0.149	0.164	0.210	0.331	0.306	0.392
Pgcli	0.036	0.047	0.044	0.082	0.074	0.148	0.181	0.233	0.182	0.295	0.203
Keras	0.032	0.041	0.038	0.052	0.038	0.066	0.042	0.061	0.033	0.067	0.041

Note that the displayed results are partially selected from our paper.
For full results, please refer to our paper.

7. Results-RQ2 (Cont.)

RQ2: What's the impact of different similarity prediction methods used in TraceFUN on TLR performance?

Answer: The performance improvements by VSM and CL used in TraceFUN are different regarding different TLR methods and datasets. It is necessary to select a suitable similarity prediction method according to specific TLR methods and datasets to improve the performance better. For the TLR task on issues and commits, users are suggested to use VSM and CL.

7. Results-RQ3

RQ3: What's the impact of different sizes of newly labeled links generated by TraceFUN on TLR performance?

F1-Score of T-BERT trained by TraceFUN

Project	Original	5%		20%		50%		80%		110%	
		VSM	CL	VSM	CL	VSM	CL	VSM	CL	VSM	CL
Flask	0.644	0.667 4%	0.654 2%	0.682 6%	0.671 4%	0.718 12%	0.713 11%	0.759 18%	0.757 18%	0.778 21%	0.763 19%
Pgcli	0.737	0.737 0%	0.734 0%	0.767 4%	0.757 3%	0.765 4%	0.769 4%	0.810 10%	0.714 -3%	0.817 11%	0.697 -5%
Keras	0.945	0.939 -1%	0.940 -1%	0.936 -1%	0.938 -1%	0.910 -4%	0.920 -3%	0.895 -5%	0.910 -4%	0.866 -8%	0.889 -6%

Note that the displayed results are partially selected from our paper.
For full results, please refer to our paper.

7. Results-RQ3 (Cont.)

RQ3: What's the impact of different sizes of newly labeled links generated by TraceFUN on TLR performance?

F1-Score of TNN trained by TraceFUN

Project	Original	5%		20%		50%		80%		110%	
		VSM	CL	VSM	CL	VSM	CL	VSM	CL	VSM	CL
Flask	0.033	0.031 -6%	0.035 7%	0.063 91%	0.064 95%	0.149 351%	0.164 396%	0.210 537%	0.331 902%	0.306 828%	0.392 1088%
Pgcli	0.036	0.047 30%	0.044 23%	0.082 127%	0.074 107%	0.148 311%	0.181 403%	0.233 547%	0.182 407%	0.295 719%	0.203 463%
Keras	0.032	0.041 27%	0.038 18%	0.052 63%	0.038 19%	0.066 106%	0.042 33%	0.061 92%	0.033 4%	0.067 109%	0.041 28%

Note that the displayed results are partially selected from our paper.
For full results, please refer to our paper.

7. Results-RQ3 (Cont.)

RQ3: What's the impact of different sizes of newly labeled links generated by TraceFUN on TLR performance?

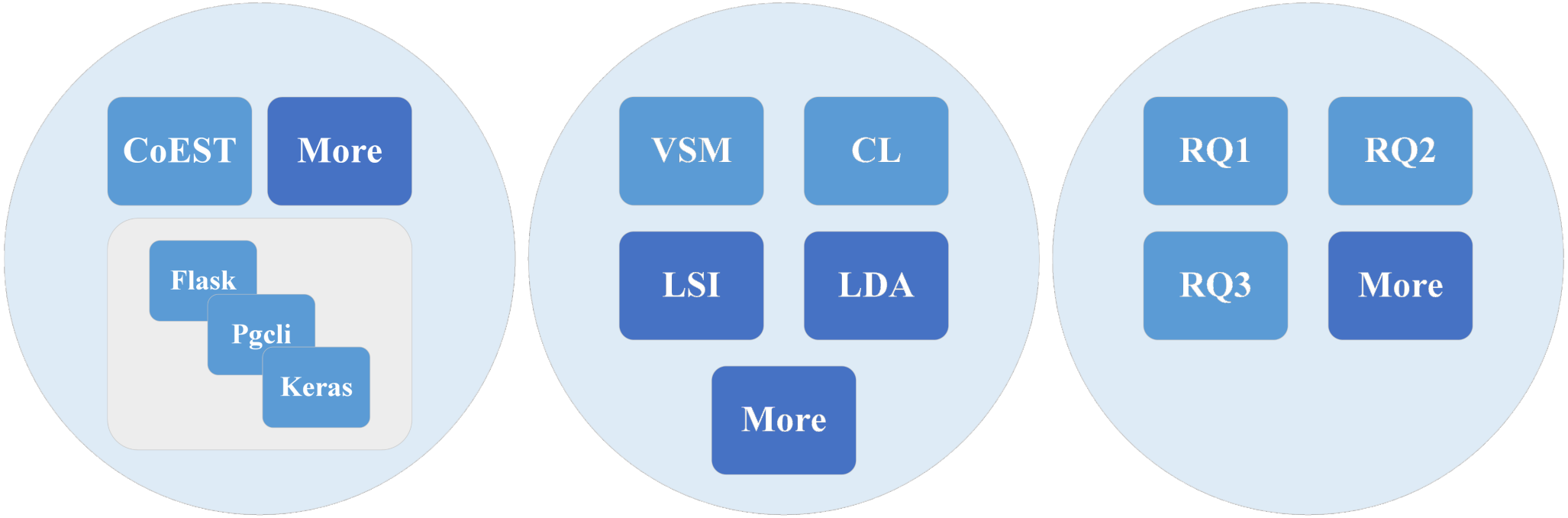
Answer: Generally, TLR performance can be improved by adding more labeled links via TraceFUN. However, for different TLR methods and datasets, the size of newly labeled links greatly impacts the performance. Therefore, it is necessary to fine-tune the size of new links labeled by TraceFUN to obtain a better result according to specific TLR methods and datasets.

8. Contribution

Key Contributions:

- To the best of our knowledge, this paper presents the first attempt to use unlabeled data for TLR.
- TraceFUN, for the first time, introduces VSM and CL methods to measure the similarity between unlabeled and labeled artifacts for generating new training samples.
- We have evaluated TraceFUN by comparing it with two state-of-the-art methods using 5-fold cross-validation on three GitHub projects. Results show that TraceFUN boosts T-BERT and TNN with a maximum improvement of F1-score up to 21% and 1,088%, respectively.
- We made the source code of TraceFUN publicly available at https://github.com/TraceFUN.

9. Future Work



Datasets

Similarity Prediction Methods

Research Questions

9. Future Work (Cont.)

Visualization Tool

TraceFun Tool

- Home Page
- Similarity Prediction
- Data Label
- TLR Evaluation
- Results

Similarity Prediction

Upload

Vector Space Model | Latent Semantic Index | Latent Dirichlet Allocation | Contrastive Learning

ID	Name	Prediction	Download	Status
1	run.py	Re-Predict	Download	Predicted
2	aaa.py	Start	Download	Unpredicted

Similarity Prediction Task Fresh

Similarity Prediction Method:
Dataset ID: 0
Dataset Name:
Task Status: Unpredicted

Source Artifact: 0%
Target Artifact: 0%

9. Future Work (Cont.)

Visualization Tool

TraceFun Tool

Home Page
Similarity Prediction
Data Label
TLR Evaluation
Results

Data Label

Label

#	ID	Name
☑	1	run.py
☐	2	aaa.py

Data Label Option

Dataset ID: 1
Dataset Name: run.py
Original Links Quantity: 0

Similarity Link List Selection
请选择

New Links (Proportion)
5 20 50 80 110

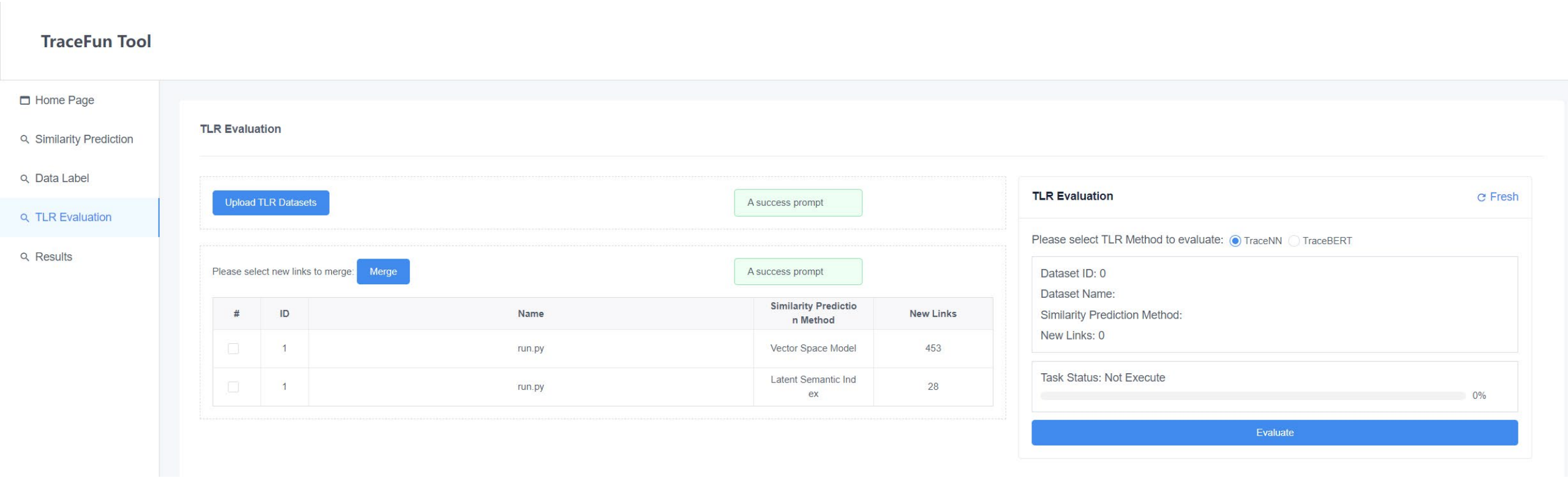
New Links (Quantity)
Custom Quantity 0

取消 Label

arity Prediction Method	New Links	Status	Download
Vector Space Model	453	Not Execute	下载
atent Semantic Index	28	Finish	下载

9. Future Work (Cont.)

Visualization Tool



Thank you for your listening!

https://github.com/TraceFUN