I'm in West Virginia and Kentucky this weekend as part of my Year of Travel challenge.

I talked to some students who were using the Summit personalized learning tools we've been building at the Chan Zuckerberg Initiative and who were learning how to code. These kids were showing me the games, robots, drones, and VR apps (!!!) they were coding.

They told me they were learning much faster with per... See More
Create new kinds of software tools that leverage massive codebases to solve problems beyond what is possible with traditional techniques.

number of repositories

last 8 years

15 million repositories

Billions of lines of code

High quality, tested, maintained programs
Statistical Software Tools

**Writing Code**
Code Completion

```java
Camera camera = Camera.open();
camera.SetDisplayOrientation(90);
```

**Porting Code**
Programming Language Translation

**Program Analysis**
Points-to/Type Analysis

```javascript
function collect(val, idx, obj) {
    if (val >= this.threshold) { ... }
}
dat.filter( collect, ctx );
```

**Testing/Debugging**
Statistical Bug Detection

```python
... for x in range(a):
    print a[x]
```

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques.
Programming Languages

- Program Syntax
- Program Semantics
- Program Synthesis
- Program Representation
Programming Languages + Machine Learning

- Program Syntax
- Program Semantics
- Program Synthesis
- Program Representation
- Precision vs Scalability
- Probabilistic Models
- Explainability
- Learning/Inference Algorithms
Probabilistic Model for Code

Existing Programs → Learning → Model

Probabilistic Model
Probabilistic Model for Code

Existing Programs

Learning

Model

Probabilistic Model

function \texttt{area}(a) \{
    \texttt{return a.width * a.width}
\}

\textbf{Goal:} Assign probability to a program
<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last two tokens, Hindle et. al. [ICSE’12]</td>
<td>22.2%</td>
</tr>
<tr>
<td>Last two APIs, Raychev et. al. [PLDI’14]</td>
<td>30.4%</td>
</tr>
</tbody>
</table>

is this the best we can do?
<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last two tokens, Hindle et. al. [ICSE’12]</td>
<td>22.2%</td>
</tr>
<tr>
<td>Last two APIs, Raychev et. al. [PLDI’14]</td>
<td>30.4%</td>
</tr>
<tr>
<td>Last three APIs</td>
<td></td>
</tr>
<tr>
<td>Declaration Site + Last two APIs</td>
<td></td>
</tr>
<tr>
<td>Variable Name + Method Name + Last API</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
## JavaScript APIs

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last two tokens, Hindle et. al. [ICSE’12]</td>
<td>22.2%</td>
</tr>
<tr>
<td>Last two APIs, Raychev et. al. [PLDI’14]</td>
<td>30.4%</td>
</tr>
</tbody>
</table>

**Program synthesis** 66.4%
Model Requirements

Existing Programs

Learning

Model

- Widely Applicable
- Efficient Learning
- High Precision
- Explainable Predictions
Program Synthesis

 Synthesize a function $f$ from a domain specific language that explains the data
Function Examples

\[ f(p_1) = \begin{cases} \text{for, } & \text{if, } \\ \text{length} = = 0 \end{cases} \]

\[ f(p_2) = \begin{cases} \text{notify, } & \text{position, } \\ \text{hide} \end{cases} \]

```javascript
for (j = 0; j < groups.length; j++) {
    idsInGroup = groups[j].filter(
        function(id) { return id >= 42; }
    );
    if (idsInGroup.length == 0) {
        elem.notify(..., {
            position: 'top',
            hide: false,
        });
    }
}
```
Function Representation

In general:
Unrestricted programs (Turing complete)

Our Work:
TCond Language for navigating over trees
and accumulating context

\[
\begin{align*}
\text{TCond} & ::= \varepsilon \mid \text{WriteOp} \ T\text{Cond} \mid \text{MoveOp} \ T\text{Cond} \mid \text{BranchProg} \\
\text{BranchProg} & ::= \textbf{if} \ \text{pred}(x) \ \textbf{then} \ T\text{Cond} \ \textbf{else} \ T\text{Cond} \\
\text{MoveOp} & ::= \text{Up}, \ \text{Left}, \ \text{Right}, \ \text{DownFirst}, \ \text{DownLast}, \\
& \quad \text{NextDFS}, \ \text{PrevDFS}, \ \text{NextLeaf}, \ \text{PrevLeaf}, \ \text{PrevNodeType}, \ \text{PrevNodeValue}, \ \text{PrevNodeContext} \\
\text{WriteOp} & ::= \text{WriteValue}, \ \text{WriteType}, \ \text{WritePos}
\end{align*}
\]
Expressing Functions: TCond Language

TCond ::= $\varepsilon$ | WriteOp TCond | MoveOp TCond | BranchProg

BranchProg ::= if pred(x) then TCond else TCond

MoveOp ::= Up, Left, Right, DownFirst, DownLast, NextDFS, PrevDFS, NextLeaf, PrevLeaf, PrevNodeType, PrevNodeValue, PrevNodeContext

WriteOp ::= WriteValue, WriteType, WritePos
elem.notify(
    ..., 
    ..., 
    {
        position: 'top',
        hide: false,
        ...
    }
);
Example

elem.notify(
    ... ,
    ... ,
    {
        position: 'top',
        hide: false,
        ...
    }
);
elem.notify(
    ..., 
    ..., 
    {
        position: 'top',
        hide: false,
    },
);
Example

Query

```javascript
elem.notify(
  ...
, ...
, {
    position: 'top',
    hide: false,
    ?
  }
);
```

TCond

```
γ

Left
WriteValue
Up
WritePos
Up
DownFirst
DownLast
WriteValue
```

```
{}

{hide}
{hide}
{hide, 3}
{hide, 3}
{hide, 3}
{hide, 3}
{hide, 3, notify}
```
Example

Query

elem.notify(
  ...
  ...
  {
    position: 'top',
    hide: false,
    ?
  }
);

TCond

Left
WriteValue
Up
WritePos
Up
DownFirst
DownLast
WriteValue

γ

{}  
{hide}  
{hide}  
{hide, 3}  
{hide, 3}  
{hide, 3}  
{hide, 3}  
{hide, 3, notify}

↓

{ Previous Property, Parameter Position, API name }
# JavaScript 150k Dataset
(Source Code in AST Format)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>44.2%</td>
</tr>
<tr>
<td>Probabilistic Context-Free Grammars (PCFG)</td>
<td>51.1%</td>
</tr>
<tr>
<td>SVM</td>
<td>70.5%</td>
</tr>
<tr>
<td>N-gram</td>
<td>71.3%</td>
</tr>
</tbody>
</table>

*Program Synthesis* 81.5%
<table>
<thead>
<tr>
<th>Model</th>
<th>Error Rate</th>
<th>Training</th>
<th>Queries/s</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>38.1%</td>
<td>~80 Hrs</td>
<td>300</td>
<td>53 MB</td>
</tr>
<tr>
<td>n-gram</td>
<td>35.9%</td>
<td>4 Sec</td>
<td>41000</td>
<td>24 MB</td>
</tr>
<tr>
<td>Synthesis</td>
<td>31.4%</td>
<td>8 Hrs</td>
<td>28000</td>
<td>19 MB</td>
</tr>
</tbody>
</table>
## Hutter Prize Wikipedia (Natural Language + Metadata)

<table>
<thead>
<tr>
<th>Model</th>
<th>Bits per Character</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>N-gram</strong></td>
<td>1.94</td>
</tr>
<tr>
<td><strong>Program Synthesis</strong></td>
<td>1.67</td>
</tr>
<tr>
<td>Stacked LSTM [Graves et. al. 2013]</td>
<td>1.62</td>
</tr>
<tr>
<td>MRNN [Sutskever et.al. 2011]</td>
<td>1.60</td>
</tr>
<tr>
<td>MI-LSTM [We et.al. 2016]</td>
<td>1.44</td>
</tr>
<tr>
<td>HM-LSTM [Chung et. al. 2017]</td>
<td>1.34</td>
</tr>
</tbody>
</table>
Work @ ETH Zurich

Prof. Martin Vechev
Veselin Raychev
Pavol Bielik
Christine Zeller
Pascal Roos
Benjamin Bischel
Svetoslav Karaivanov
Benjamin Mularczyk
Prabhakaran Santhanam
Pavle Đorđević
Learning-Based Probabilistic Programming Tools

Writing Code
Code Completion

Camera camera = Camera.open();
camera.SetDisplayOrientation(90);

Porting Code
Programming Language Translation

Camera camera = Camera.open();
camera.SetDisplayOrientation(90);

Testing/Debugging
Statistical Bug Detection

function collect(val, idx, obj) {
  if (val >= this.threshold) {
    ...
  }
}
dat.filter( collect, ctx );

Program Analysis
Points-to/Type Analysis

... for x in range(a):
  print a[x]

likely error

http://plml.ethz.ch/