Learning from Source Code

Prof. Dr. Zhi Jin

Key Laboratory of High Confidence Software Technologies (MoE),
Peking University, China
zhijin@pku.edu.cn

高可信软件技术
教育部重点实验室

30 Aug. 2020, KSEM2020, Hangzhou

北京大学

Agenda

- Can Machine Learn from Source Code, and Why
  - Be Inspired by Natural Language, Naturalness
- How Machine learns from Source Code
  - Source Code Comprehension
  - Capture Features of Source Code
- Learn What
  - Learn Task-Specific Knowledge and/or tacit knowledge
  - AiXCoder: a programming assistant
- What Next
  - Long Way to Go for Learning Knowledge what we Mean
  - Combine with Knowledge Graph

Human Exchange Knowledge using Natural Language

- Human beings communicate and exchange knowledge with each other
  - Teaching, reading, speaking, ……, sharing knowledge
- The system of communication and knowledge exchanging among human beings is natural language
  - which is an ordinary, instinctive part of everyday life

- Human being Gain most of the Knowledge via Learning from Natural Language

Naturalness of Natural Language led to Revolution of NLP

- Although natural languages have complex forms of expressive
  - Most human utterances are far simpler and much more repetitive and predictable

- That leads to the revolution in NLP with rich resources and advanced techniques
  - help to automatically extract knowledge from natural language documents

Techniques for the NLP

- 60s, Dictionary and grammar-based efforts
- 70s-80s, logic and formal semantics
  - That shows: dealing with NLP from the first principles is too cumbersome to perform practical tasks at scale
- 80s, corpus-based statistically rigorous method
- Later, lots of deep techniques …..

Natural Language Understanding - IBM Cloud

- Use advanced NLP to analyze text and extract meta-data from content such as concepts, entities, keywords, categories, sentiment, emotion, relations, and semantic roles.
- Apply custom annotation models developed using Watson Knowledge Studio to identify industry/domain specific entities and relations in unstructured text with Watson NLU.

Natural Language and Program Language

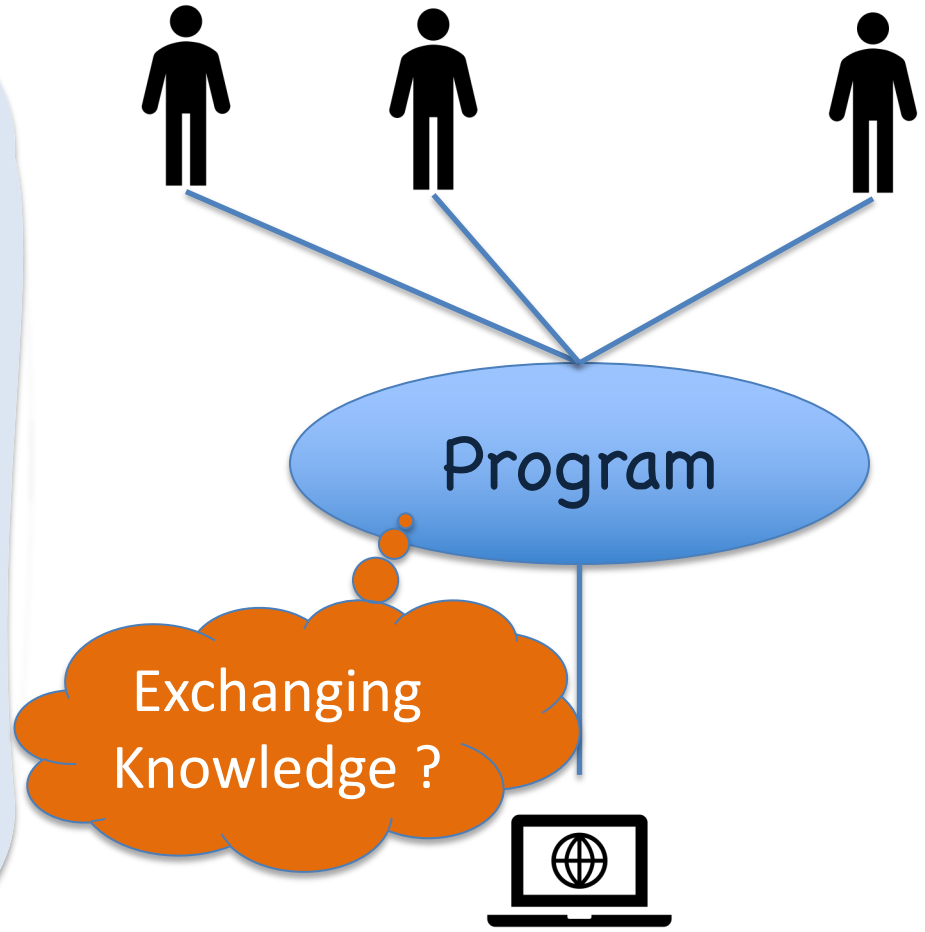
- Natural language was invented in human social life and evolves with the evolution of human social life
  - Inherent naturalness
- While, Program languages, artificial languages
  - instead of being an act of communication, from one human to another
  - they are ways to tell computers what to do

A. Hindle, E. T. Barr, M. Gabel, Z. Su and P. Devanbu, On the Naturalness of Software, Communication of the ACM, 59(5): 122-131, 2016

Communication with both Human and Machine

Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do

Knuth, D. E. Literate programming, Comput. J. 1984

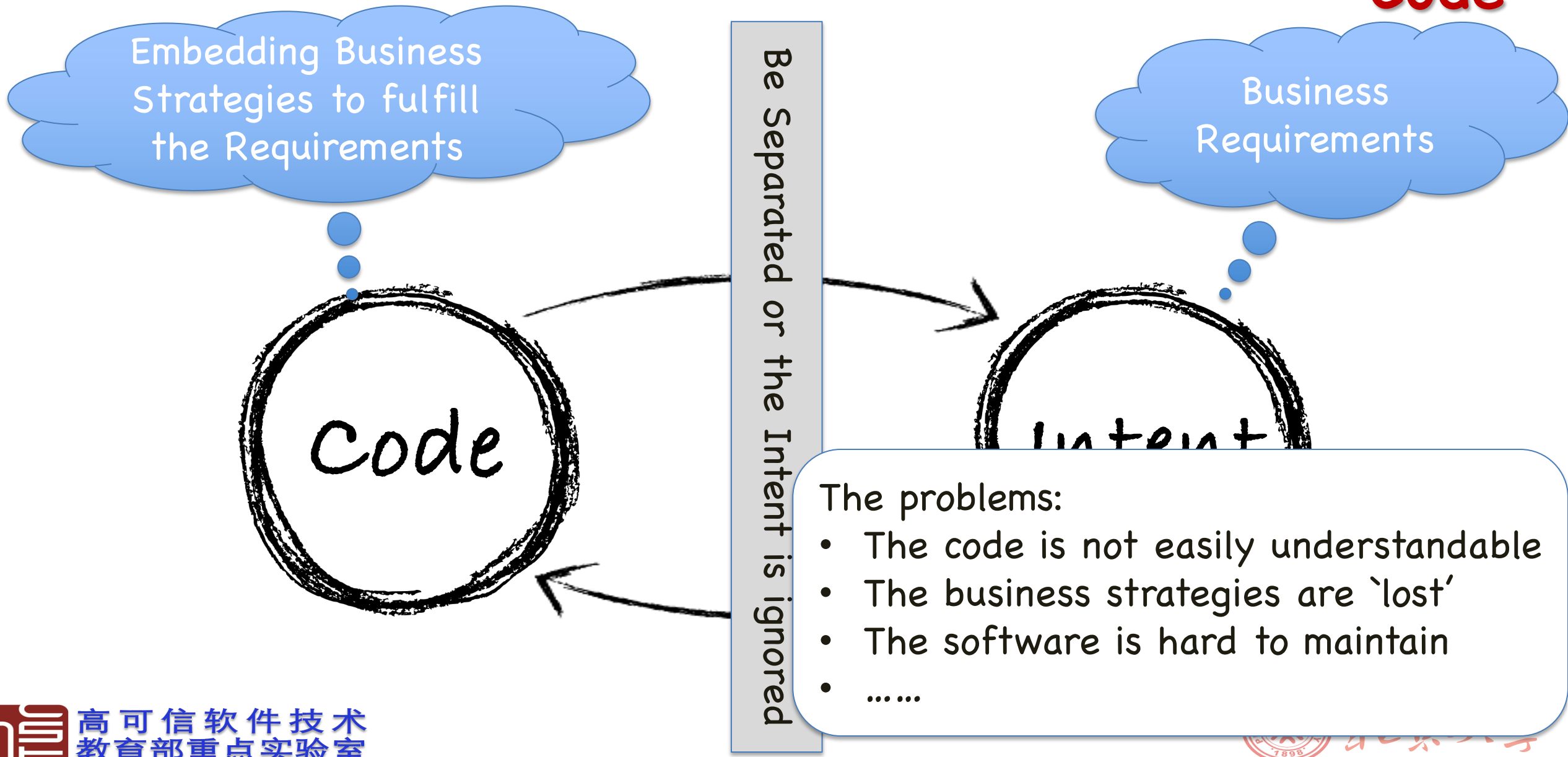


A. Hindle, E. T. Barr, M. Gabel, Z. Su and P. Devanbu, On the Naturalness of Software, Communication of the ACM, 59(5): 122-131, 2016

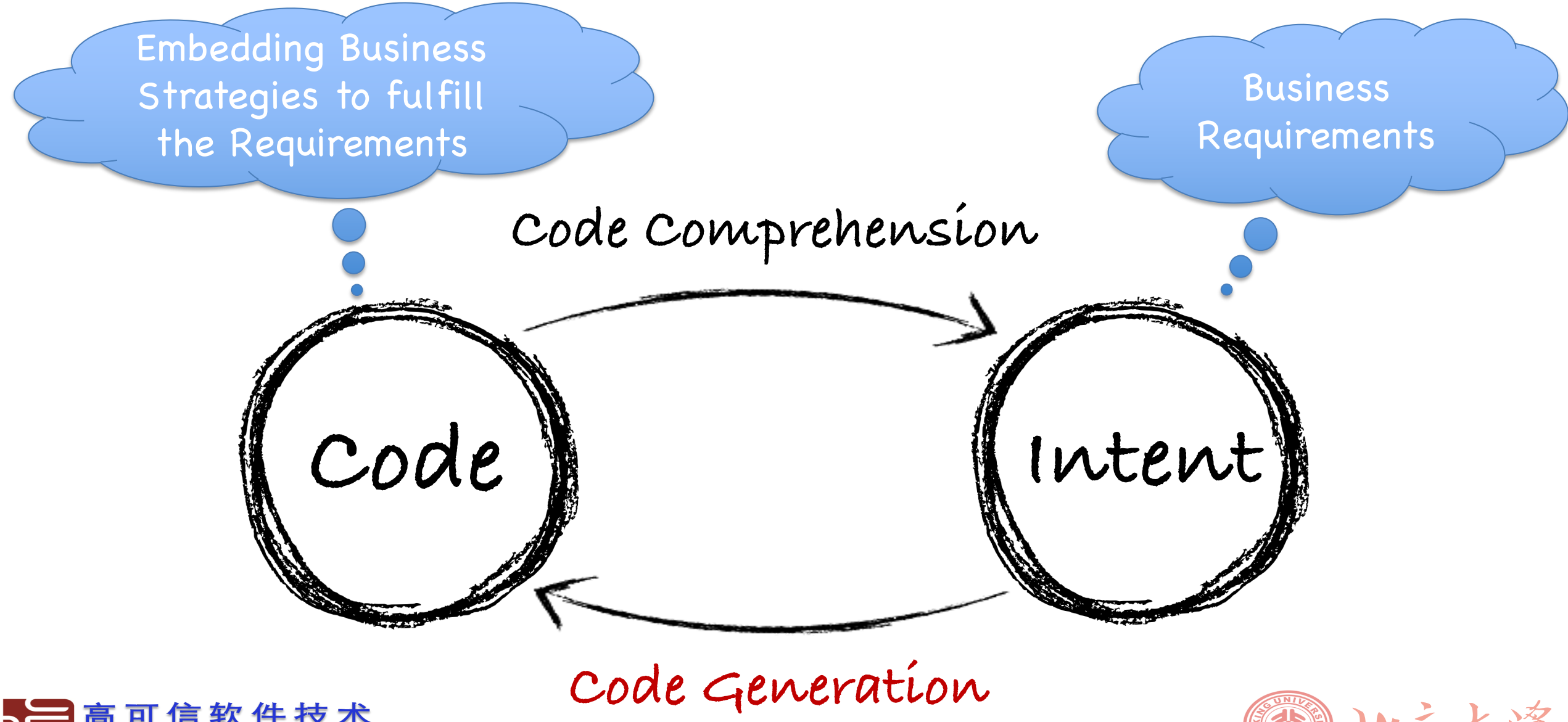
Techniques for the NLP Transformed to Program Language Process

- Source code in program languages exhibits a good level of repetitive and predictability
  - e.g. "for (int := 0, i<n, i++)" occur frequently, name convention, …
- Code regularities / patterns can also be captured by like the n-gram statistical language model ?
  - at lexical level
  - at semantic level in the sense of
    - semantic annotations, e.g. date types,
    - sematic roles, e.g. variable, operator, keyword, function call, …
    - pairwise association, e.g. begin-end, ……
    - etc.

Motivation of Learning from the Source Code



The Purpose to Learn from Source Code



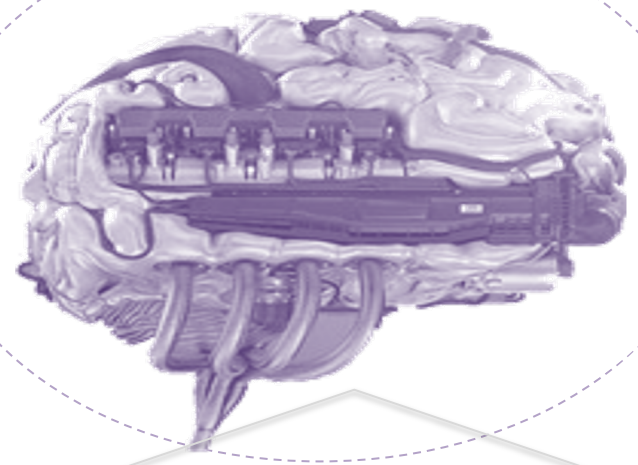
Agenda

- Can Machine Learn from Source Code, and Why
  - Be Inspired by Natural Language, naturalness
- How Machine learns from Source Code
  - Code/Program/Software Comprehension
  - Find the Features of Source Code
- Learn What
  - Learn Task Specific Knowledge and tacit knowledge
  - AiXCoder: A programming assistant
- What can Do Next
  - Combine with Knowledge Graph

The Possibilities

Code Representation Model

Code



Intent

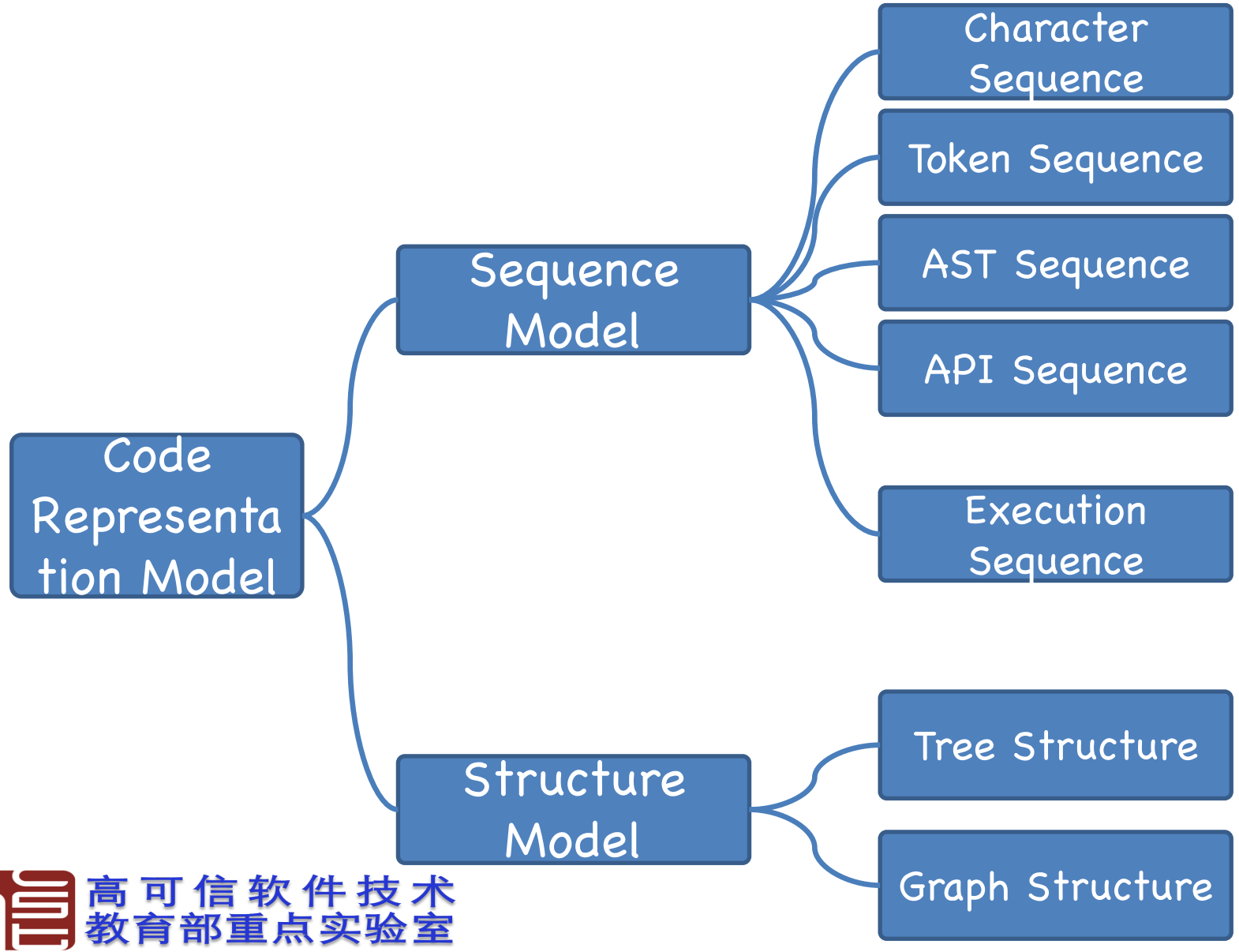
Big Code Base

Open source movement

Legacy software

Out Sourcing Annotation

Types of Code Representation Models

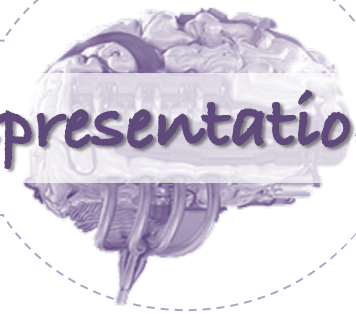


类型	文献	模型	应用
Token	Raychev 等人[46]	N-gram, RNN	代码补全
	White 等人[54]	RNN	代码补全
	Bhoopchand 等人[7]	LSTM+Pointer	代码补全
	Allamanis 等人[3]	CNN+Attention	函数名预测
	Iyer 等人[30]	RNN+Attention	代码注释生成
	Louis 等人[38]	Seq2Seq	冗余注释检测
	Gupta 等人[20]	LSTM+RL	代码纠错
字符	Commins 等人[13]	LSTM	代码生成
应用程序接口（API）	Gu 等人[18]	Seq2Seq	API 序列检索
	Gu 等人[17]	RNN,MLP	代码检索
抽象语法树节点	Li 等人[34]	LSTM+Pointer	代码补全
	Liu 等人[37]	LSTM	代码补全
	Yin 等人[59]	LSTM	程序生成

类型	文献	模型	应用
程序执行过程	Reed 等人[47]	LSTM	程序综合
	Cai 等人[10]	LSTM	程序综合
	Xiao 等人[58]	LSTM	程序综合
	Chen 等人[11]	LSTM+RL	程序综合
	Wang 等人[60]	GRU	程序错误分类

类型	文献	模型	应用
抽象语法树	Mou 等人[43]	CNN	程序分类
	Li 等人[34]	LSTM+Pointer	代码补全
	Liu 等人[37]	LSTM	代码补全
	Yin 等人[59]	Seq2Seq	程序生成
	Rabinovich 等人[43]	Seq2Seq	程序生成
	Alon 等人[5]	CRF	变量名预测
	White 等人[53]	RNN	代码克隆检测
	Wei 等人[52]	LSTM	代码克隆检测
图	Li 等人[35]	GRU	程序验证
	Allamanis 等人[2]	GRU	变量名命名
	Allamanis 等人[1]	GRU	代码修复

Usage Scenarios of Code Representation Models



Code Representation Model

Code Comprehension

Code Recognition

- Code Clone Detection
- Code Classification
- Malicious Code Detection

Code Pattern Detection

- Bad Smell Detection
- Code Defect Detection
- Bug Localization

Code to NL

- Code Summarization
- Code Comments Generation
- Commit Message Generation

Code Generation

- Code Generation
- Code Completion
- Code Refactoring
- Bug Auto-fixing

The Challenges and Opportunities to Use Techniques for NLP in Source Code Learning

- **Vocabulary**: In natural language, the vocabulary is usually limited to the most common words, e.g., 30,000 words, and words outside the vocabulary are treated as unknown words
- But for source code,

Table 1: Statistics for code snippets in our dataset

#Methods	#All Tokens	# All Identifiers	# Unique Tokens	#Unique Identifiers
69,708	8,713,079	2,711,496	234,146	234,05

85% identifiers and 30% tokens will be treated as <UNK>, if using the most common 30000 tokens
That may make the techniques useless.

The Challenges and Opportunities to Use Techniques for NLP in Source Code Learning

- **Structure:** Source code is strong structured while natural language text is weakly structured
  - Program contains explicit and hierarchical structure
    - How to take advantage of rich and unambiguous structure information of source code to boost effectiveness
  - Program contains code blocks of different granularity, e.g. statements, loops, methods, classes, etc.
    - They are nested and composable. The nesting can be very deep, leading to long dependency
    - There are differences in semantics between code blocks

The Challenges and Opportunities to Use Techniques for NLP in Source Code Learning

- Not only the sequence of the sentences / tokens / characters, Different Flows imply Semantics:
  - e.g. control flow, data flow, these capture the executive semantics of program, which is closer to the functionalities
- But these flows led to the graphical structure that is much more difficult to deal with

The Purpose of Learning from Language

For machine learning from Natural Language

(1) task oriented: build a model via training for specific tasks, like translation, cloze, dialog, etc.

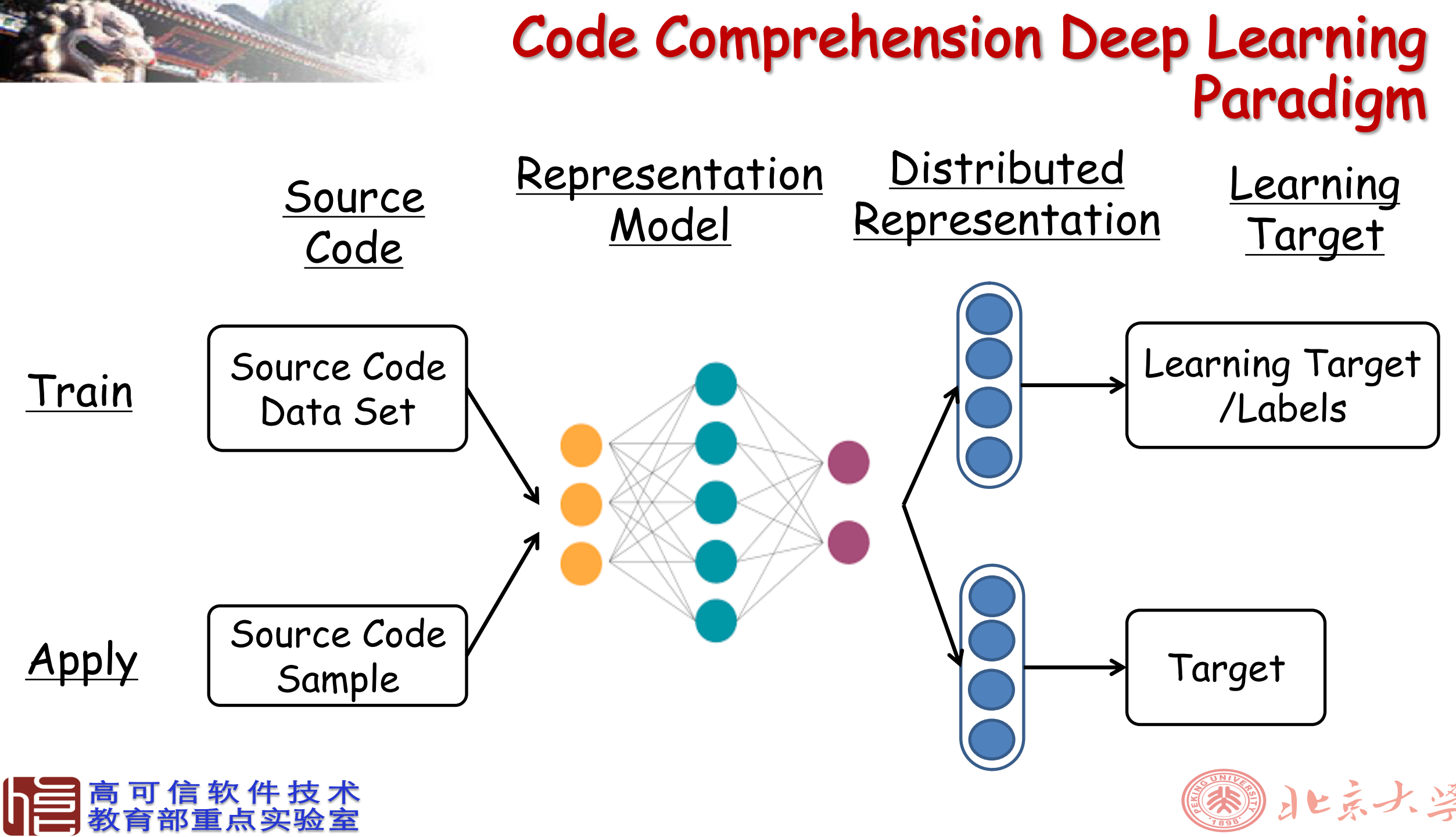
(2) general purpose: build a model for entity/relation identification to build like knowledge graph for human

For machine learning from Source code:

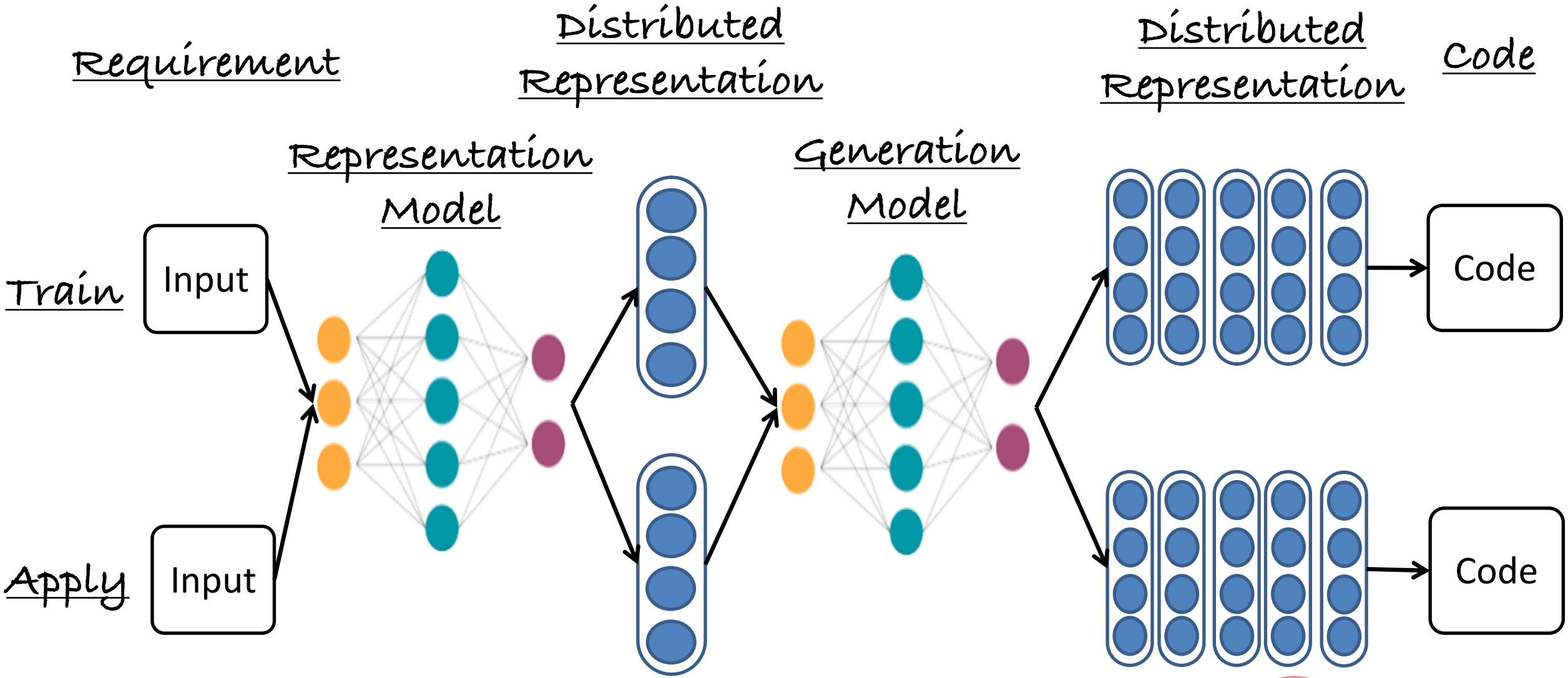
(1) task oriented: build a model via training for specific tasks, like code summarization, code searching, code clone detection, code bad smell detection, etc.

(2) general purpose: build a model for what ?

Code Comprehension Deep Learning Paradigm



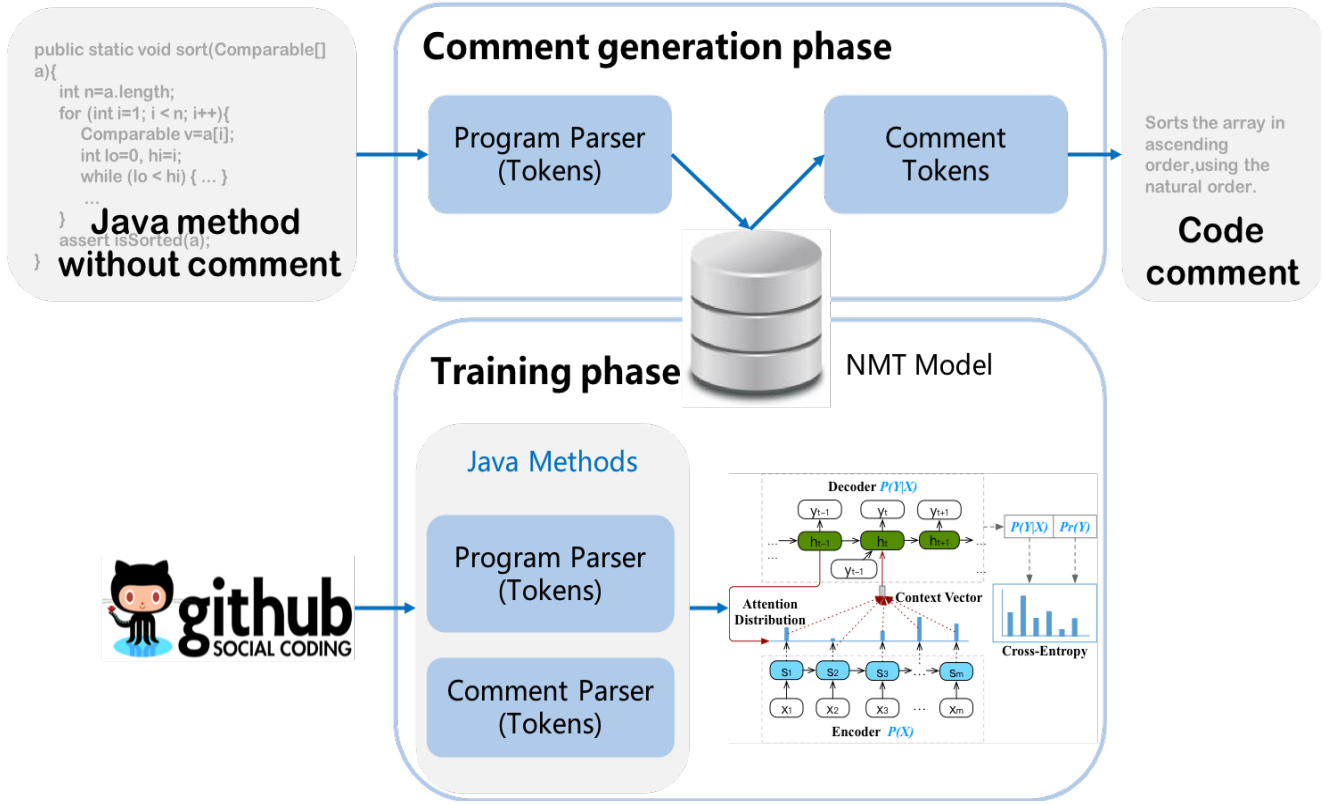
Code Generation Deep Learning Paradigm



Agenda

- Can Machine Learn from Source Code, and Why
  - Be Inspired by Natural Language, naturalness
- How Machine learns from Source Code
  - Code/Program/Software Comprehension
  - Find the Features of Source Code
- **Learn What**
  - **Learn Task Specific Knowledge and tacit knowledge**
  - **AiXCoder: A programming assistant**
- What can Do Next
  - Combine with Knowledge Graph

Code Summarization: compared with machine translation



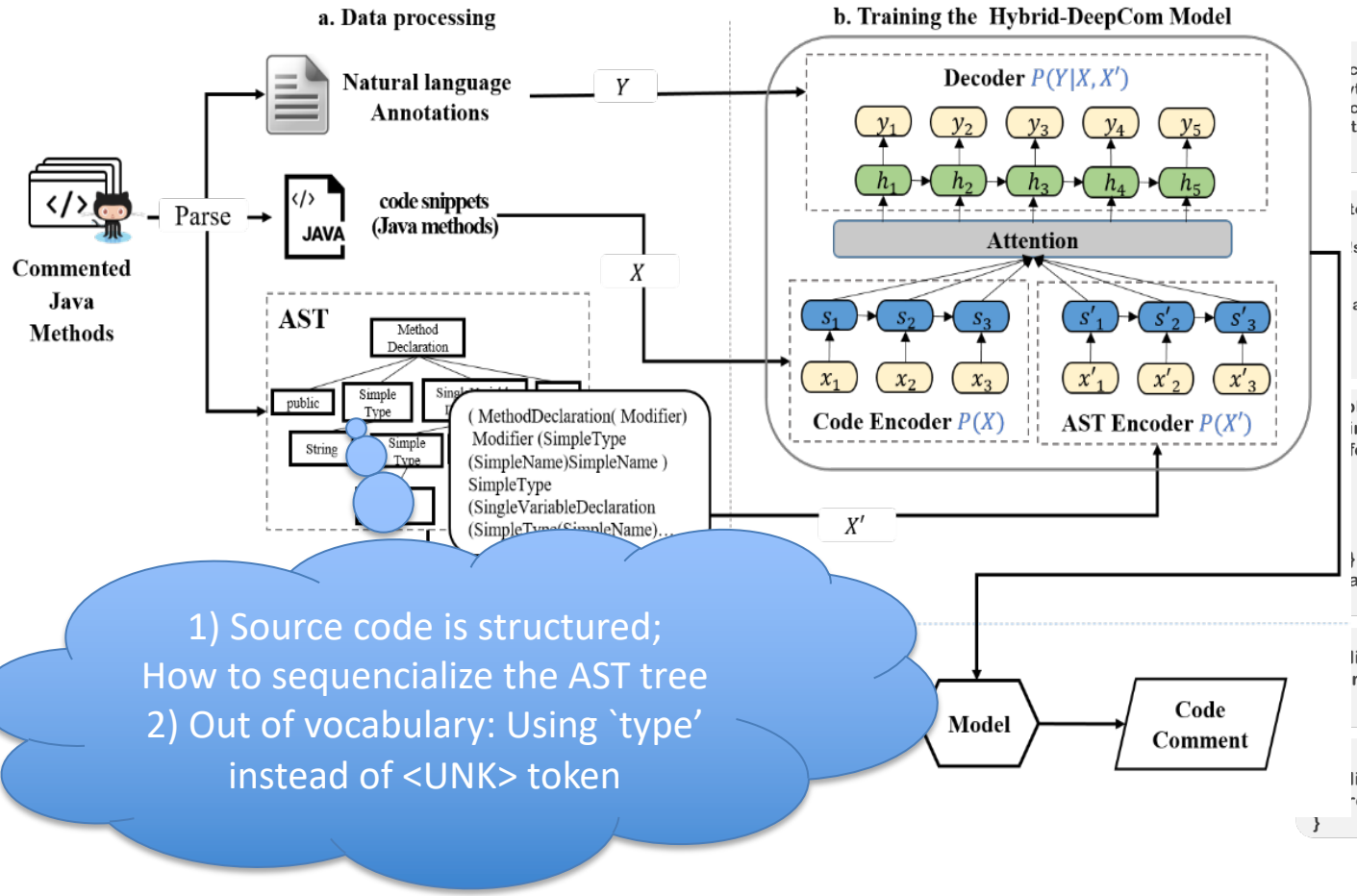
Research Questions:

1) Source code is structured, e.g. AST; How to sequencialize the AST tree ?

2) How to deal with the problem of Out of vocabulary ?

3) Is external knowledge useful to make improvement ?

Code Summarization: Structure and Out-of-Vocabulary



1) Source code is structured;
How to sequencialize the AST tree
2) Out of vocabulary: Using `type' instead of <UNK> token

```
c static byte[] bitmapToByte(Bitmap b){
teArrayOutputStream o = new ByteArrayOutputStream();
compress(Bitmap.CompressFormat.PNG,100,o);
turn o.toByteArray();
```

DeepCom: convert Bitmap to byte array
Human-Written: convert Bitmap to byte array

```
e static void addDefaultProfile(SpringApplication app,
        SimpleCommandLinePropertySource source){
source.containsProperty("spring.profiles.active")
        &&!System.getenv().containsKey("SPRING_PR
                                OFILES_ACTIVE")){
app.setAdditionalProfiles
        (Constants.SPRING_PROFILE_DEVELOPMENT);
```

DeepCom: If no profile has been configured , set by default the "dev" profile.
Human-Written: If no profile has been configured , set by default the "dev" profile.

```
lic static void sort(Comparable[] a){
int n=a.length;
for (int i=1; i < n; i++){
    Comparable v=a[i];
    int lo=0, hi=i;
    while (lo < hi) { ... }
    ...
}
assert isSorted(a);
```

DeepCom: Sorts the array in ascending order, using the natural order.
Human-written: Rearranges the array in ascending order, using the natural order.

```
lic boolean isEmpty(){
return root == null;
```

DeepCom: Returns true if the symbol is empty.
Human-written: Is this symbol table empty?

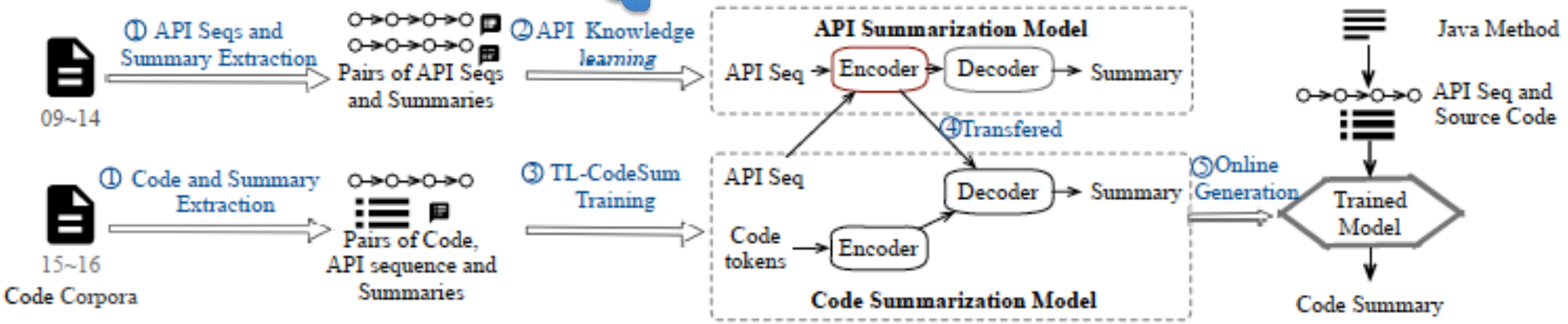
```
lic boolean contains(int key){
return rank(key) != -1;
}
```

DeepCom: Checks whether the given object is contained within the given set.
Human-written: Is the key in this set of integers?

Xing Hu, Ge Li, Xia Xin, David Lo, Zhi Jin, Deep Code Comment Generation, ICPC 2018 (ACM Distinguish Paper Award)
Xing Hu, Ge Li, Xin Xia, David Lo, Zhi Jin, Deep Code Comment Generation with hybrid lexical and syntactical information, Empirical Software Engineering, 25(3): 2179-2217 (2020)

Code Summarization: Knowledge Argumentation

1) Training a model that builds the mappings from API sequence to its corresponding natural language description;
2) Transferring the API knowledge to the code summarization task



Approaches	Precision	Recall	F-score
CODE-NN	26.21	14.17	18.40
API-Only	30.72	21.14	25.05
Code-Only	38.89	28.81	33.10
API+Code	41.06	30.34	34.90
TL-CodeSum(fixed)	**42.20**	34.38	37.89
TL-CodeSum(fine-tuned)	40.78	**35.41**	**37.91**

Approaches	BLEU score	METEOR
CODE-NN	25.3	6.92
API-Only	26.45	10.71
Co		
Al		
TL-Co		
TL-Code		

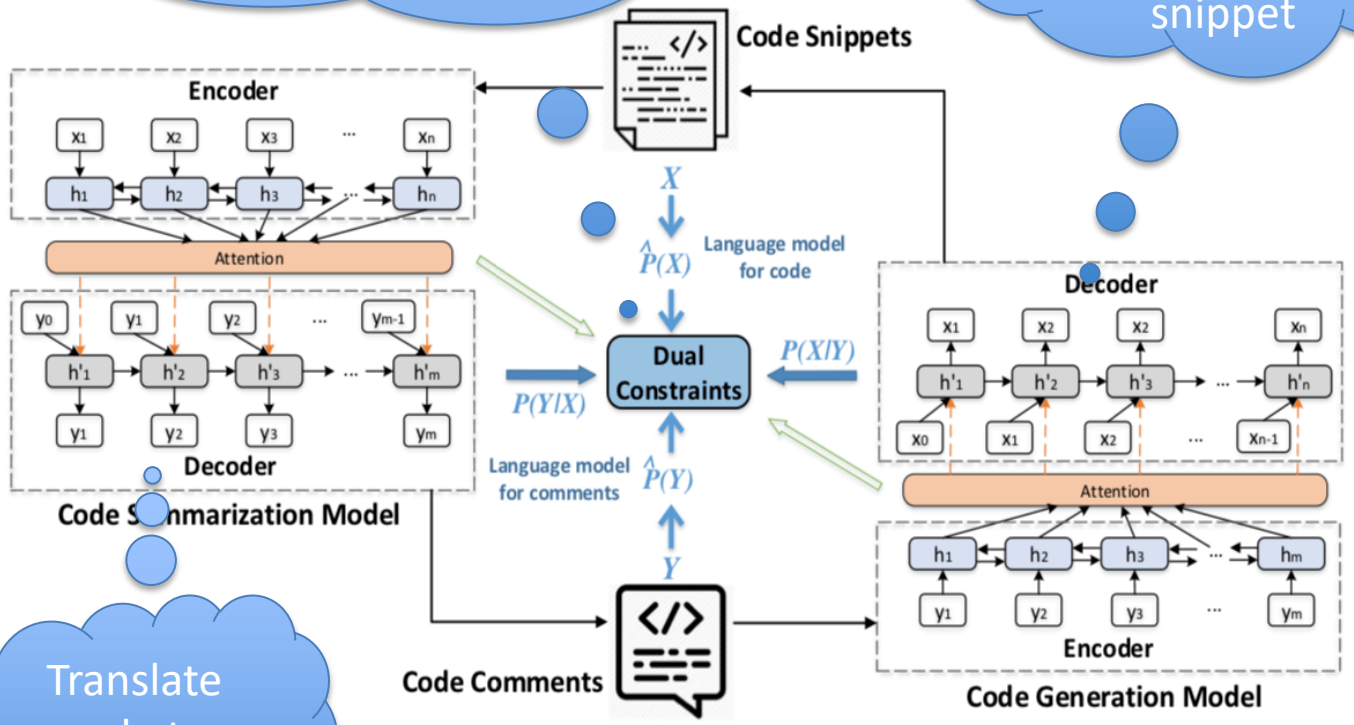
What we learn here: Integrating external knowledge is beneficial and effective. That makes the approach significantly outperforms others.

Xing Hu, Ge Li, Xin Xia, David Lo, Shuai Lu, Zhi Jin, Summarizing Source Code with Transferred API Knowledge, IJCAI 2018

Code Summarization: of Code Summarization

adding regularization terms in the loss function to constrain the duality between two models

Translate natural language description to source code snippet



Translate code to comments

Table 2: The overall performance of our CS models compared with baselines

Methods	Java			Python		
	BLEU	METEOR	ROUGE-L	BLEU	METEOR	ROUGE-L
CODE-NN	27.60	12.61	41.10	17.36	9.288	37.81
DeepCom	39.75	23.06	52.67	20.78	9.979	37.35
Tree2Seq	37.88	22.55	51.50	20.07	8.957	35.64
RL+Hybrid2Seq	38.22	22.75	51.91	19.28	9.752	39.34
API+CODE	41.31	23.73	52.25	15.36	8.571	33.65
Basic Model	41.01	23.26	51.64	20.47	10.38	38.77
Dual Model	**42.39**	**25.77**	**53.61**	**21.80**	**11.14**	**39.45**

Table 4: Ablation study of different settings on CS task. Model (M) 1 is the basic model of independent training.

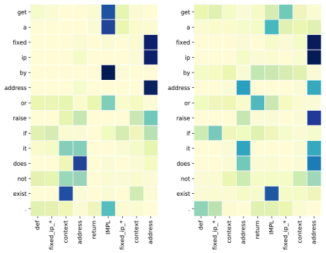
M	Probabilistic Duality	Attention Duality	Java			Python		
			BLEU	METEOR	ROUGE-L	BLEU	METEOR	ROUGE-L
1	-	-	41.01	23.26	51.64	20.47	10.38	38.77
2	✓	-	41.73	25.54	53.60	21.66	10.81	38.83
3	-	✓	41.96	25.80	53.57	21.57	10.91	39.07
4	✓	✓	42.39	25.77	53.61	21.80	11.14	39.45

Source Code: public static void closeQuiet(@Nullable Closeable closeable){
if (closeable != null){
try{ closeable.close();}
catch(IOException ignored){}
}
}
Human-Written: closes resource without reporting any error.
Dual Model: quietly closes given closeable without reporting.

(a) CS task

Comment: prints an integer to standard output and flushes standard output.
Human-Written: public static void print(Object x){out.print(x); out.flush();}
Dual Model: public static void print(int x){out.print(x); out.flush();}

(b) CG task



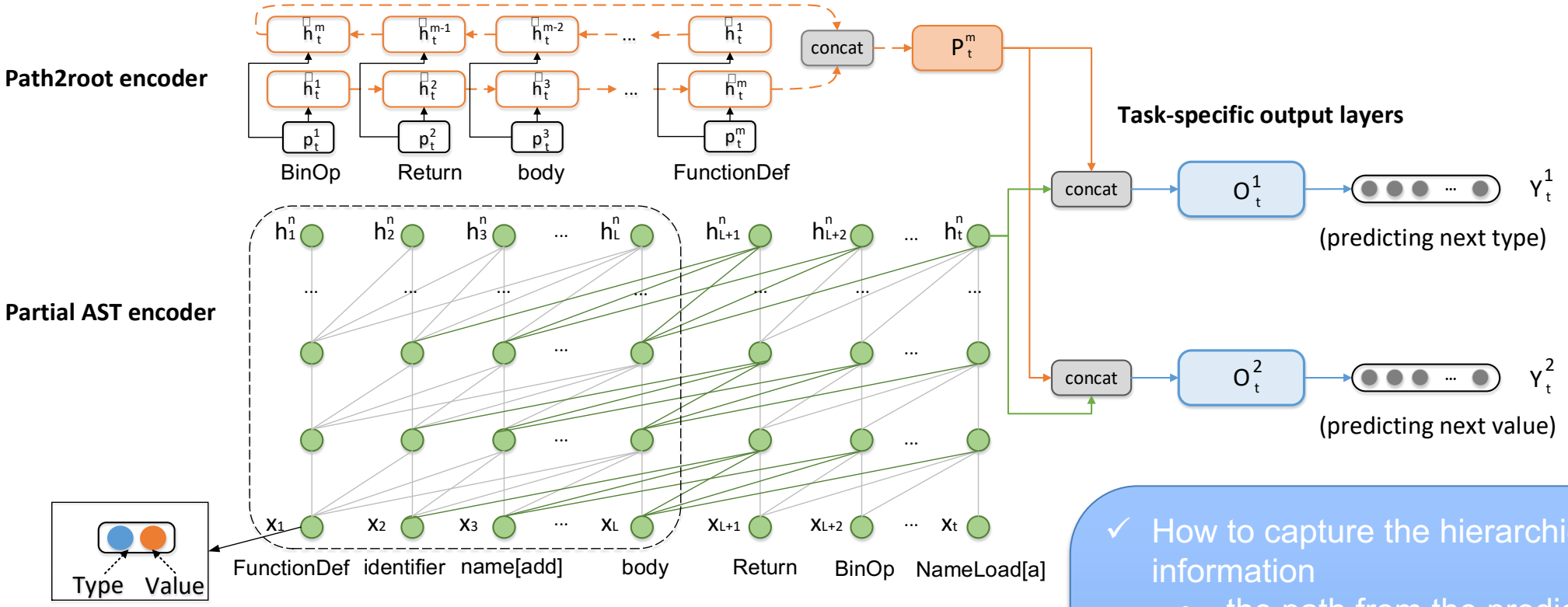
(c) Basic model (d) Dual model

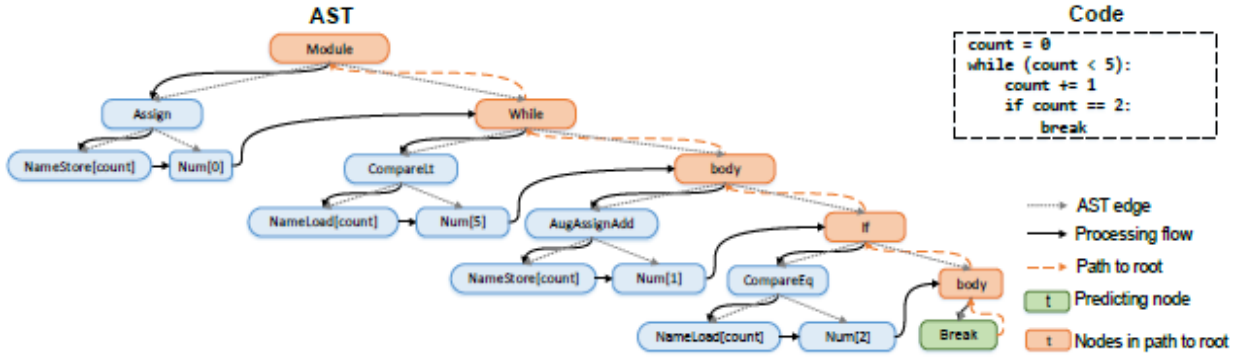
Table 3: BLEU scores and percentage of valid code (PoV) on CG task

Methods	Java		Python	
	BLEU	PoV	BLEU	PoV
SEQ2TREE	13.80	22.6%	4.472	22.7%
Basic Model	10.86	19.6%	10.43	41.8%
Dual Model	**17.17**	**27.4%**	**12.09**	**51.9%**

Bolin Wei, Ge Li, Xin Xia, Zhiyi Fu, Zhi Jin, Code Generation as a Dual Task of Code Summarization, NeurIPS 2019

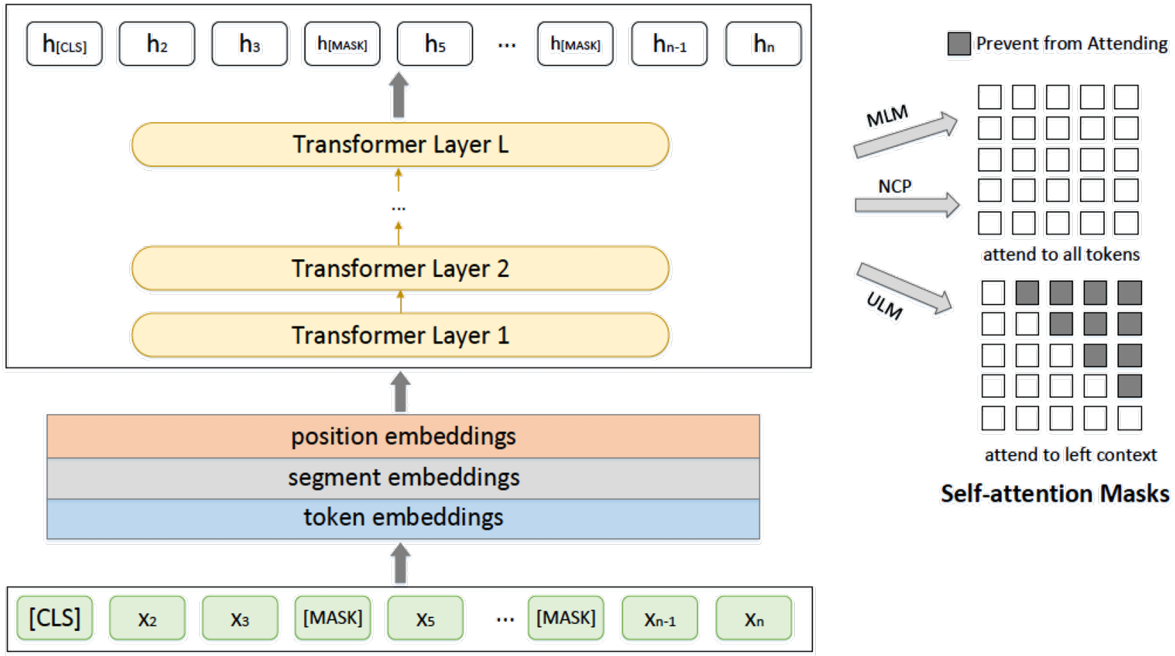
Code Completion: Self-Attention Architecture with Multi-Task Learning





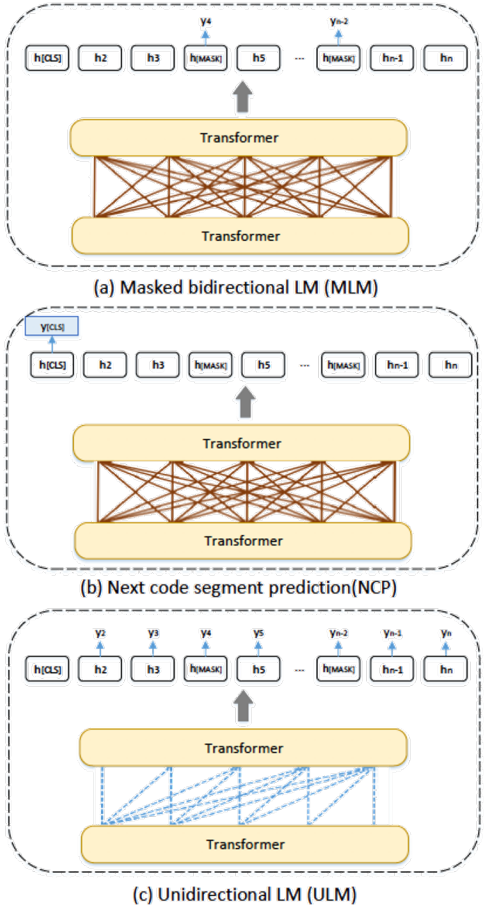
- ✓ How to capture the hierarchical structural information
  - the path from the predicting node to the root node
- ✓ How to deal with very long-term dependency
  - Adopt transformer-XL as the base language model
- ✓ Code completion is not a single task
  - need multiple task learning

Code Completion: Multi-task Learning based Pre-trained Language Model



Code Understanding and Generation Pre-trained LM with Shared Parameters

Self-attention Masks



Masked bidirectional language model: mask the identifiers. The objective is learning to predict the masked tokens based on the bidirectional context.

Next Code Segment Prediction: understand the relationships between code segments by pre-training a binarized next code segment predicting task

Unidirectional Language Modeling: left-to-right language modeling task because for the generation (completion), only leftward contextual tokens are allowed

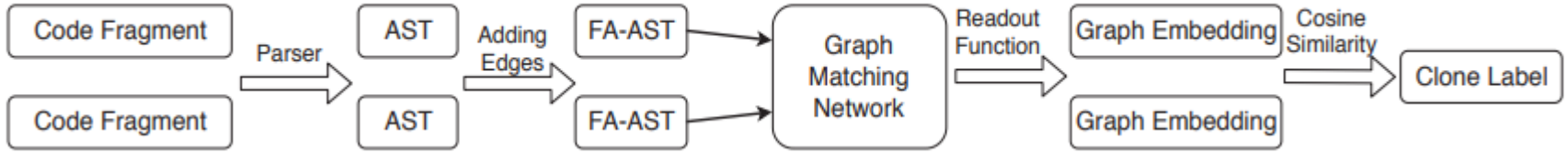
Figure 2: Overview of CugLM pre-training. The model parameters are shared across the pre-training objectives (i.e., MLM, NCP, and ULM). We use different self-attention masks to control the access to context for each token.

Fang Liu, Ge Li, Yunfei Zhao, Zhi Jin, Multi-task Learning based Pre-trained Language Model for Code Completion, ASE 2020, Accepted.

Code Clone Detection: Graph Neural Network and Flow-Augmented AST

The challenge is Detect the semantic similarity on the implementation of functionalities

Explicitly augment the control/data flow with AST to capture the execution traces




```
protected String downloadURLtoString(URL url) throws IOException {
    BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
    StringBuffer sb = new StringBuffer(100 * 1024);
    String str;
    while ((str = in.readLine()) != null) {
        sb.append(str);
    }
    in.close();
    return sb.toString();
}

public static String fetchUrl(String urlString) {
    try {
        URL url = new URL(urlString);
        BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream()));
        String line = null;
        StringBuilder builder = new StringBuilder();
        while ((line = reader.readLine()) != null) {
            builder.append(line);
        }
        reader.close();
        return builder.toString();
    } catch (MalformedURLException e) {
    } catch (IOException e) {
    }
    return "";
}
```

Are they similar? Both syntactic and semantic

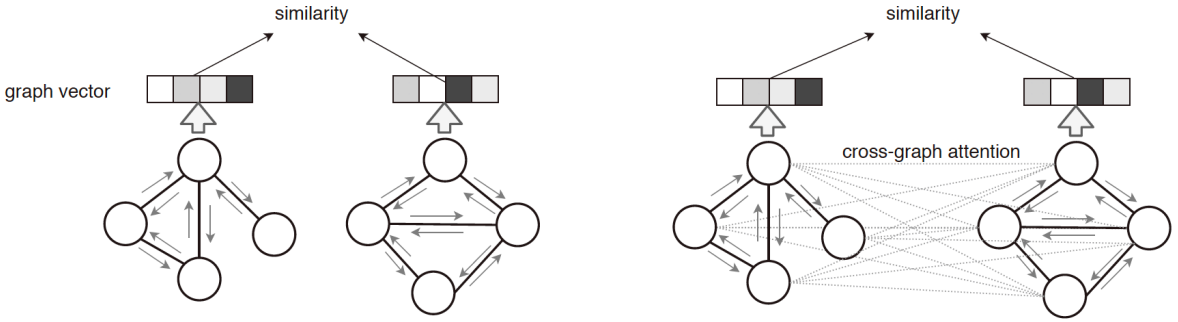


Fig. 7. Basic architecture of the GGNN embedding model (left) and GMN model (right).

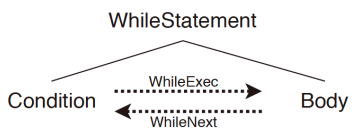


Fig. 4. Control flow edges for While statements.

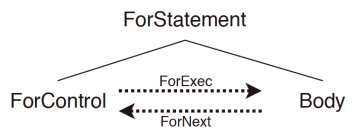


Fig. 5. Control flow edges for For statements.

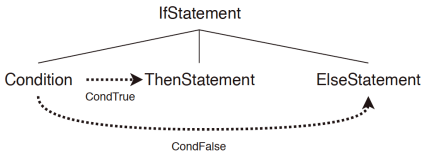
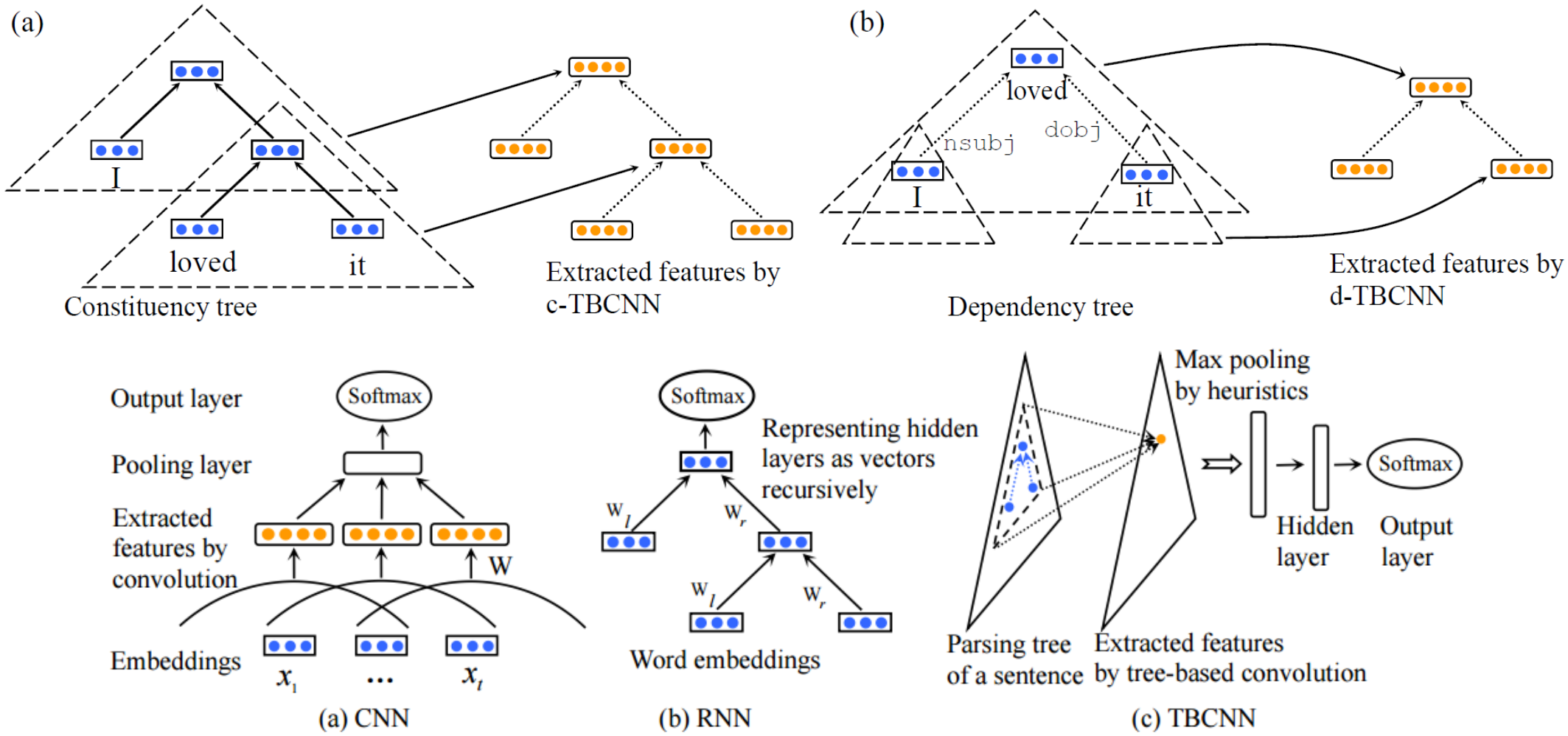


Fig. 3. Control flow edges for If statements.

Wenhan Wang, Ge Li, Bo Ma, Xin Xia, Zhi Jin, Detecting Code Clones with Graph Neural Network and Flow-Augmented Abstract Syntax Tree,. (SANER 2020)

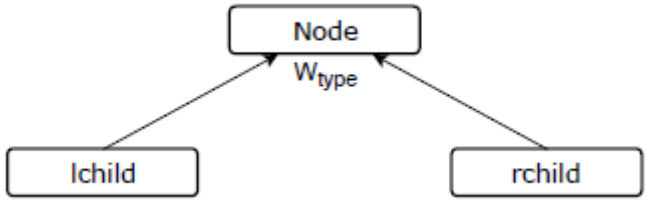
CNN over Tree Structure, AST

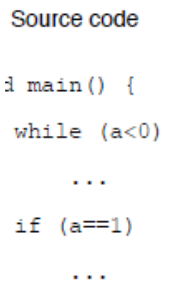


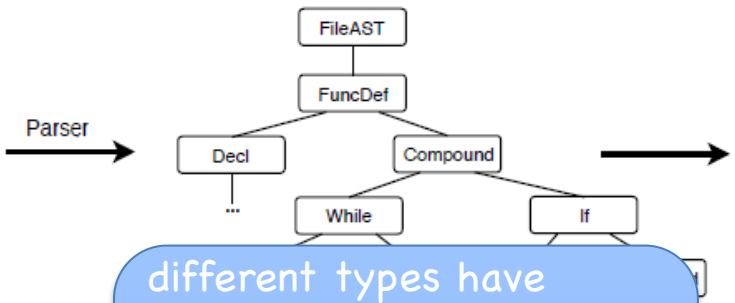
Lili Mou, Ge Li, Lu Zhang, Tao Wang, Zhi Jin, Convolutional Neural Networks over Tree Structures for Programming Language Processing, AAAI 2016

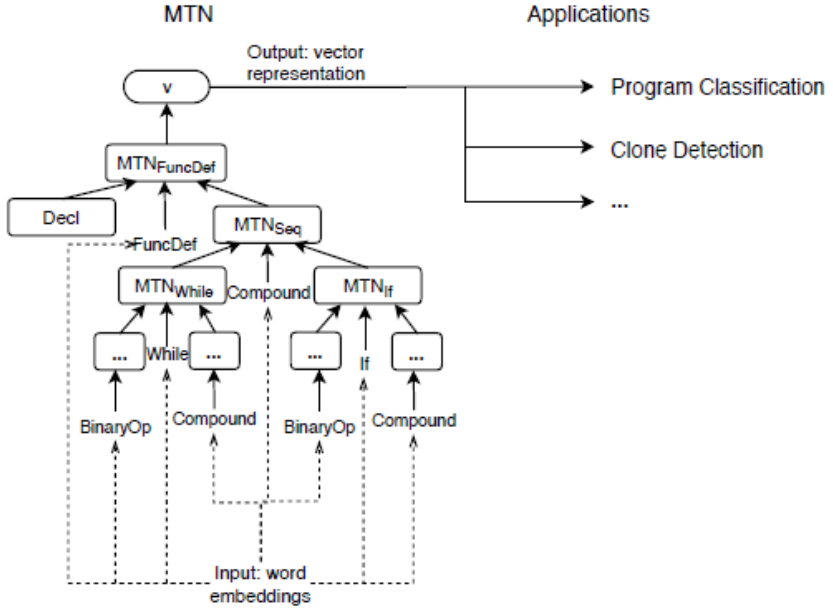
Lili Mou, Zhi Jin, Tree-Based Convolutional Neural Networks: Principles and Applications, Springer 2018.

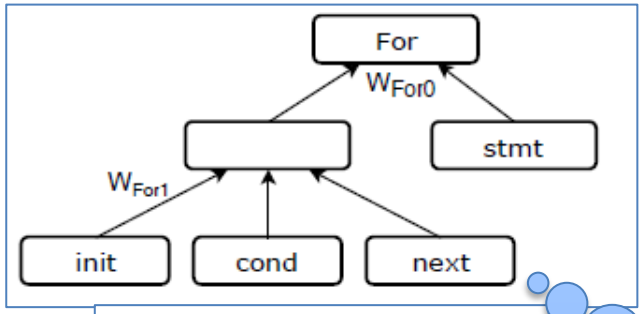
Modular Tree Network for Source Code Representation Learning

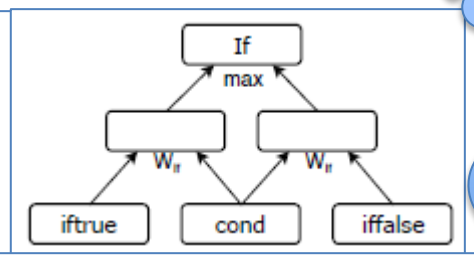


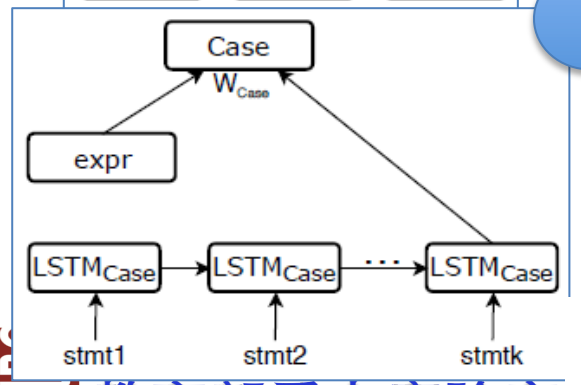












different types have different meanings on the structure
The idea is different neural modules for different subtrees

AST contains multiple subtree types: For, While, If...

Unit	Function	Parameters
decl, body	$h_{FuncDef} = tanh(W_{FuncDef}[h_{decl} : h_{body}] + b_{FuncDef})$	$2d^2 + d$
cond, stmt	$h_{While} = tanh(W_{While}[h_{cond} : h_{stmt}] + b_{While})$	$2d^2 + d$
hile→ cond, stmt	$h_{DoWhile} = tanh(W_{DoWhile}[h_{cond} : h_{stmt}] + b_{DoWhile})$	$2d^2 + d$
or→ init, cond, next, stmt	$W_{For0}[tanh(W_{For1}[h_{init} : h_{cond} : h_{next}] + b_{For1}) : h_{stmt}] + b_{For0}$	$5d^2 + 2d$
If→ cond, iftrue	$h_{If} = tanh(W_{If}[h_{cond} : h_{iftrue}] + b_{If})$	$2d^2 + d$
If→ cond, iftrue, itfalse	$h_{If} = max(tanh(W_{If}[h_{cond} : h_{iftrue}] + b_{If}), tanh(W_{If}[h_{cond} : h_{iffalse}] + b_{If}))$	
Switch→ cond, stmt	$h_{Switch} = tanh(W_{Switch}[h_{cond} : h_{stmt}] + b_{Switch})$	$2d^2 + d$
Case→ expr, stmt1, ..., stmtk	$h_{Case} = tanh(W_{Case}[h_{expr} : LSTM_{Case}(h_{stmt1}, ..., h_{stmtk})] + b_{Case})$	$10d^2 + 5d$
Seq→ child1, ..., childk	$h_{seq} = LSTM_{seq}(h_{child1}, ..., h_{childk})$	$8d^2 + 4d$

Wenhan wang, Ge Li, Sijie Shen, Xin Xia and Zhi Jin, Modular Tree Network for Source Code Representation Learning, ACM TOSEM (accepted)


aiXcoder
智能编程机器人 • 为开发者而生
#coding:utf-8
import tensorflow as tf
import input_data
import time
"""
权重初始化
"""
def weight_variable(shape):
initial = tf.truncated_normal(shape, stddev =0.1)
shape, mean: float=0.0, stddev: float=1.0, dtype: DType=dtypes.float32, seed=None, name=None
shape, dtype, seed: int=0, seed2: int=0, name=None
aiXcoder2.0上线一个月
国际下载量超过13万！

keystore-management – ToKeyStoreMetadataPersister.java [core]

keystore-management 〉 core 〉 src 〉 main 〉 java 〉 de 〉 adorsys 〉 keymanagement 〉 core 〉 metadata 〉 ToKeyStoreMetadataPersister 〉 isMetadataEntry

Add Configuration... Git:

TemplateMappingGeneratorImpl.java | KeySetTemplate.java | MetadataPersistenceConfig.java | ToKeyStoreMetadataPersister.java | KeyStore.java

```
29
30      private static final String METADATA_SUFFIX = "KEYMETADATA";
31      private static final Pattern METADATA_PATTERN = Pattern.compile("(.+):([0-9A-F]+)-KEYMETADATA$");
32
33      private final MetadataPersistenceConfig persistenceConfig;
34      private final SecretKeyGenerator secretKeyGenerator;
35      private final KeyDecoder decoder;
36
37      @Inject
38
39
40
41      @Override
42      public boolean isMetadataEntry(String forAlias, KeyStore keyStore) {
43          if (!forAlias.endsWith(METADATA_SUFFIX)) {
44              return false;
45          }
46
47          Matcher matcher = METADATA_PATTERN.matcher(forAlias);
48          if (!matcher.matches()){
49              return false;
50          }
51          String keyAlias = matcher.group(1);
52          String crc32 = matcher.group(2);
53
54          return crc32.equals(crc32(keyAlias));
55      }
56
57      @Override
58      @SneakyThrows
59
60      public KeyMetadata extract(String forAlias, KeyStore keyStore) {
61          if (isMetadataEntry(forAlias, keyStore)) {
62              return null;
63          }
64
65          String alias = metadataAliasForKeyAlias(forAlias);
66
```

aiXcoder Welcome

aiXcoder

(IntelliJ plugin for Java)

Search code examples from billions li of open source code:

Open Code Search Engine in Browser...

Set the balance between response t and recommended code length:

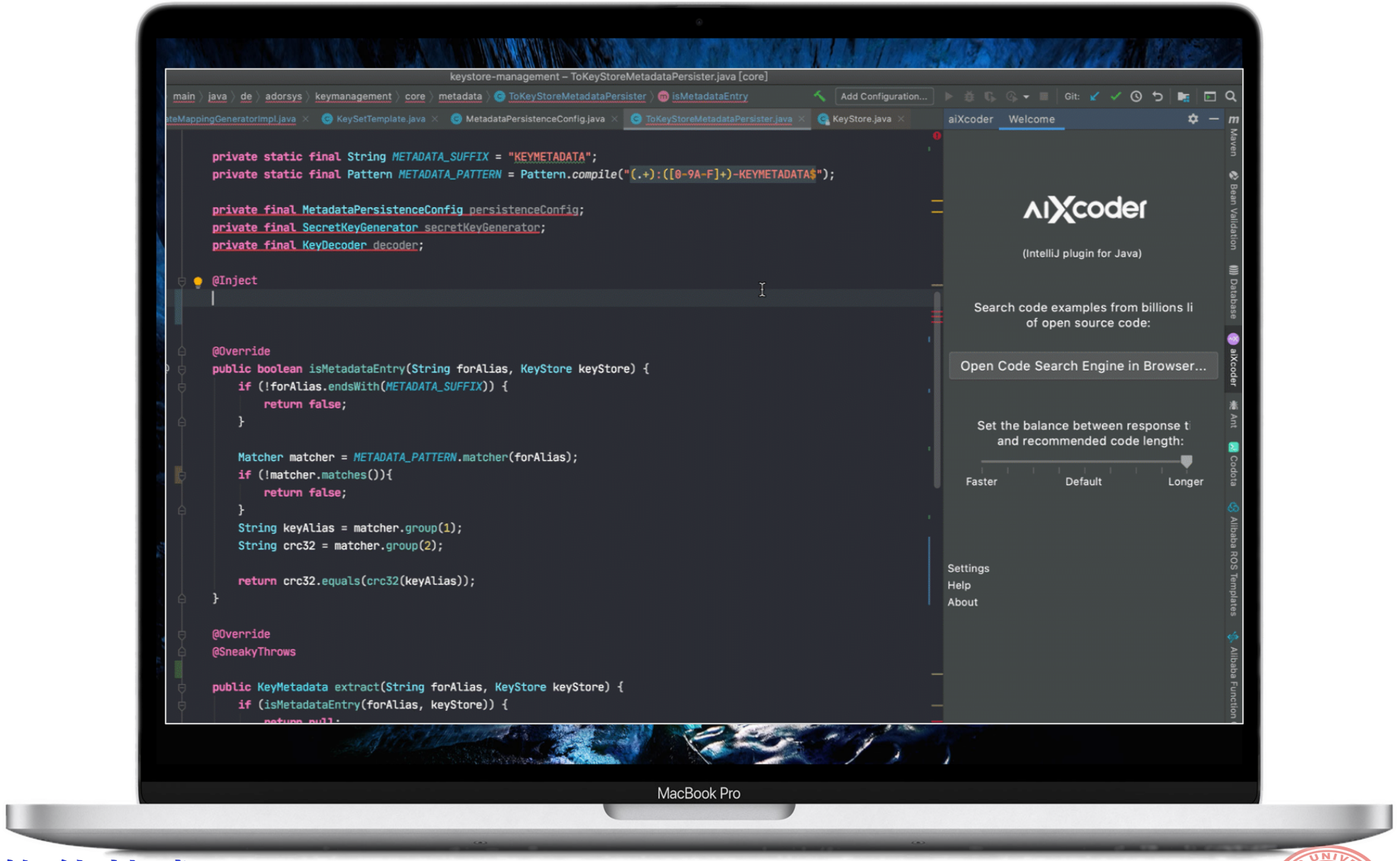
Faster Default Longer

Settings
Help
About

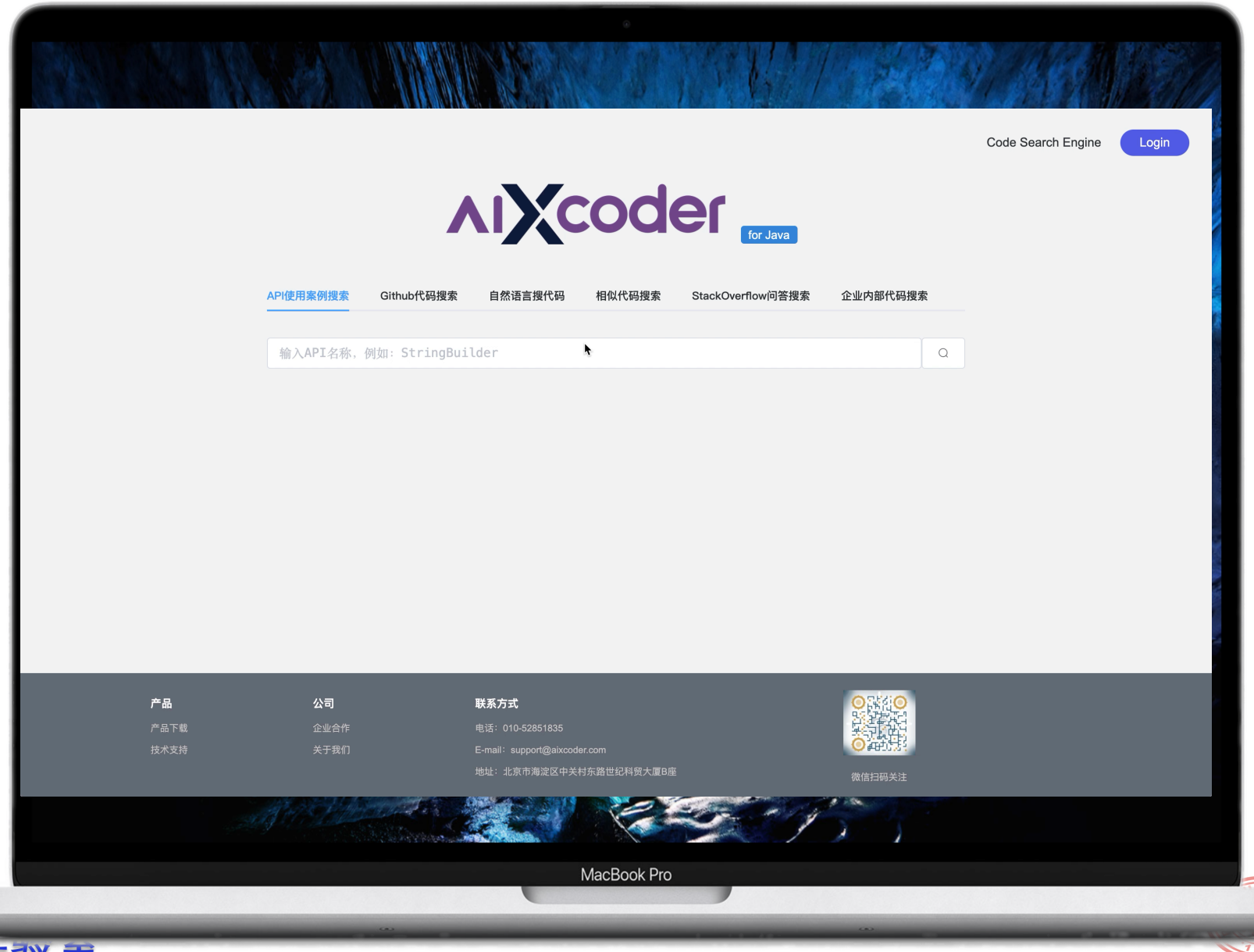
9: Git 6: TODO Alibaba Cloud View Java Enterprise Endpoints Spring Terminal Event Log

38:5 LF UTF-8 4 spaces develop

aiXcoder 智能代码补全引擎

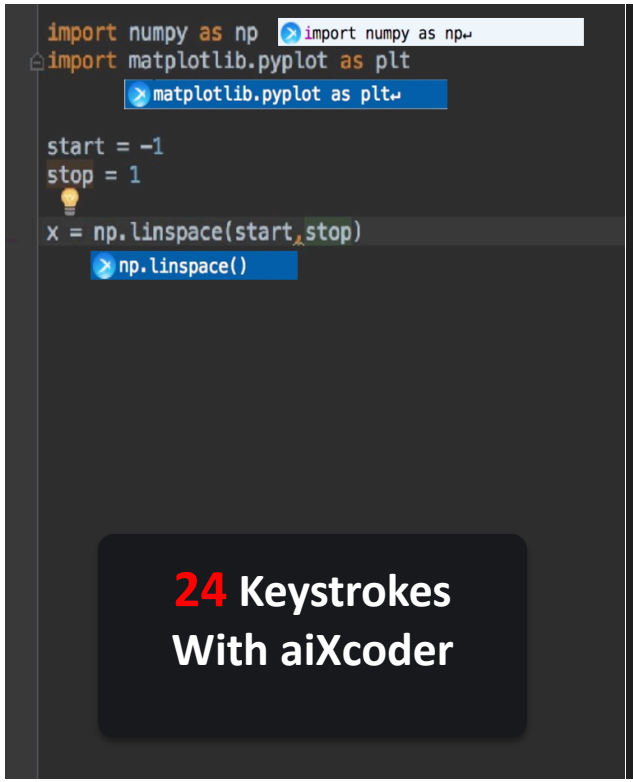


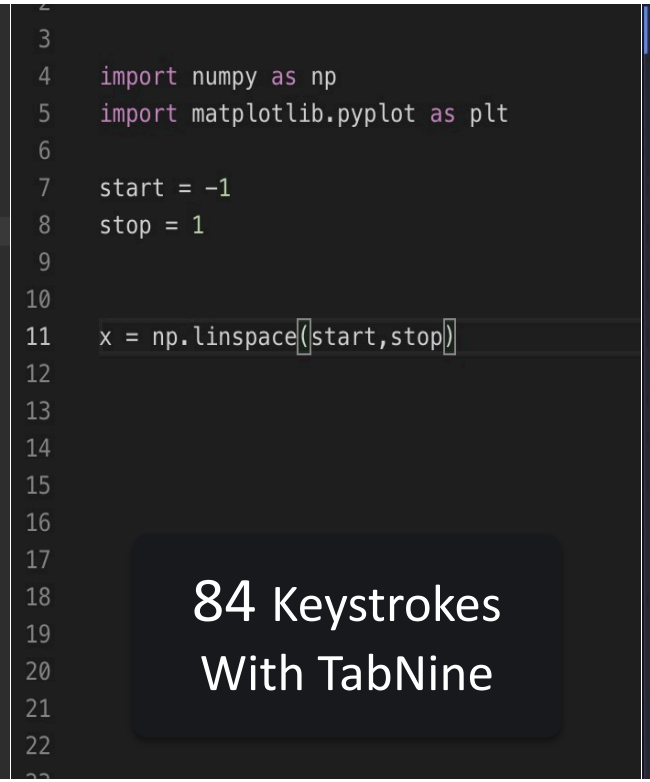
AIXcoder 智能代码搜索引擎

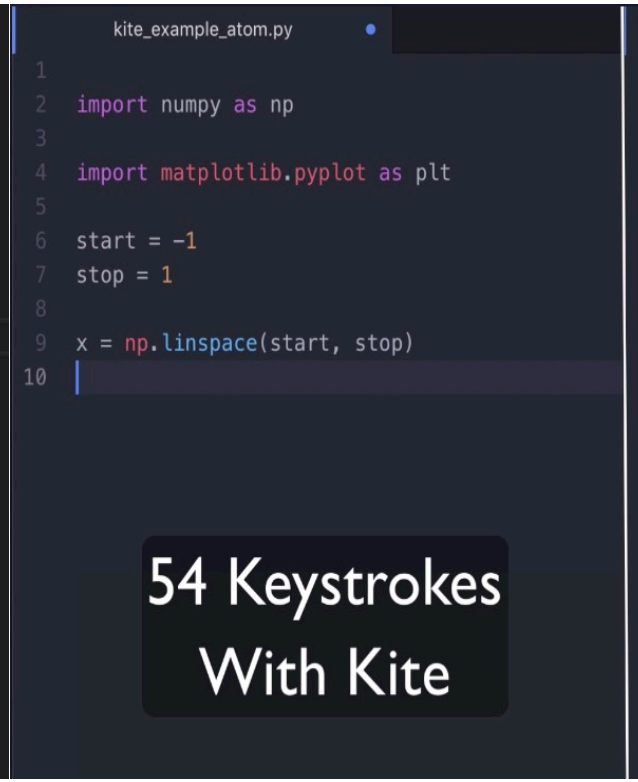


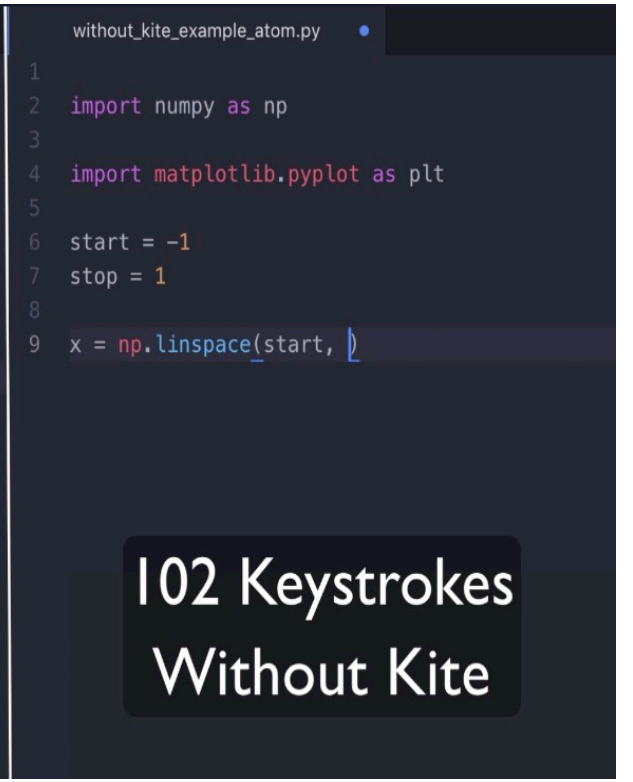
aiXcoder vs. Others

Source Code: https://www.kite.com/









aiXcoder 2.0

JET BRAINS — IDEs .NET Team Tools Dev Guide

Code tools

AiXcoder Code Completer

★★★★★

aiXcoder

Overview Versions Reviews

AiXcoder Code Completer & Code Search Engine

AiXcoder is a powerful code completer & code serach engine based on state-of-the-art deep learning technology. It has the potential of recommending you a full line of code, which will help you code faster. AiXcoder also provides a code search engine to help you search for API use cases on GitHub.

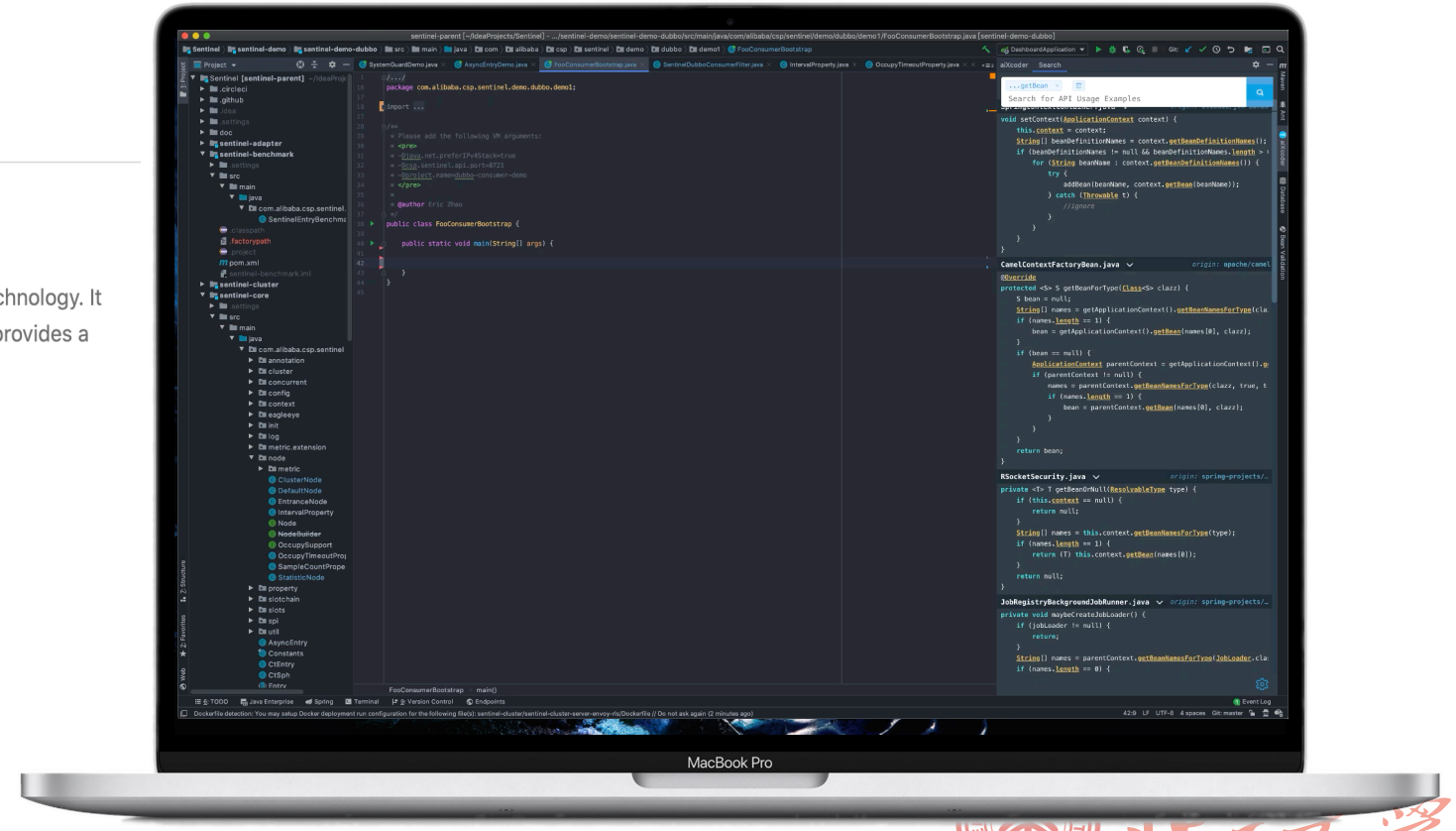
```
* @author Eric Zhao
*/
public class FooConsumerBootstrap {

    public static void main(String[] args) {

        AnnotationConfigApplicationContext consumerContext = new AnnotationConfigApplicationContext();
        consumerContext.register(ConsumerConfiguration.class);
        consumerContext.refresh();

        FooServiceConsumer service = consumerContext.getBean(FooServiceConsumer.class);

        for (int i = 0; i < 15; i++) {
            try {

            }
        }
```

more...

JetBrains Plugins Repository

aiXcoder

AiXcoder Code Completer IDE

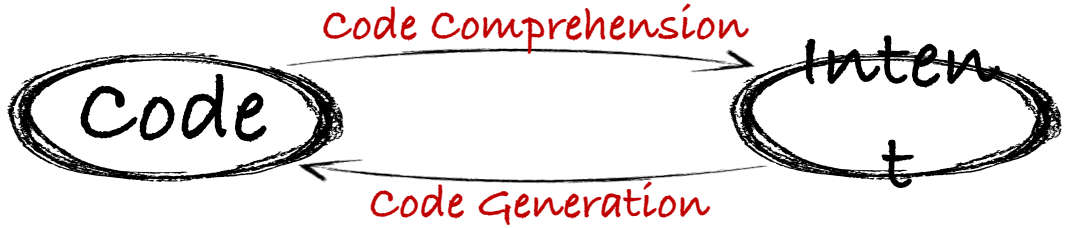


Agenda

- Can Machine Learn from Source Code, and Why
  - Be Inspired by Natural Language, naturalness
- How Machine learns from Source Code
  - Code/Program/Software Comprehension
  - Find the Features of Source Code
- Learn What
  - Learn Task Specific Knowledge and tacit knowledge
  - AiXCoder
- What Left and What are Next

Software Automation, Long Way to Go

- From finest grain to coarser and coarser, until be able to extract entity/concept level tokens
  - Character level
  - Identifier/keyword token level
  - Structure level
  - Execution flow level
  - ……




```
public static byte[] bitmapToByte(Bitmap b){
    ByteArrayOutputStream o = new ByteArrayOutputStream();
    b.compress(Bitmap.CompressFormat.PNG,100,o);
    return o.toByteArray();
}
```

DeepCom: convert Bitmap to byte array
Human-Written: convert Bitmap to byte array

```
private static void addDefaultProfile(SpringApplication app,
                SimpleCommandLinePropertySource source){
    if(!source.containsProperty("spring.profiles.active")
                &&!System.getenv().containsKey("SPRING_PR
                                        OFILES_ACTIVE")){
        app.setAdditionalProfiles
                (Constants.SPRING_PROFILE_DEVELOPMENT);
    }
}
```

DeepCom: If no profile has been configured , set by default the “dev” profile.
Human-Written: If no profile has been configured , set by default the “dev” profile.

```
public static void sort(Comparable[] a){
    int n=a.length;
    for (int i=1; i < n; i++){
        Comparable v=a[i];
        int lo=0, hi=i;
        while (lo < hi) { … }
        …
    }
    assert isSorted(a);
}
```

DeepCom: Sorts the array in ascending order, using the natural order.
Human-written: Rearranges the array in ascending order, using the natural order.

```
public boolean isEmpty(){
    return root == null;
}
```

DeepCom: Returns true if the symbol is empty.
Human-written: Is this symbol table empty?

```
public boolean contains(int key){
    return rank(key) != -1;
}
```

DeepCom: Checks whether the given object is contained within the given set.
Human-written: Is the key in this set of integers?

Still in Source Code Program Language Domain

nation, Long Way to Go

```
public static byte[] bitmapToByte(Bitmap b){
    ByteArrayOutputStream o = new ByteArrayOutput
    b.compress(Bitmap.CompressFormat.PNG,100,o);
    return o.toByteArray();
}
```

```
private static void addDefaultProfile(SpringApplication app,
                SimpleCommandLinePropertySource source){
    if(!source.containsProperty("spring.profiles.active")
                &&!System.getenv().containsKey("SPRING_PR
                                        OFILES_ACTIVE")){
        app.setAdditionalProfiles
                (Constants.SPRING_PROFILE_DEVELOPMENT);
    }
}
```

```
public static void sort(Comparable[] a){
    int n=a.length;
    for (int i=1; i < n; i++){
        Comparable v=a[i];
        int lo=0, hi=i;
        while (lo < hi) { ... }
        ...
    }
    assert isSorted(a);
}
```

```
public boolean isEmpty(){
    return root == null;
}
```

```
public boolean contains(int key){
    return rank(key) != -1;
}
```

DeepCom: ... map to byte array

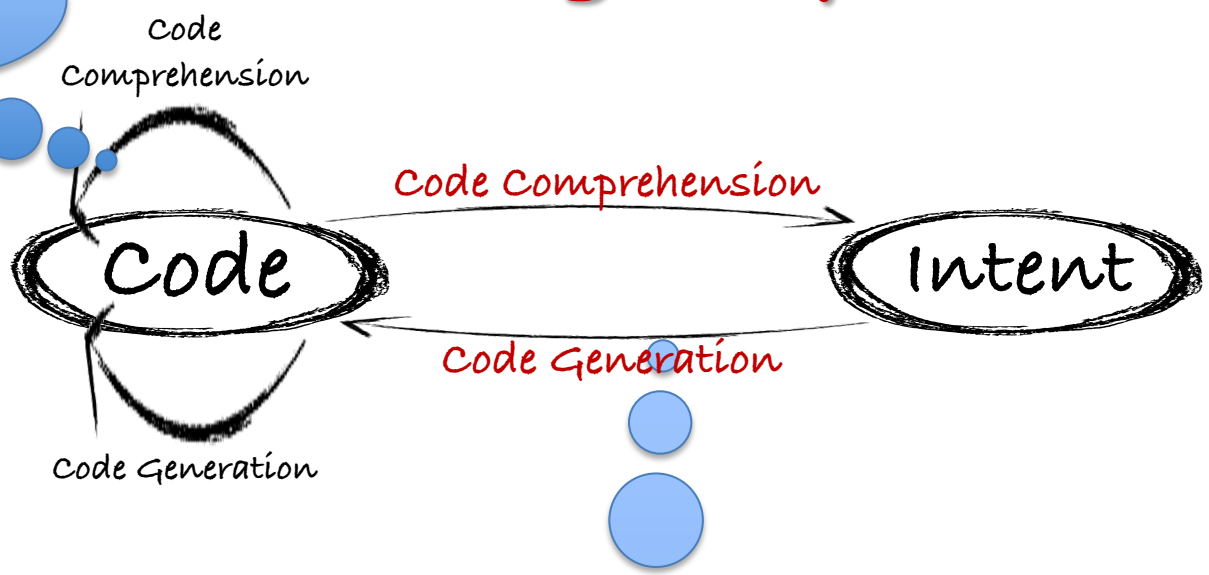
DeepCom: If no profile has been configured , set by default the “dev” profile.
Human-Written: If no profile has been configured , set by default the “dev” profile.

DeepCom: Sorts the array in ascending order, using the natural order.
Human-written: Rearranges the array in ascending order, using the natural order.

DeepCom: Returns true if the symbol is empty.
Human-written: Is this symbol table empty?

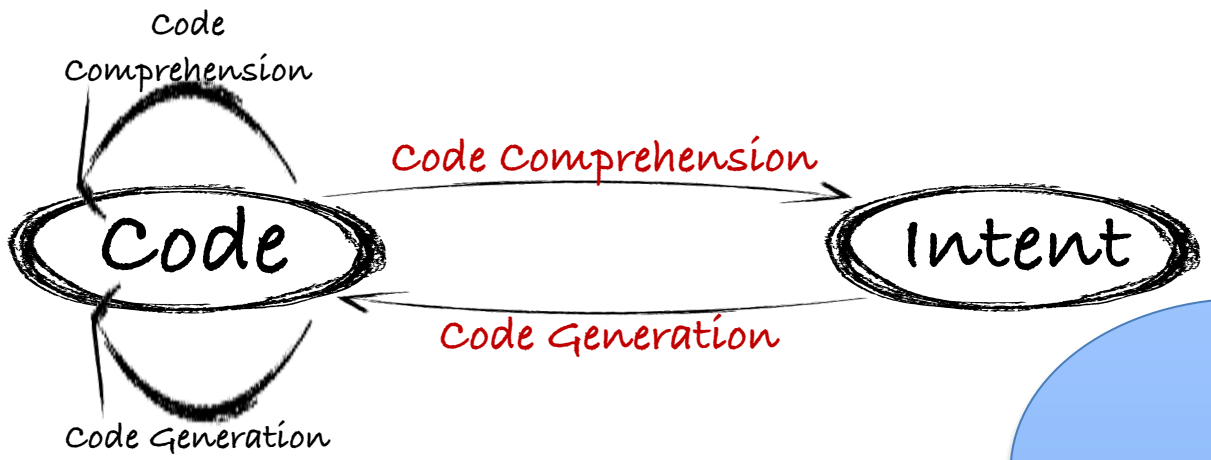
DeepCom: Checks whether the given object is contained within the given set.
Human-written: Is the key in this set of integers?

What are missing ?



Can we across the boundary and build the real link between the digital world and the business/real world ? If yes, how ?

Software Automation, Long Way to Go



From Intent to Code:
Specialization and Decompe

?

From Code to Intent:
Abstraction and Composition

From Button to Up
Character level
level

The Answers:
Domain Knowledge;
The Combination;
Open Assets: Source Code, Model, Design, ……;
……..

Thank you for your attention!