Teaching the Art of Computer Programming at a Distance by Generating Dialogues using Deep Neural Networks

Yijun Yu\(^1\), Anton Dil\(^1\), Irum Rauf\(^1\)
\(^1\)The Open University, UK

Xiaozhu Wang\(^2\)
\(^2\)The Open University of China

November 6, 2019 @ Convention Centre Dublin, Ireland
28\(^{th}\) ICDE - World Conference on Online Learning
The Art of Computer Programming
Finding an error in a Knuth text.
Stupidity: Cashing that $2.56 check you got.
Algorithms

Program S (Straight insertion sort). The records to be sorted are in locations INPUT+1 through INPUT+N; they are sorted in place in the same area, on a full-word key. rI1 ≡ j − N; rI2 ≡ i; rA ≡ R ≡ K; assume that N ≥ 2.

01 START ENT1 2-N 1                S1. Loop on j. j ← 2.
02  2H LDA INPUT+N,1 N−1          S2. Set up i, K, R.
03       ENT2 N−1,1 N−1          i ← j − 1.
04  3H CMPA INPUT,2 B + N−1−A S3. Compare K : K_i.
05       JGE 5F B + N−1−A       To S5 if K ≥ K_i.
06  4H LDX INPUT,2 B         S4. Move R_i, decrease i.
07       STX INPUT+1,2 B          R_{i+1} ← R_i.
08       DEC2 1 B         i ← i − 1.
09       J2P 3B B         To S3 if i > 0.
10  5H STA INPUT+1,2 N−1 S5. R into R_{i+1}.
11       INC1 1 N−1
12       J1NP 2B N−1 2 ≤ j ≤ N.
Teaching Java at a distance
Pedagogy of The Open University module M250

• 2\textsuperscript{nd} year flagship undergraduate module at the School of Computing and Communications

• 1400 students per presentation

• Unique challenges and traditional distance teaching approaches
  • Recruit at least 1 associated lecturer per 20 students
  • Tutoring is mostly done through online materials, but also through face to face consultations
  • Tutor Marked Assignments (TMAs) and close book Exams
  • Interactive Computer Marked Assignments (iCMA)
An algorithm in Java

```java
public static void insertionSort()
{
    int i = 1;

    while
    {
        (i < arr.length)
        {
            int j = i;
            while
            {
                (j > 0)
                {
                    if (arr[j].key < arr[j-1].key)
                    {
                        InsertionSortNode tmp = arr[j];
                        arr[j] = arr[j-1];
                        arr[j-1] = tmp;
                    }
                    j = j - 1;
                }
            }
            i = i + 1;
        }
    }
}
The needs for AI

• Neural Networks
  • Image classification (convolutional)
  • Recurrent NN
  • Reinforced Learning
  • ...

• Unique challenges
  • Unlike natural languages, algorithms are structured
  • Algorithms are agnostic to programming languages
  • They are more germane to the machine, i.e., assembler
  • Students still need to learn in a natural way
Overview: Dialogues with Echo

[Diagram showing the process of parsing a Java program, creating an abstract syntax tree, and interacting with a voice assistant to answer questions and provide context.]
Knowledge Representation for Code

• What are the key features of an algorithm
  • Abstract Syntax Trees
  • Dependence Graphs
  • Def-Use chains

• What are not key features
  • Exact names of variables/methods

• Suitable representations
  • Tree-based Convolution Neural Networks [AAAI’16]
  • Gated Graph Neural Networks [SANER’19]
  • Tree Capsules Networks [FSE’19]
The Machine Learning Process

Programmer

1. \text{algo}\left(\text{ast2vec}\left(\text{parse}(\text{lang, code})\right)\right)\
2. \text{algo}\left(\text{ast2vec}\left(\text{parse}(\text{lang1, code1})\right)\right) = \text{algo}\left(\text{ast2vec}\left(\text{parse}(\text{lang2, code2})\right)\right)\

Repositories

\{\text{algoName}, \text{lang.}, \text{code}\}\}

Parsers

AST = parse(\text{lang.}, \text{code})

Bi-TBCNNs

name = algo(vec)

vec = ast2vec(AST)

AST2vec

https://github.com/yijunyu/bi-tbcnn
Algorithm Classification
Probability of Correct Prediction

Source code editor
(insertion sort)

Visual Explanation Of Classification Results

Pick a Program

https://github.com/yijunyu/demo
Conclusions and Future work

• Summary
  • With student’s *unstructured* inputs and the underlying *structured* knowledge representations, NNs are effective in generating dialogues with *unstructured* outputs.
  • Dialogues can take different forms:
    • An Interactive Visual Programming Environment [ICSE’19]
    • Structured Arguments [JSS’15]
    • Chatbots [EPSRC Johnny project]

• Future work
  • iCMA replacement with smarter feedback (M250)
  • Evaluation with the examinable components, i.e. automated marking, using Generative Adversarial Networks [FSE’19]
References


