

Cross-Domain Deep Code Search with Meta Learning

- Yitian Chai¹, Hongyu Zhang², Beijun Shen¹, Xiaodong Gu¹

¹School of Software, Shanghai Jiao Tong University

²The University of Newcastle



Deep Learning Based Code Search



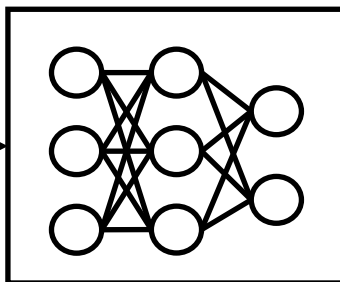
Query

🔍 how to sort arrays?

Code

```
public void sort(int[] para) {  
    int ret = Array.sort(para);  
}
```

Deep Neural Networks



Code Corpus

Training

Code Corpus

stackoverflow

GitHub

Google Code Search LABS



When Deep Learning Met Uncommon Languages



Main Challenges:

- Domain-Specific Languages
- New Languages
- Private Codebase

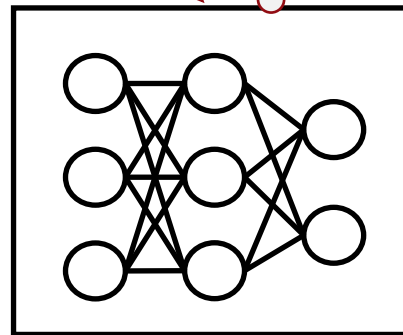


How to sort an array in **Solidity**?

What is Solidity?

Too few people use it, I don't know

I'm well trained with a **large amount** of data. I can search code for **Python, C# and Java**...





Our Idea



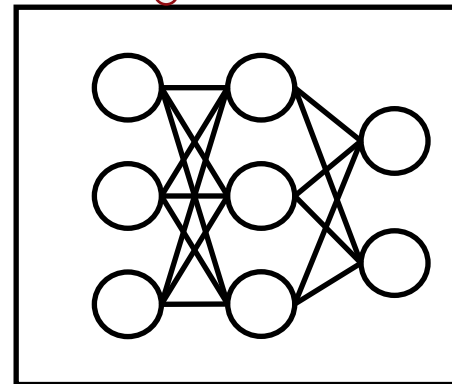
▪ Towards Few-Shot Learning

I can provide you with **a few** examples.



Really?! Just a few samples? I need **large** data.

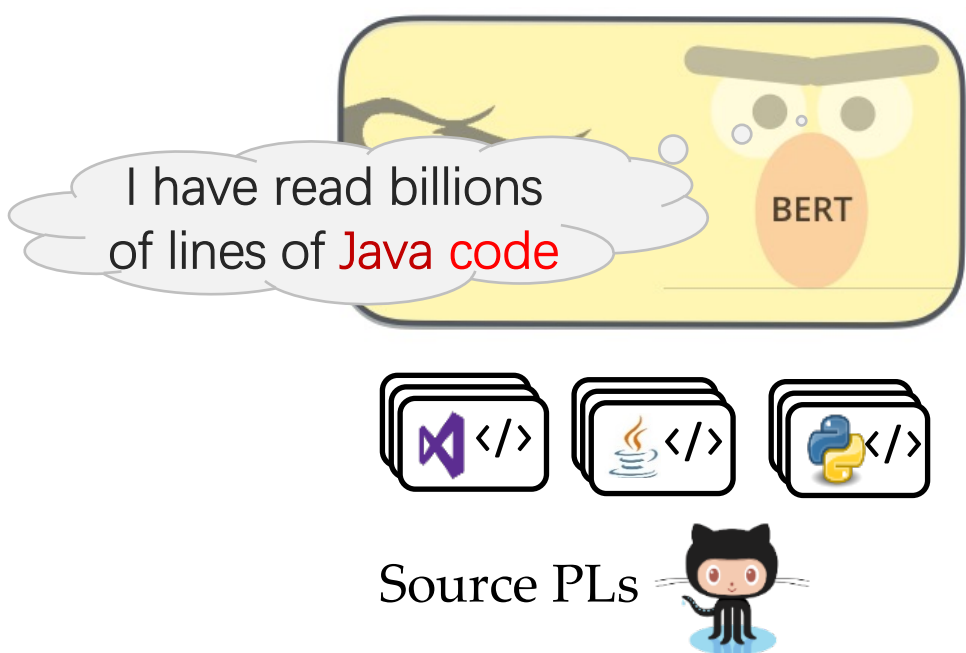
Aha! I have already been trained how to sort an array in Java, Python, etc



Pre-Training & Fine-Tuning ?

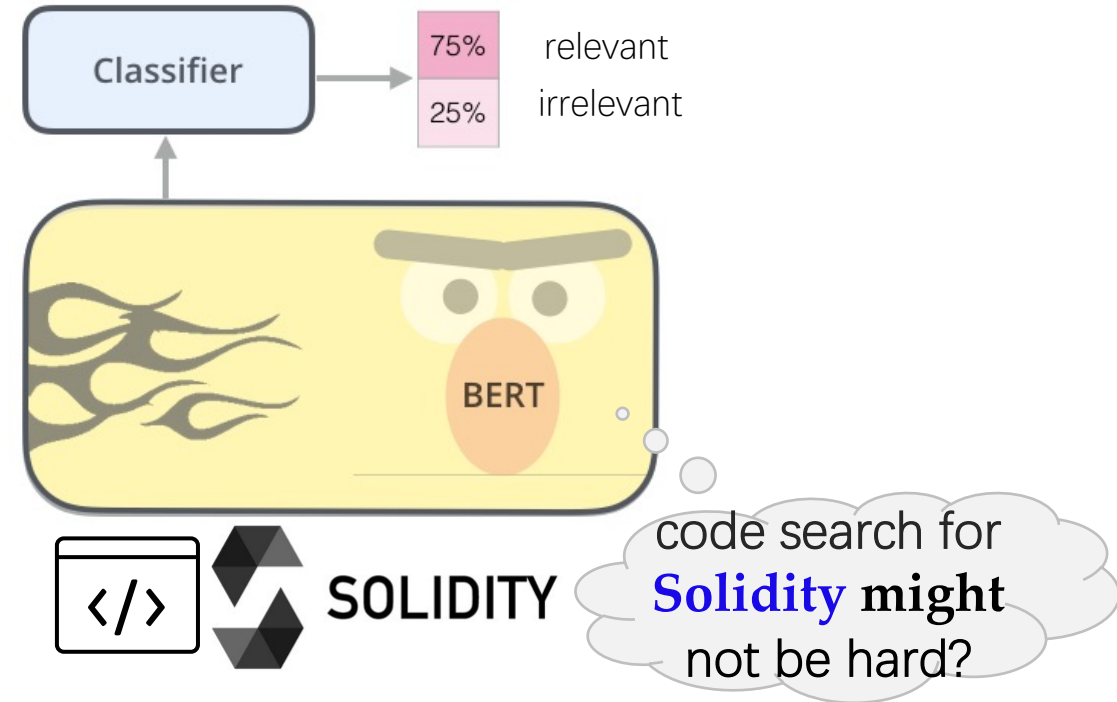


Unsupervised pre-training on **large-scale common** programming language



Phase1: Pre-Training

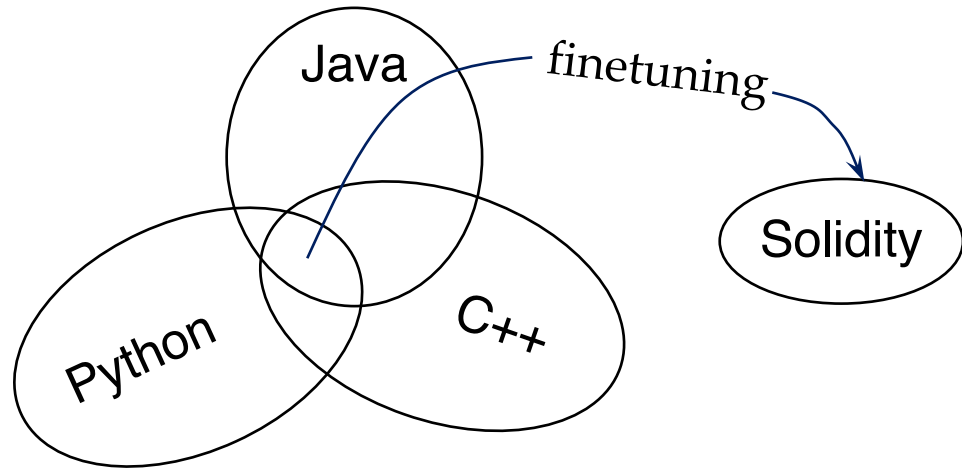
Fine-tuning on **small-scale domain-specific** programming language



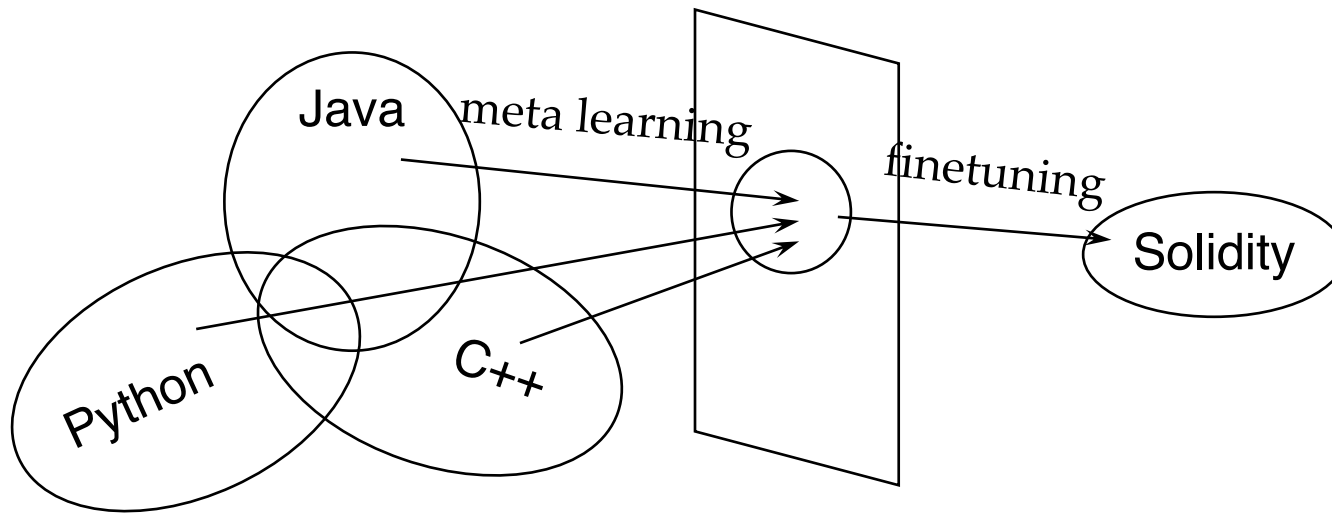
Phase2 : Fine-Tuning



Problems with Pre-Trained Language Models



Pretraining & Finetuning



Meta Learning

Problems:

- Conflict Representations
- Distracted Parameters

Our solution:

Learning a good initialization of various models using meta-learning.



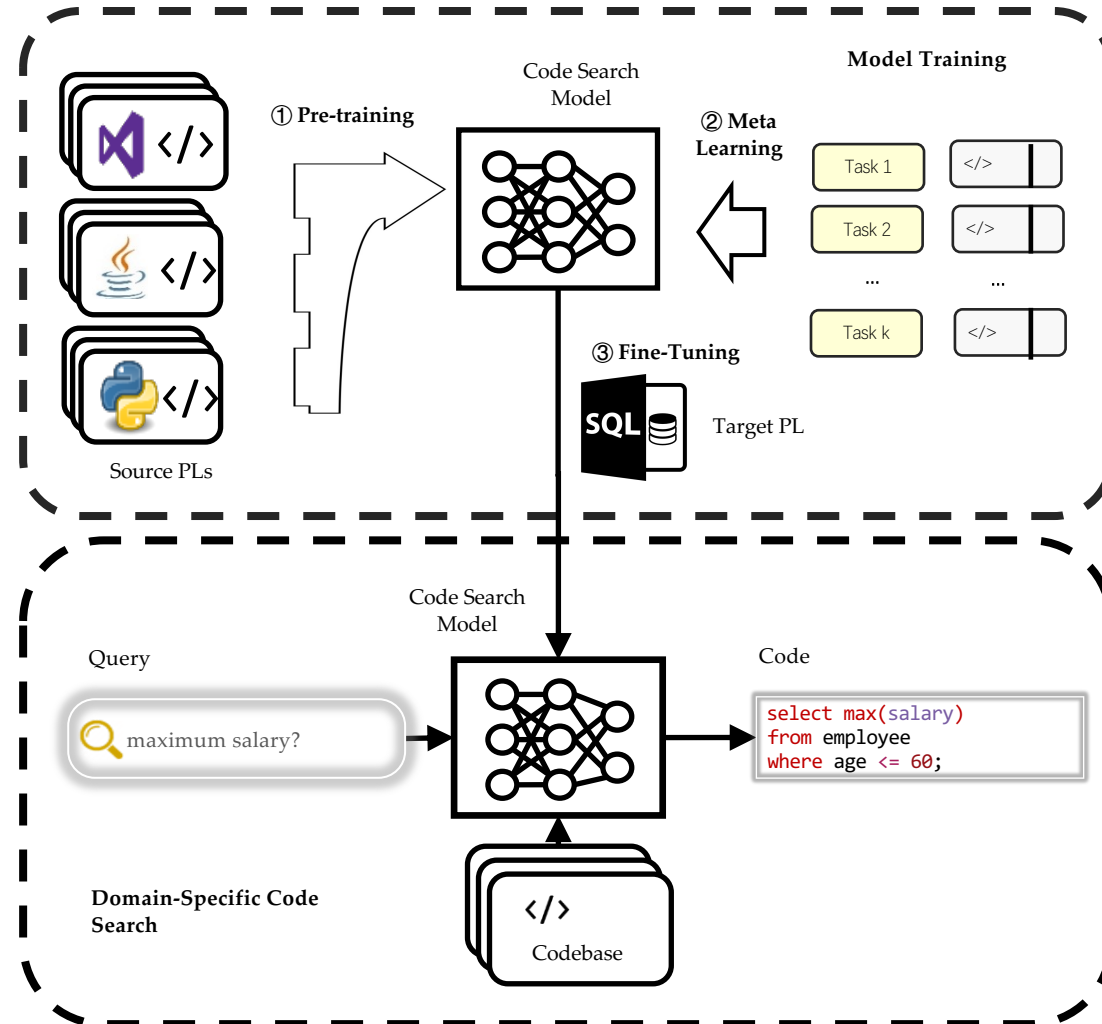
Cross-Domain Deep Code Search with Meta Learning

1. Pre-Training

2. Meta-Learning

3. Fine-Tuning

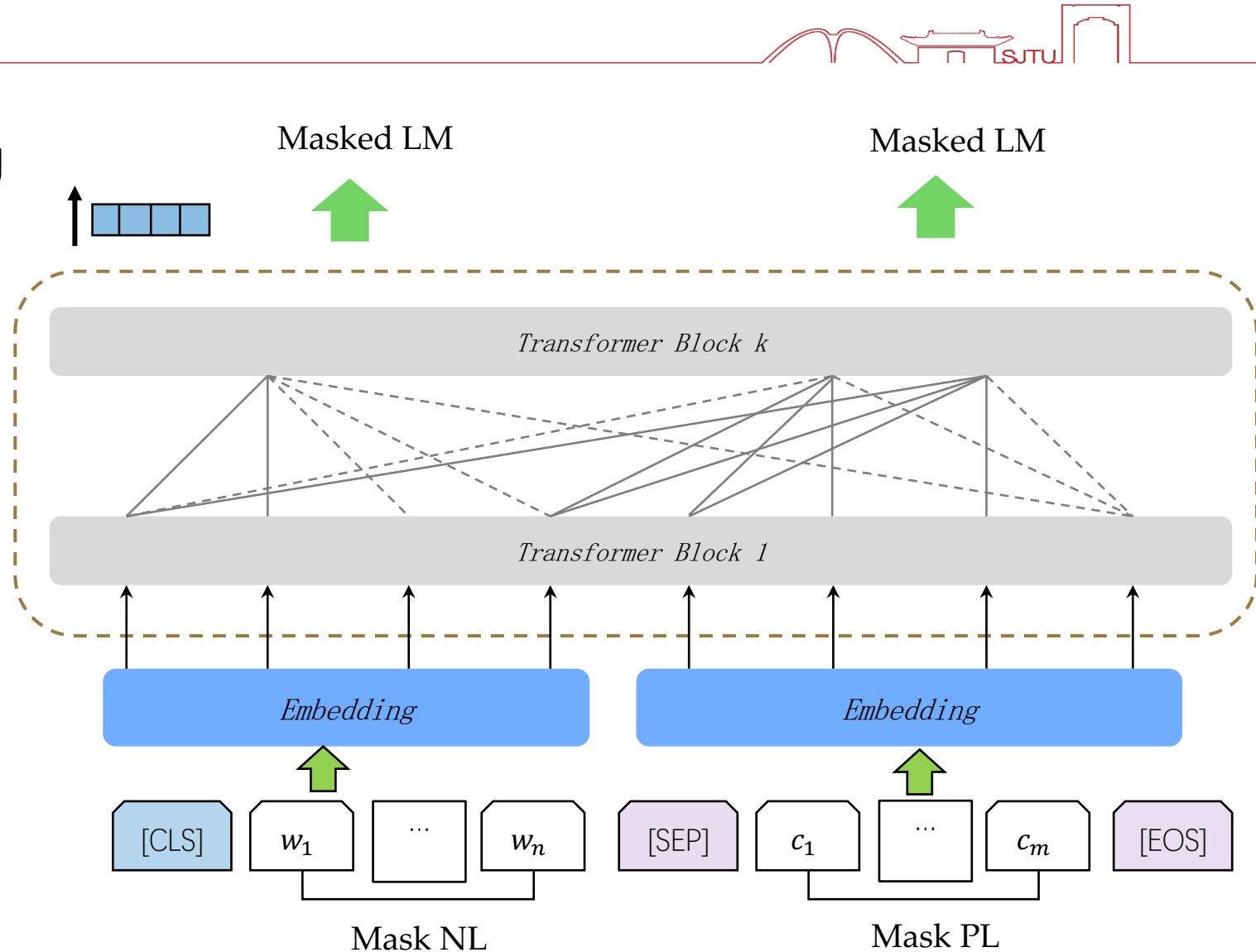
4. Code Search





Step 1. Pre-Training

- Masked Language Modeling (MLM)
 - Input: natural language text+code snippet
 - Task: Randomly mask 15% tokens, and let the model predict the original tokens





Step 2. Meta-Learning and Few-Shot Learning

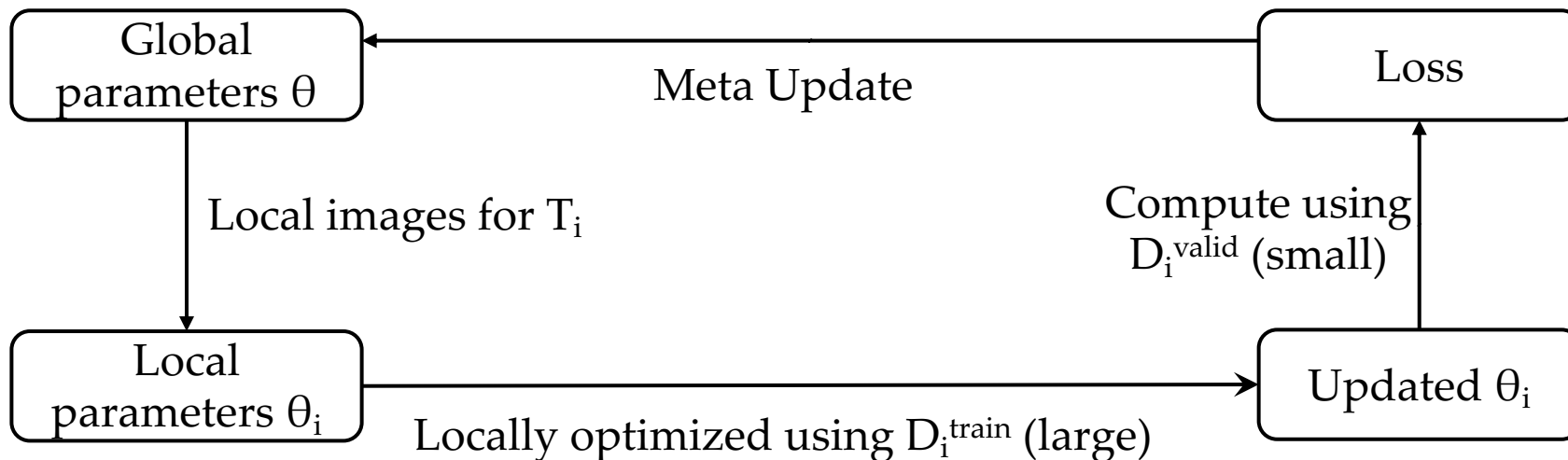


Meta Learning

Also known as “Learning to learn”, a series of learning tasks are constructed and a meta learner collects the information of each learning task and adjusts the learning strategy according to the learning situation of each task.

Few-Shot Meta Learning

A series of few-shot learning tasks are constructed. A meta learner collects the information of each few-shot task and adjusts the learning strategy according to the learning situation of each few-shot task.

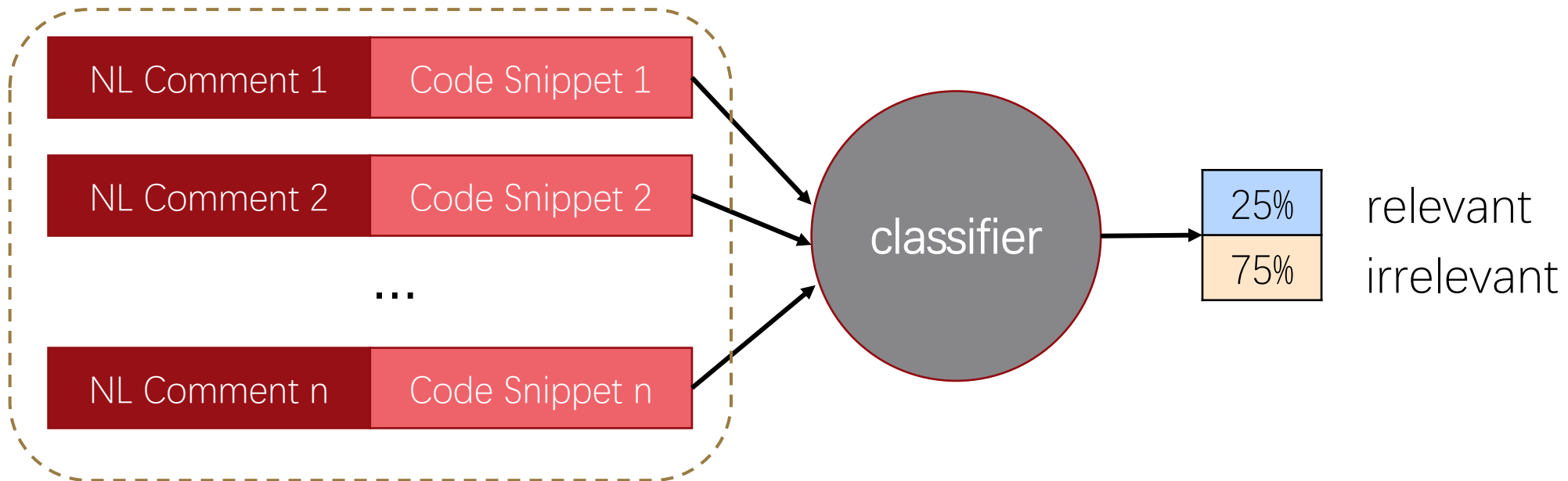




Step 3. Fine-Tuning on the Target Language

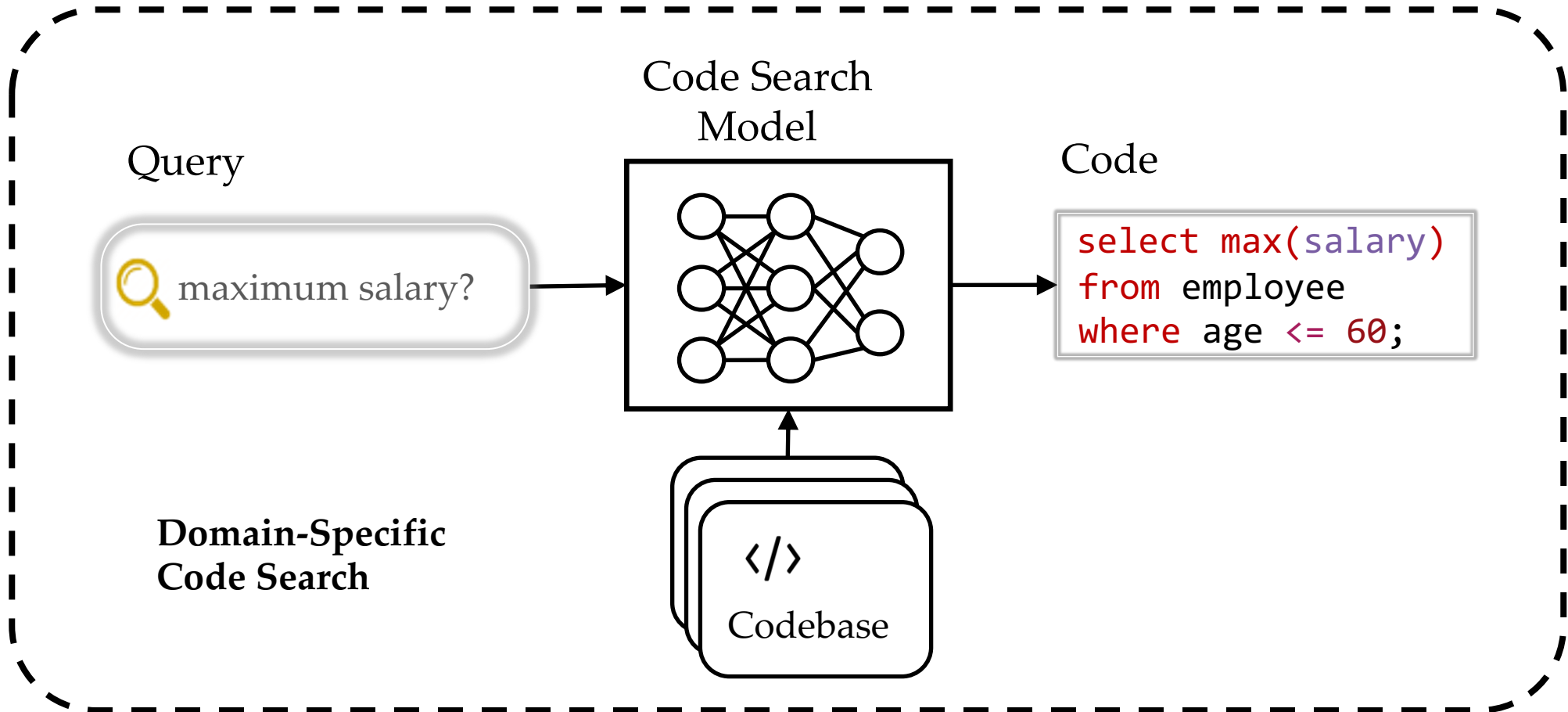


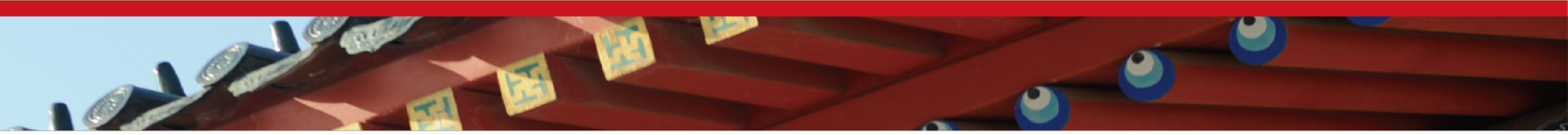
- Code Search → Binary Classification





Step 4. Code Search on the Target Language





Research Questions



1. How effective is CDCS in cross-domain code search?
2. What is the impact of data size on the performance of cross-domain code search?
3. How effective is CDCS applied to other pre-trained programming language models?
4. How do different hyperparameters affect the performance of CDCS?



Datasets



- Datasets for pre-training and meta-learning

Phase		Python	Java
pre-training	# functions	412,178	454,451
	# comments	412,178	454,451
meta learning	# functions	824,342	908,886
	# comments	824,342	908,886

- Datasets for fine-tuning

Language	Train(Fine-tune)	Valid	Test
Solidity	56,976	4,096	1,000
SQL	14,000	2,068	1,000



Metrics



- **MRR**

- $$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{\text{Rank}(i)}$$

- **Top-k accuracy**

- measures how many answers in the first k results hit the query.



Comparison Methods



1. Code Search with Pre-training
2. Code Search based on pre-trained model with Natural Language
3. Within-domain Code Search with CodeBERT
4. Cross-Language Code Search with CodeBERT



Experimental Results



- Effectiveness in Cross-Domain Deep Code Search (RQ1)

Language	Model	Acc@1	Acc@5	Acc@10	MRR
SQL	No-Pretraining	0.002	0.010	0.022	0.0124
	CodeBERT(NL-based)	0.652	0.926	0.966	0.7690
	CodeBERT(within-domain)	0.607	0.899	0.945	0.7351
	CodeBERT(cross-language)	0.675	0.920	0.960	0.7818
	CDCS	0.746	0.952	0.972	0.8366
Solidity	No-Pretraining	0.002	0.008	0.014	0.0101
	CodeBERT(NL-based)	0.453	0.732	0.821	0.5801
	CodeBERT(within-domain)	0.515	0.798	0.857	0.6383
	CodeBERT(cross-language)	0.532	0.779	0.848	0.6436
	CDCS	0.658	0.829	0.879	0.7336

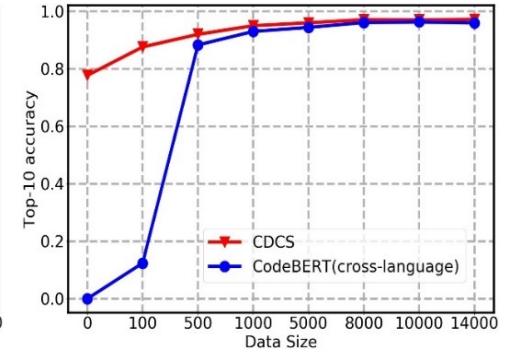
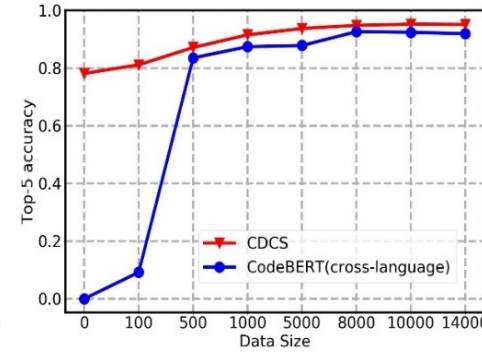
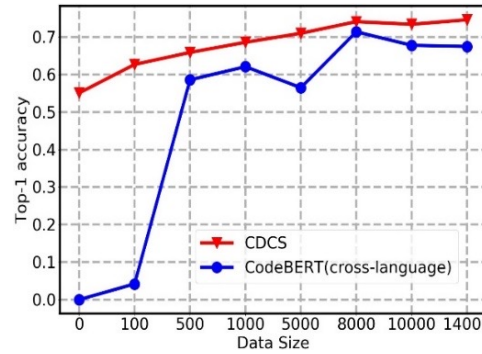
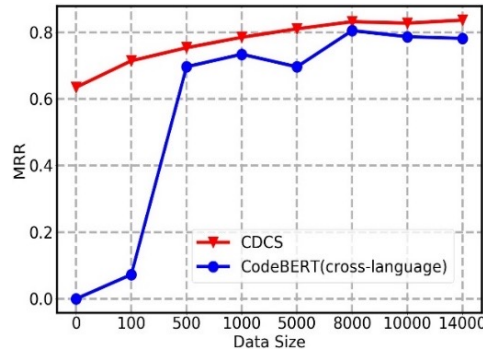


Experimental Results

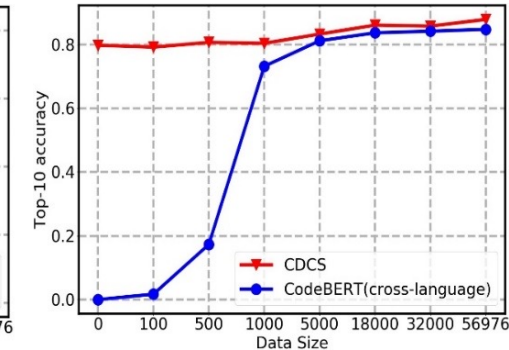
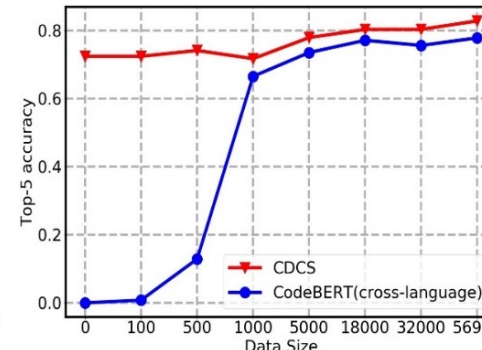
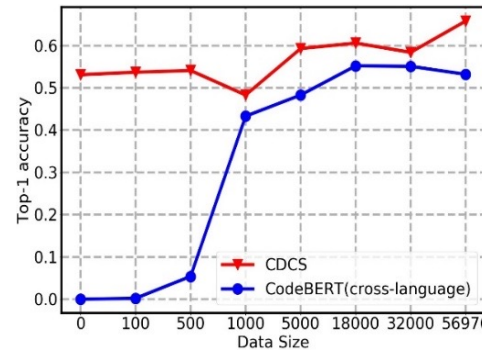
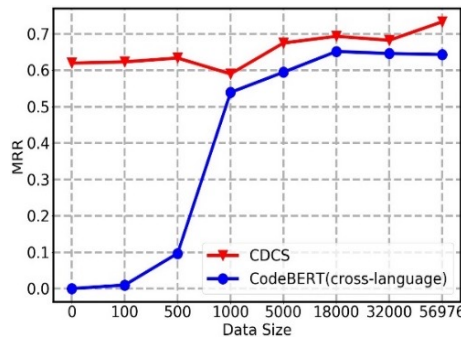


■ Effect of Data Size (RQ2)

SQL



Solidity





Experimental Results



- Performance on other Pre-trained Models(RQ3)

Language	Model	Acc@1	Acc@5	Acc@10	MRR
SQL	No-Pretraining	0.002	0.010	0.022	0.0124
	GPT2(NL-based)	0.481	0.808	0.889	0.6204
	GPT2(within-domain)	0.470	0.785	0.877	0.6088
	GPT2(cross-language)	0.447	0.767	0.875	0.5899
	CDCS _{GPT-2}	0.511	0.823	0.905	0.6464
Solidity	No-Pretraining	0.002	0.008	0.014	0.0101
	GPT2(NL-based)	0.484	0.751	0.830	0.6079
	GPT2(within-domain)	0.487	0.772	0.848	0.6073
	GPT2(cross-language)	0.481	0.760	0.827	0.6057
	CDCS _{GPT-2}	0.561	0.781	0.846	0.6607

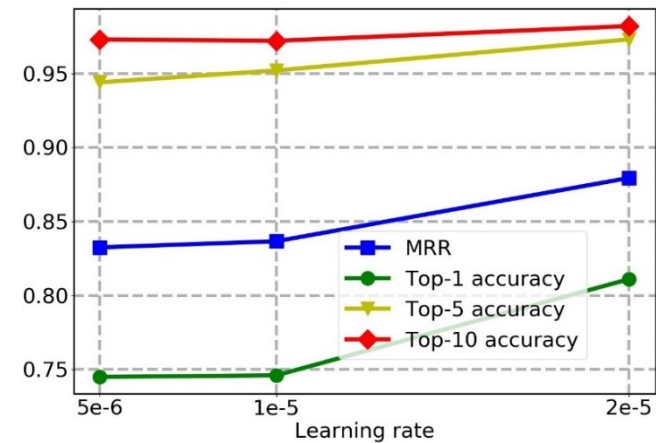
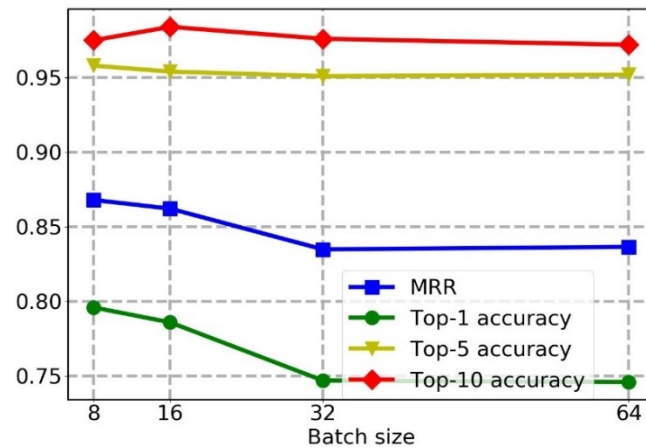


Experimental Results

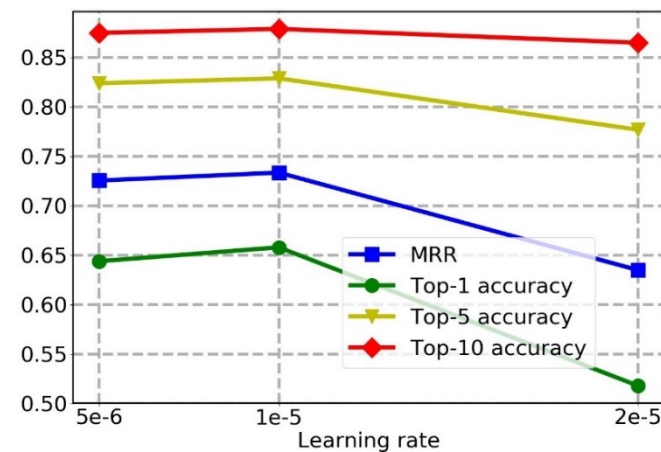
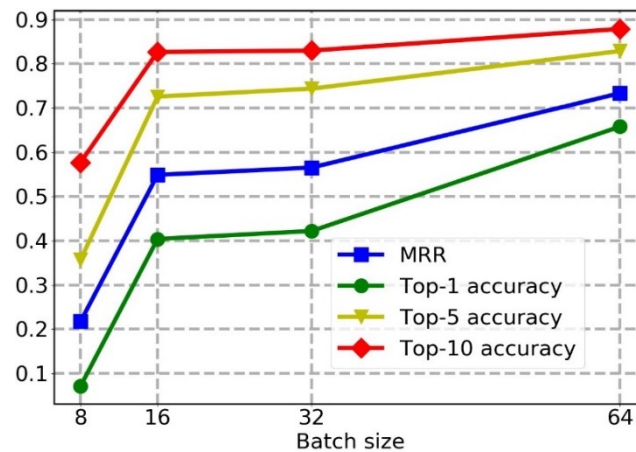


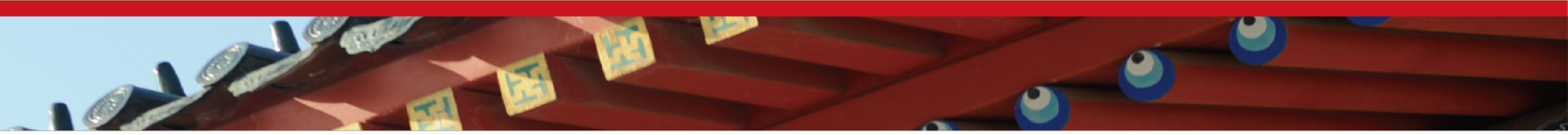
Impact of Different Hyperparameters (RQ4)

SQL



Solidity





Conclusion



CDCS - cross-domain code search approach.

- extends pre-trained models with meta learning
- achieves significant improvement in domain-specific code search.

Future Work

- more languages
- other software engineering tasks.



Thank You!

Q&A