




Grading Computer Programs Using Machine Learning

Rohit Takhar

Aspiring Minds

Automatic grading of programs– Why?

- Will help professors and TAs save a lot of time.
- Companies can recruit efficiently
- **MOOCs** - need automated open response assessments to really make it effective. True scaling of such system currently not achieved.



Take great online courses from
the world's best universities



What does a grader look for?

OBJECTIVE

To print the pattern of integers

```
1
2 3
3 4 5
4 5 6 7
```

1. Are there loops? Are there print statements?
2. Is there a nested loop structure?
3. Is the conditional in the inner loop dependent on
 - a variable modified in the outer loop?
 - a variable used in the conditional of the outer loop?

Implementation

```
void print_1(int N){
    for(i=1; i<=N; i++){
        print newline;
        count = i;
        for(j=0; j<i; j++){
            print count;
            count++;
        }
    }
}
```

Grading programs

***Automata* – Automatic program evaluation engine**

Machine Learning based scoring engine

A model to predict the logical correctness of a program, given the control and data dependencies it possesses

Evaluation of programming best practices

Lint-styled rule-based system to detect programs not following programming best practices.

Asymptotic complexity evaluation

measures the run-time of the code for various input sizes and empirically derives the complexity

KDD 2014: A system to grade computer programming skills using machine learning
- Shashank Srikant, Varun Aggarwal

TARGET PROGRAM

```
void print(int N){  
  for(i =1 ; i<=N; i++){  
    print newline;  
    count = i;  
  
    for(j=0; j<i; j++)  
      print count; count++;  
  }  
}
```

CONTROL FEATURES – COUNTS

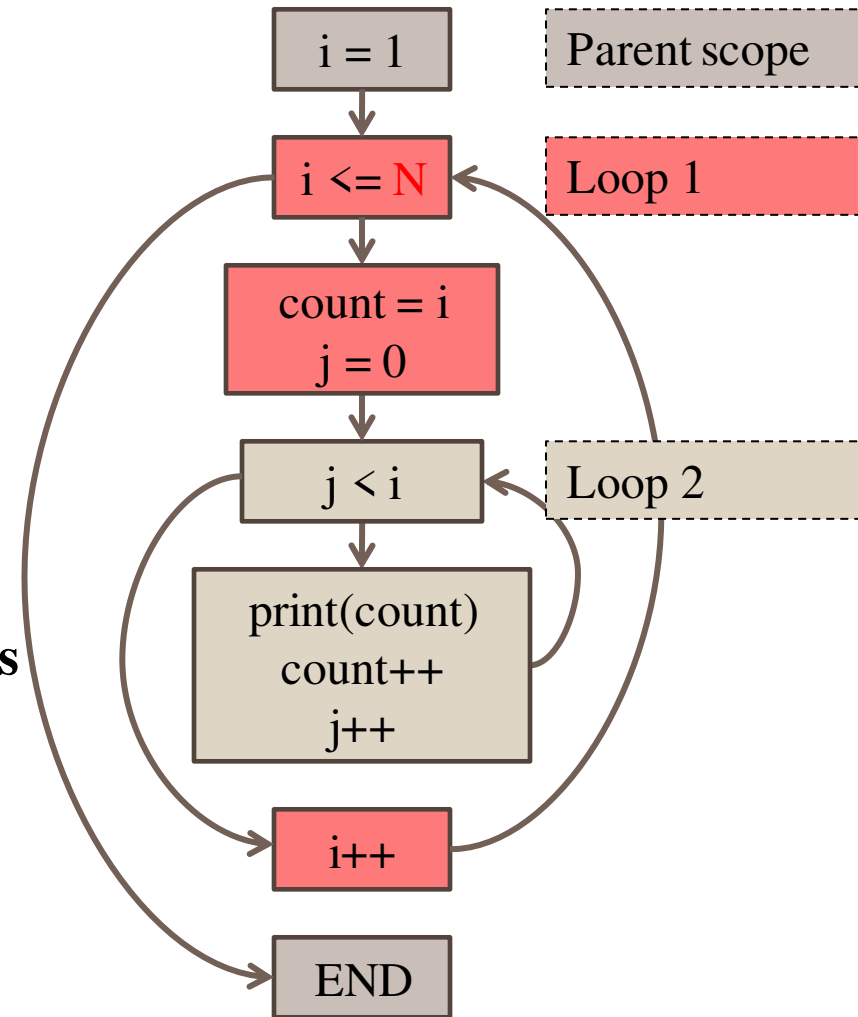
Counts of control-related keywords/tokens

E.g. count(for) = 2
count(for-in-for) = 1
count(while) = 0

Control-context of these keywords

- The Print command as loop(loop(print)))

CONTROL FLOW GRAPH



CONTROL-CONTEXT

Context of the control structure

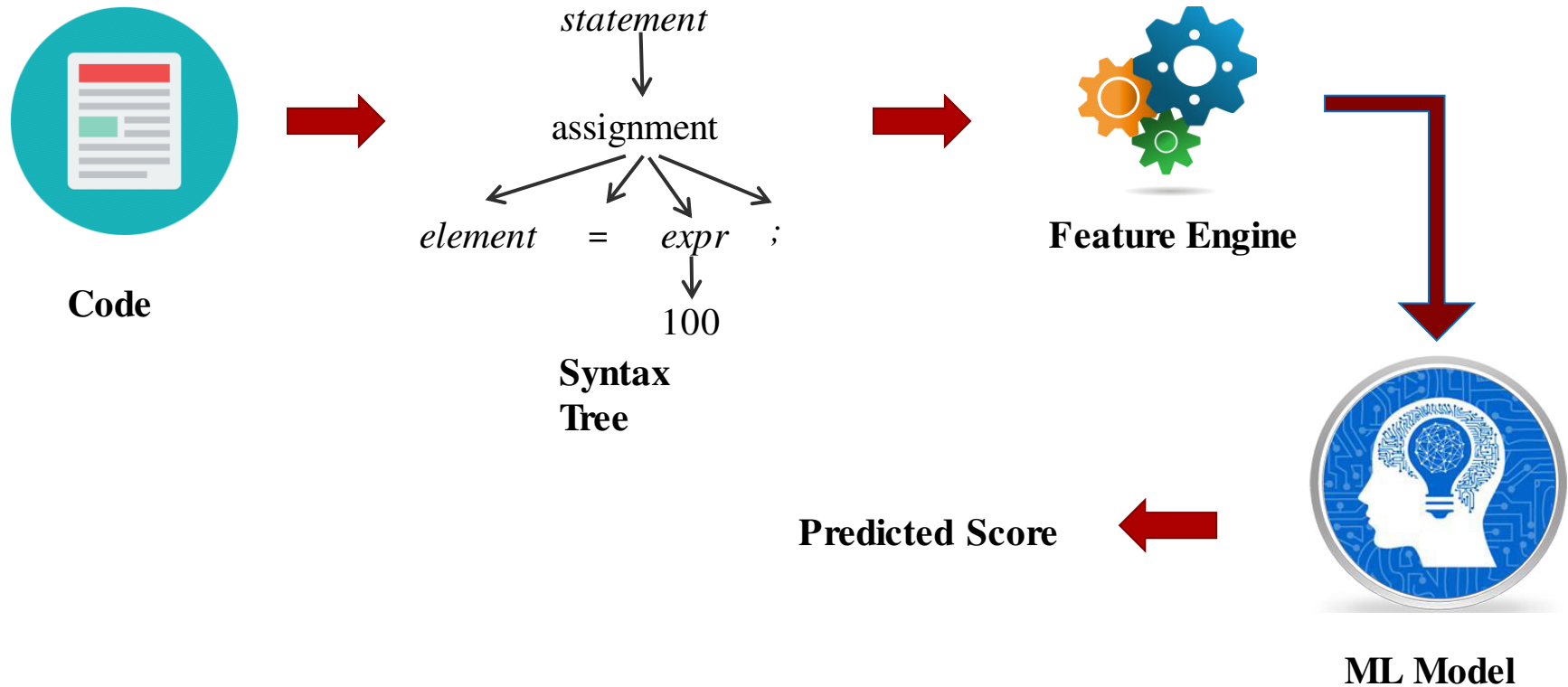
Counts of data-related tokens in context of the control structure

```
count(block1 :loop(loop_cond(<))) = 1
```

- i++ \rightarrow j < i	: var (i) related to var (j)	: appearing in a loop(loop_cond)
	previously incremented	: appearing in a loop

The relation and the increment happen in the same block

Flow diagram for compilable programs



Results: Question Specific Models

PROBLEM	# of features	Cross-val correl	Train correl	Validation correl	Test Case Score
1	80	0.61	0.85	0.79	0.54
2	68	0.77	0.93	0.91	0.80
3	193	0.91	0.98	0.90	0.64
4	66	0.90	0.94	0.90	0.80
5	87	0.81	0.92	0.84	0.84

Validation correlation \geq **0.79**

Matches inter-rater correlation between two human raters
(Published at KDD 2014)

ML Models are question specific

Binary search

```
int binarySearch(params)
{
    // Initializations
    for(;first <= last;) {
        if(array[middle] < search)
            // Look in the 2nd half
            first = middle + 1;
        else
            // Look in the first half
            last = middle - 1;
        middle = (first + last)/2;
    }
}
```

if-in-for

Bubble sort

```
int bubbleSort(params)
{
    for (i=0 ; i<n; i++)
    {
        for (j=0 ; j <n-i; j++)
        {
            if (array[j] > array[j+1])
            {
                // Swap elements
            }
        }
    }
}
```

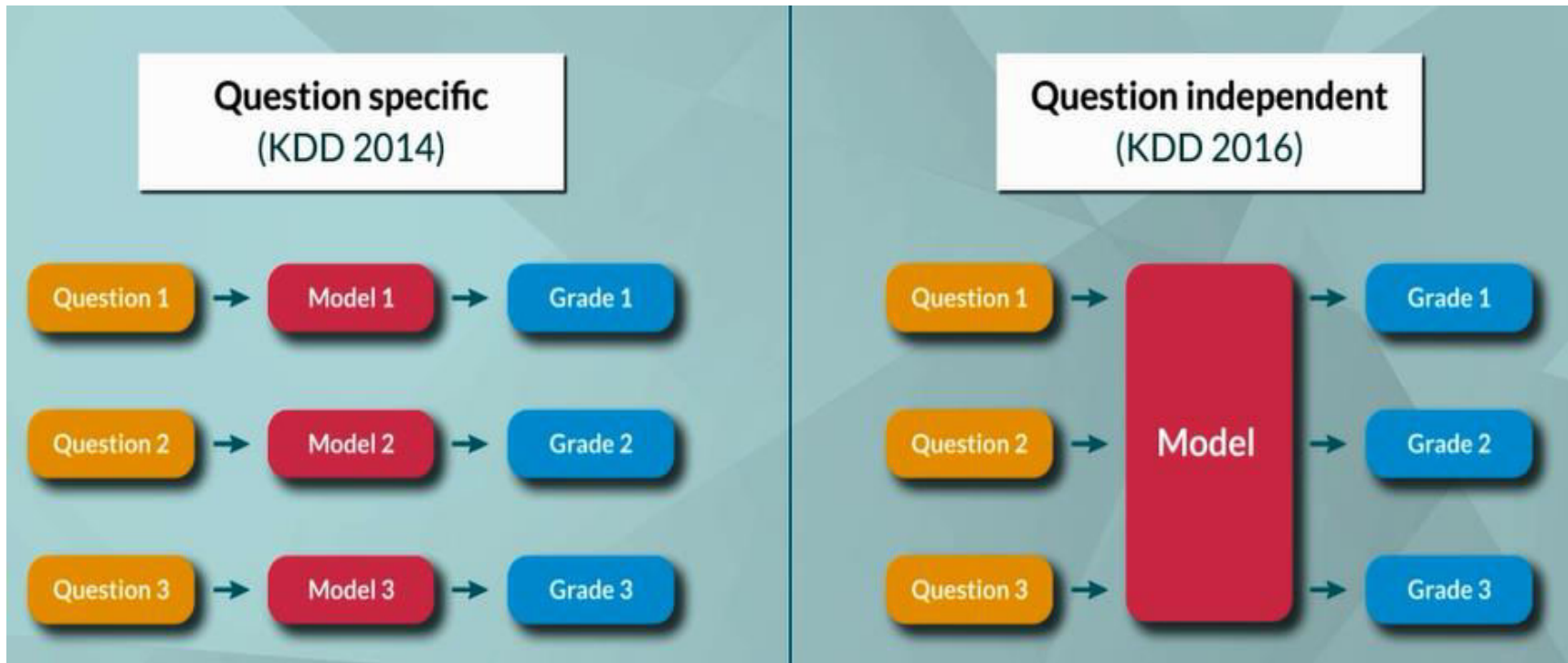
for-in-for

Solving for the industry needs scale

- The solution doesn't scale!!!
- We have 500+ questions in our database and support 35+ languages

How to get question independent models?

Question independent ML models



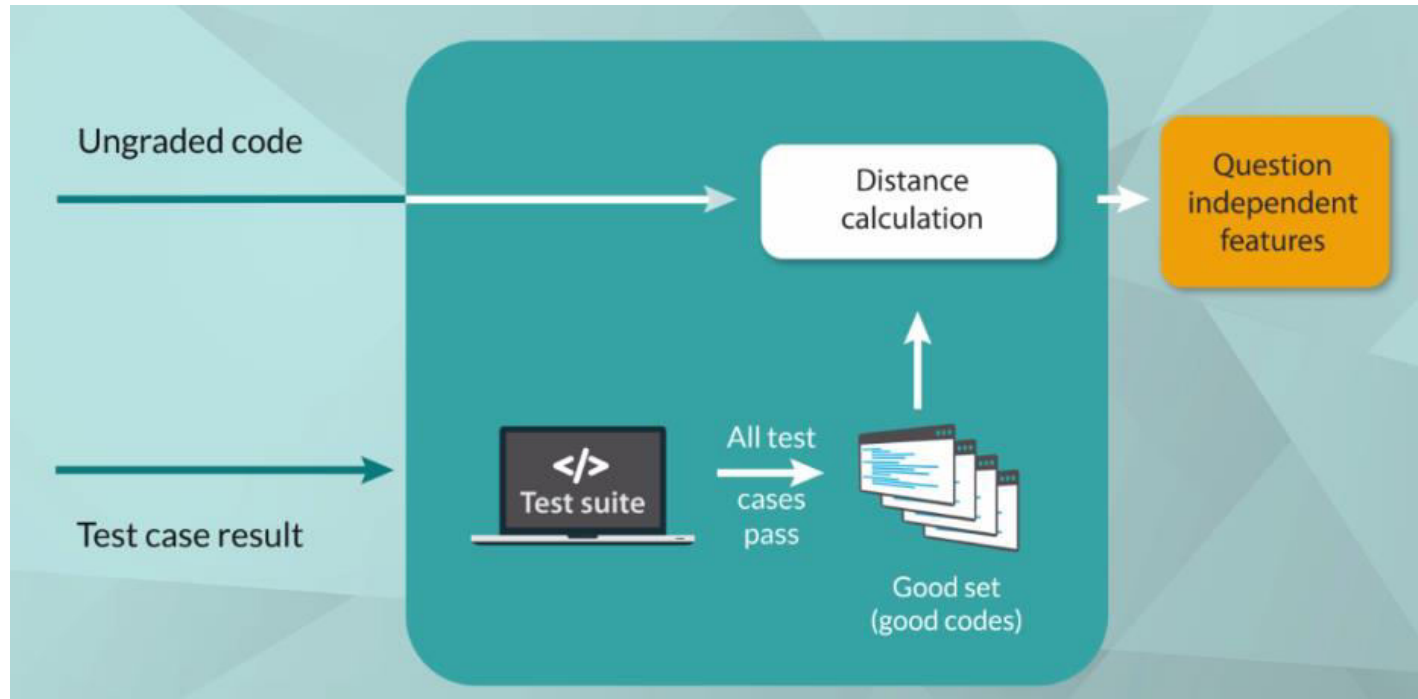
Two main ideas

Feature transformation - Convert original features into 'structurally invariant' features

How?

Exploit Automatic identification of good responses –Test cases can tell us this.

How does it work?



We just need a set of good codes for a new question, no labeled sample....

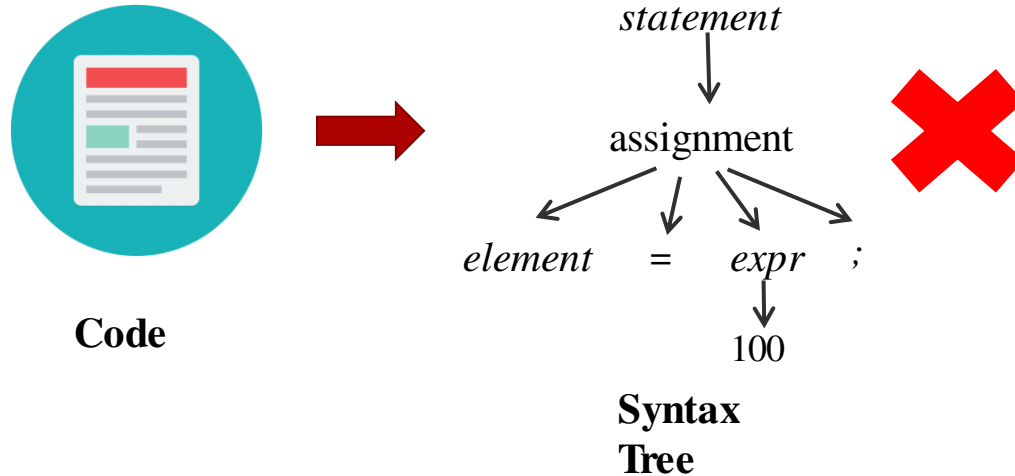
Results

Ques Set	Metric	QuesSpec	QuesIndep-N1	QuesIndep-N0	Baseline-TC
All questions	Correl	0.84	0.8	0.76	0.65
	Bias	0.14	0.24	0.28	0.35
	MAE	0.41	0.58	0.66	0.85
Unseen questions	Correl	0.85	0.8	0.76	0.65
	Bias	0.14	0.27	0.34	0.31
	MAE	0.43	0.62	0.7	0.84

- Question-independent vs test-case baseline
- Question-independent vs question-specific
- Performance on unseen problems
- Correl – Pearson coefficient (r)
- Bias – Average ($y_e - y_p$)
- MAE – Average ($|y_e - y_p|$)

(Published at KDD 2016)

For programs with syntax errors



- **26% more candidates were shortlisted for interviews**
- **19% more were hired**

(Published at IAAI 2019)

Various AI-led products

- Automatic Grading of essays
- Automatic Grading of emails
- A tab based motor skills test
- Simulated Chat customers

Lessons

- Problems are multi-disciplinary – spawn fields
- Problems need novel theoretically plausible features
- Small data-sets; Expensive to get labels; Need to create good balanced data sets
- Need generalized solutions for problem space which scales
- Format assumptions of solution may break – need to handle by novel means

Questions?