goto; copenhagen

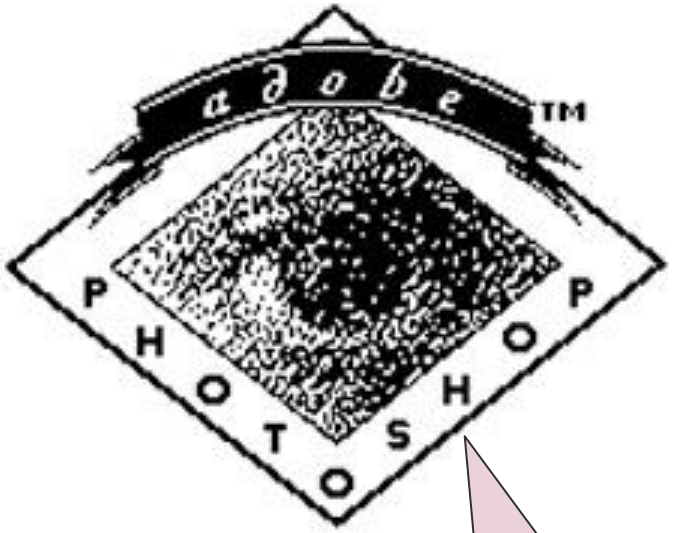
# GOTO Copenhagen 2018

Conference Nov. 19 - 21


**Click 'Rate Session' to rate session and ask questions.**

Follow us @gotocph

“Software is eating the world”




# Adobe Photoshop™

*Macintosh version 1.0.7*

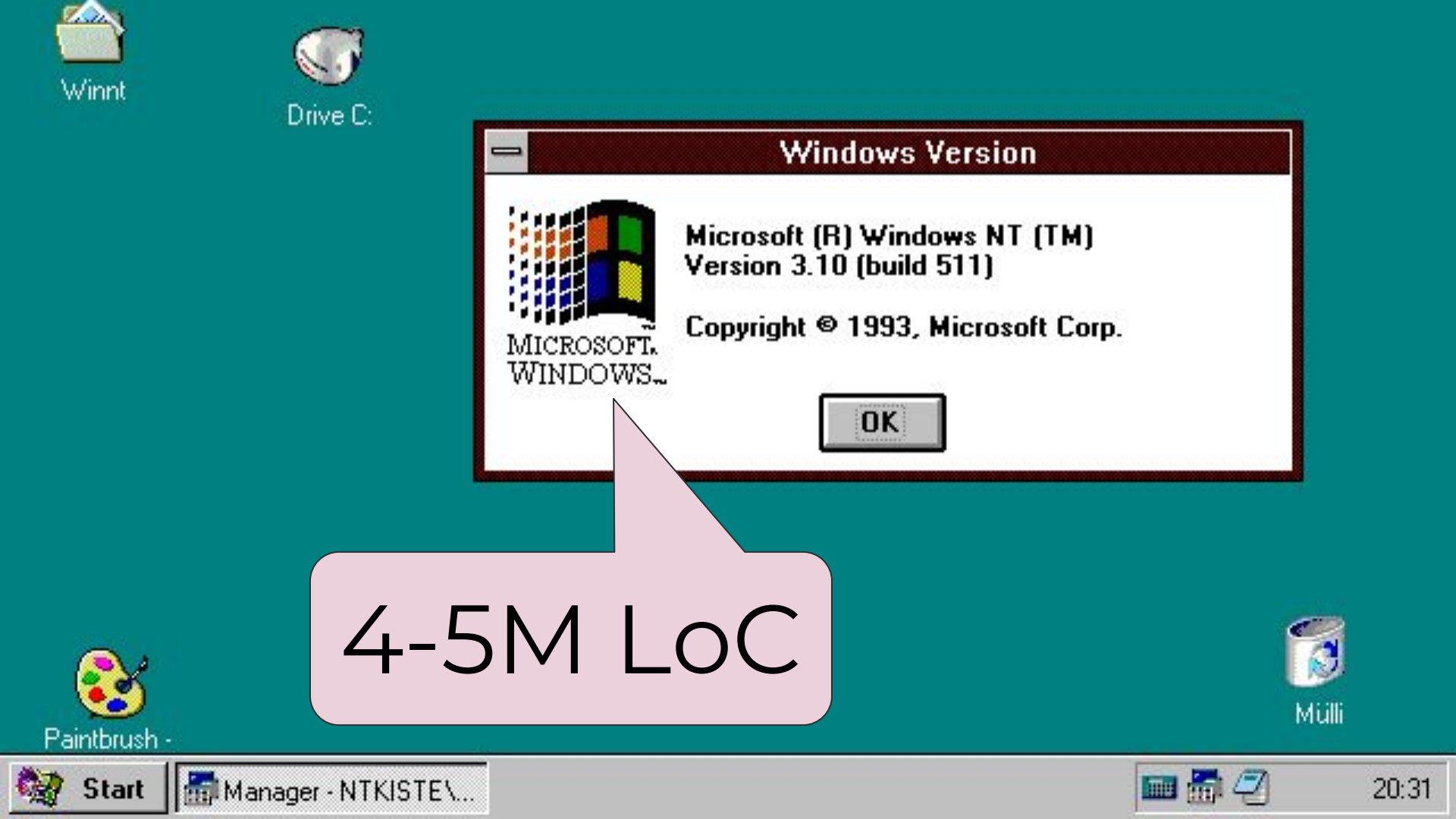
Thomas Knoll, John Knoll, Steve Guttman
and Russell Brown



Personalized for:



128k LoC


Winnt
Drive C:
Windows Version
Microsoft (R) Windows NT (TM)
Version 3.10 (build 511)
Copyright © 1993, Microsoft Corp.
MICROSOFT
WINDOWS
OK
4-5M LoC
Paintbrush -
Mülli
Start
Manager - NTKISTE\...
20:31

Apache OpenOffice™

9M LoC

18M LoC


45M LoC


150M LoC


1 billion lines of code

Google Search I'm Feeling Lucky

CRAFTSMAN
NICKEL PLATED

# Machine Learning on Source Code

Francesc Campoy

# Francesc Campoy

VP of Product & DevRel

source{d}

Machine Learning for Large Scale Code Analysis

**@francesc | #MLonCode**

# Agenda

- Machine Learning on Source  Code
- Research
- Use Cases
- The Future

# Machine Learning on Source Code

# Machine Learning on Source Code

Field of Machine Learning where the input data is source code.

MLonCode

# Machine Learning on Source Code

## Related Fields:

- Data Mining
- Natural Language Processing
- Graph Based Machine Learning

## Requires:

- Lots of data
- Really, lots and lots of data
- Fancy ML Algorithms
- A little bit of luck

# Challenge #1

## Data Retrieval

# The datasets of ML on Code

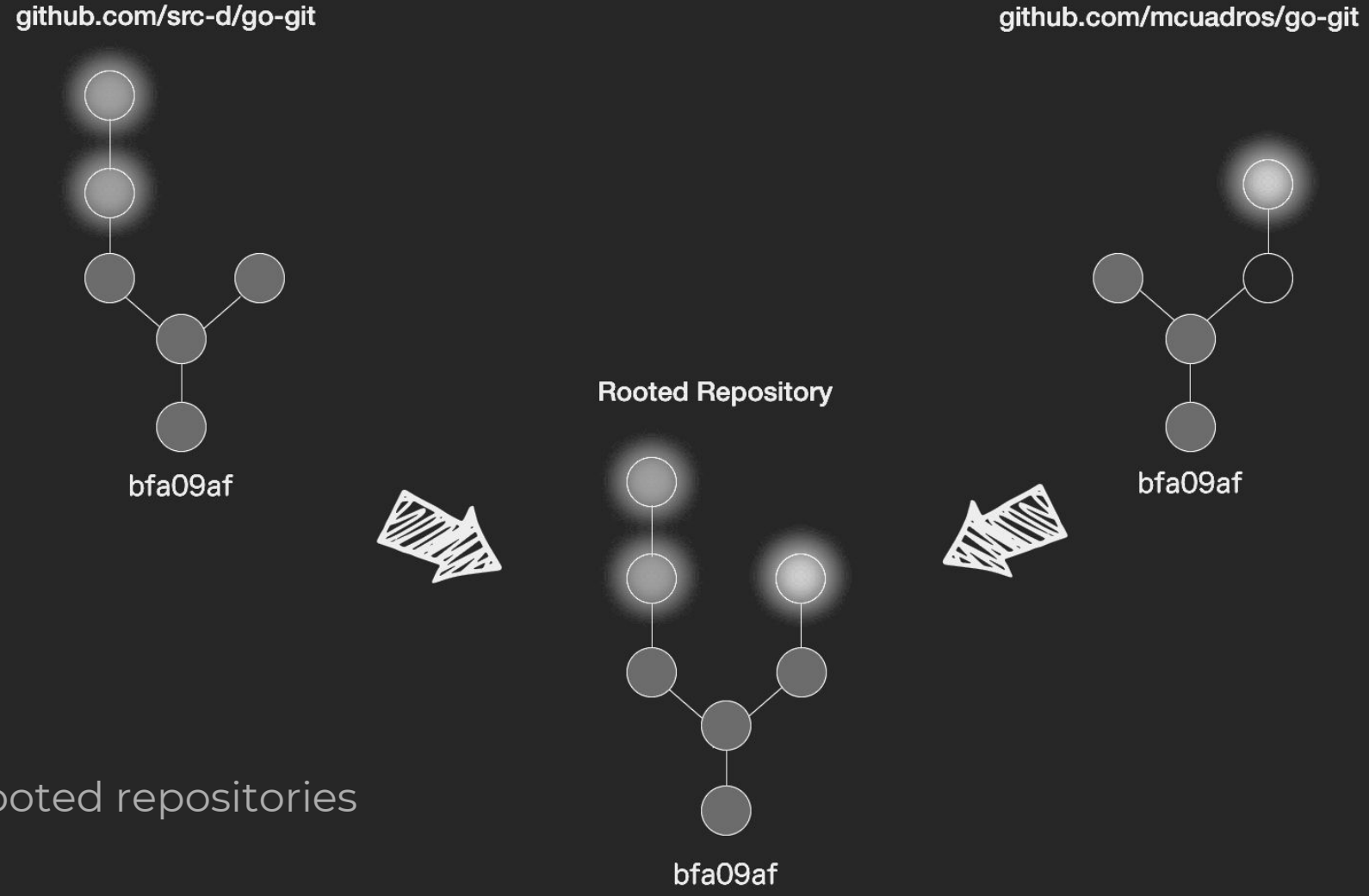
- GH Archive: https://www.gharchive.org


- Public Git Archive https://pga.sourced.tech

# Public Git Archive

Announcement: [blog.sourced.tech/post/announcing-pga](blog.sourced.tech/post/announcing-pga)

| _ | Qualitas Corpus | Sourcerer | GitHub Java Corpus | Public Git Archive |
|---|---|---|---|---|
| N projects | 111 | 19,233 | 14,807 | 182,014 |
| Year of release | 2013 | 2014 | 2013 | 2018 |
| Languages | 1 (Java) | 1 (Java) | 1 (Java) | 455 |
| Repeatable | No | No | No | Yes |
| N files | 177k | 1.9M | 1.5M | 54.5M (HEAD) |
| Lines of Code | 37M | 320M | 352M | 15,941M (HEAD) |
| Storage Size | 1.3GB | 19GB | 14GB | 3.0TB |




Rooted repositories

# the public git archive story

By **Vadim Markovtsev** 20 November 2018

## what is public git archive?

Public Git Archive is the result of months of efforts curating a dataset suitable for training Machine Learning on Source Code (aka MLonCode) models. It contains 182,000 top-starred repositories on GitHub and takes 3 TB on disk. The repositories were cloned in February-March 2018. Check out the announcement post for more information. You should check out Engine which allows to run SQL queries on top the PGA and do other cool things.

## origins

blog.sourced.tech/post/pga_history/

we are able to clone since the beginning of source{d}. Back in

gle Cloud Storage and the metadata in MongoDB. It worked

# Challenge #2

# Data Analysis

# What is Source Code

```
package main

import “fmt”

func main() {
    fmt.Println(“Hello, Copenhagen”)
}
```

```
'112', '97', '99', '107', '97', '103',
'101', '32', '109', '97', '105', '110',
'10', '10', '105', '109', '112', '111',
'114', '116', '32', '40', '10', '9',
'34', '102', '109', '116', '34', '10',
'41', '10', '10', '102', '117', '110',
'99', '32', '109', '97', '105', '110',
'40', '41', '32', '123', '10', '9',
'102', '109', '116', '46', '80', '114',
'105', '110', '116', '108', '110', '40',
'34', '72', '101', '108', '108', '111',
'44', '32', '112', '108', '97', '121',
'103', '114', '111', '117', '110', '100',
'34', '41', '10', '125', '10'
```

# What is Source Code

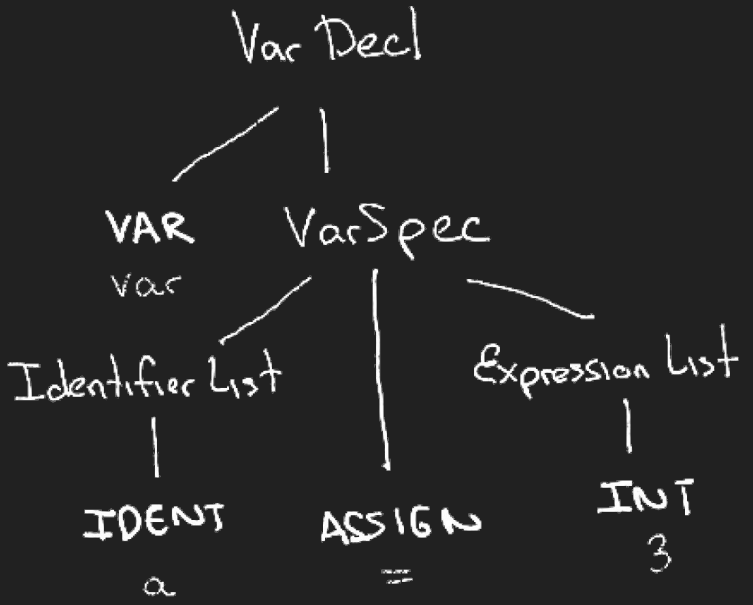
```
package main

import “fmt”

func main() {
    fmt.Println(“Hello, Copenhagen”)
}
```

```
package package
IDENT main
;

import import
STRING "fmt"
;

func func
IDENT main
(
)
```

```
{
IDENT fmt
.
IDENT Println
(
STRING "Hello, Denver"
)
;

}
;
```

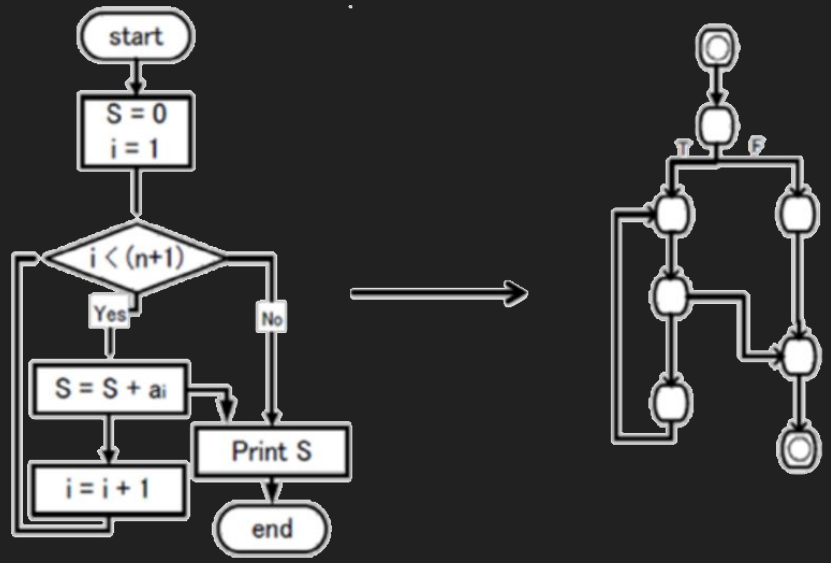
# What is Source Code

```
package main

import “fmt”

func main() {
    fmt.Println(“Hello, Copenhagen”)
}
```



# What is Source Code

```
package main

import “fmt”

func main() {
    fmt.Println(“Hello, Copenhagen”)
}
```



# What is Source Code

- A **sequence** of bytes
- A **sequence** of tokens
- An abstract syntax **tree**
- A **graph** (e.g. Control Flow Graph)

# Analyzing Code

## Tasks

- Language Classification
- File Parsing
- Token  Extraction
- History Analysis
- Reference Resolution

## Tools

- enry
- babelfish
- libuast & XPath selectors
- go-git
- kythe.io

source{d} engine

github.com/src-d/engine

Demo time!

babelfish

gitbase

jupiter

Challenge #3

# Learning from Source Code

# Neural Networks

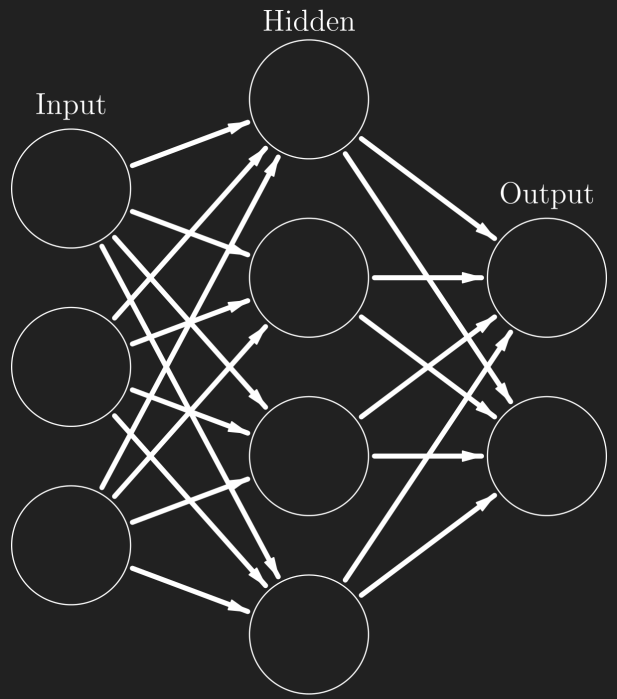
**Basically** fancy linear regression machines

Given an input of a constant length,
they predict an output of constant length.

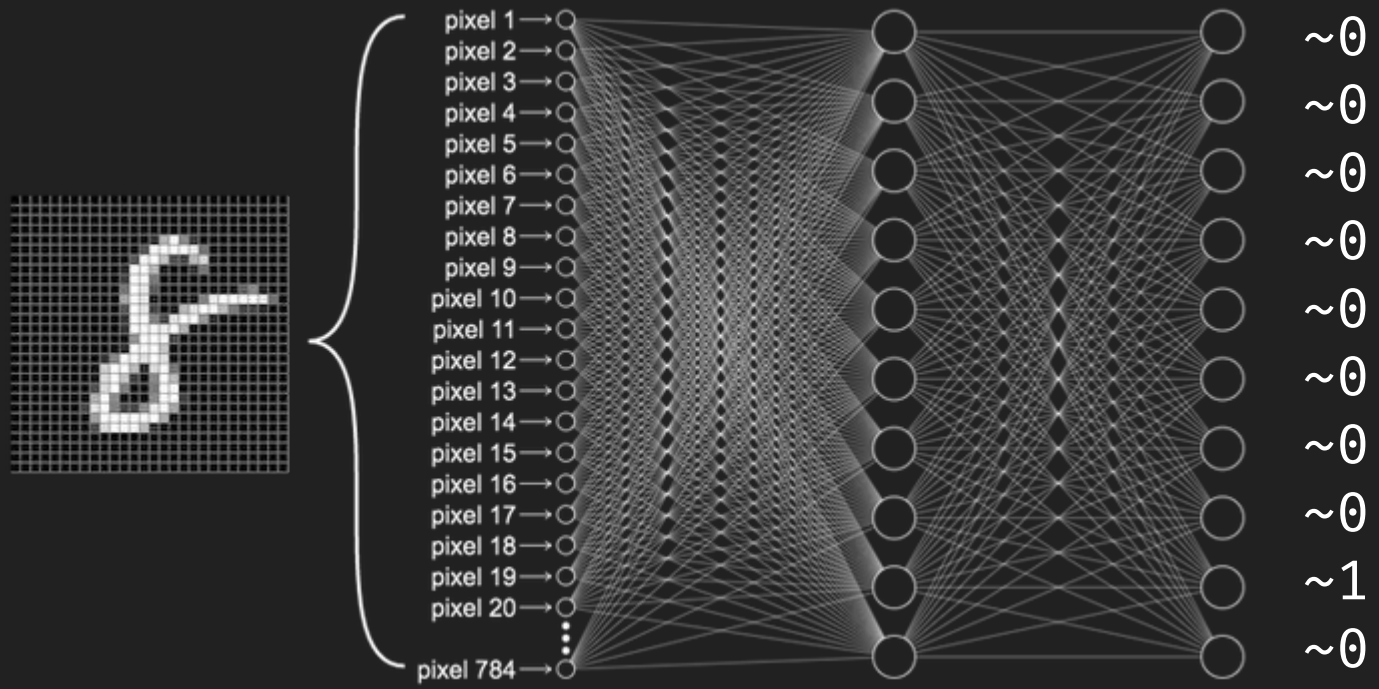
**Example:**

MNIST:

- Input: images with 28x28 px
- Output: a digit from zero to 9



# MNIST


pixel 1
pixel 2
pixel 3
pixel 4
pixel 5
pixel 6
pixel 7
pixel 8
pixel 9
pixel 10
pixel 11
pixel 12
pixel 13
pixel 14
pixel 15
pixel 16
pixel 17
pixel 18
pixel 19
pixel 20
pixel 784
~0
~0
~0
~0
~0
~0
~0
~0
~1
~0

# MLonCode: Predict the next token

# Recurrent Neural Networks

Can process sequences of variable length.

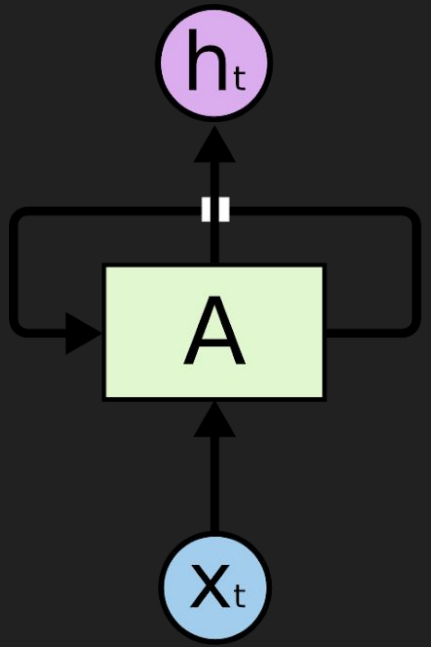
Uses its own output as a new input.

**Example:**

Natural Language Translation:

Input: “bonjour, les gauffres”

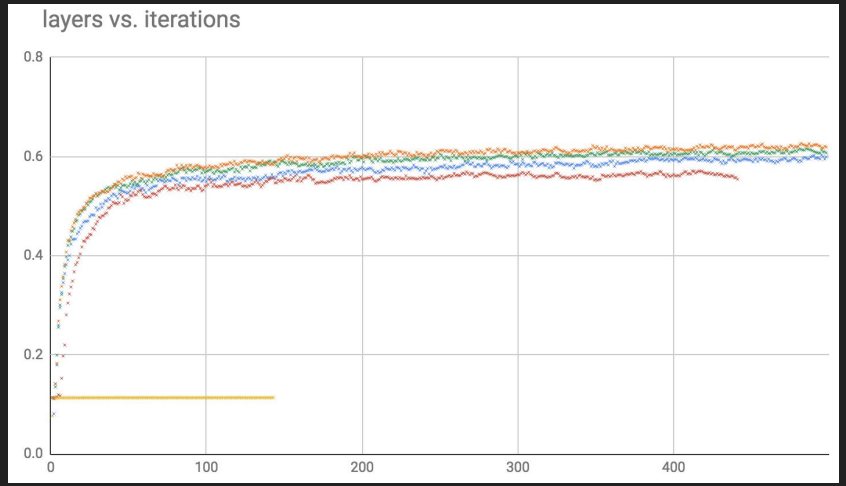
Output: “hi, waffles”



# MLonCode: Code Generation

charRNN: Given n characters, predict the next one

Trained over the Go standard library




Achieved  61% accuracy on predictions.

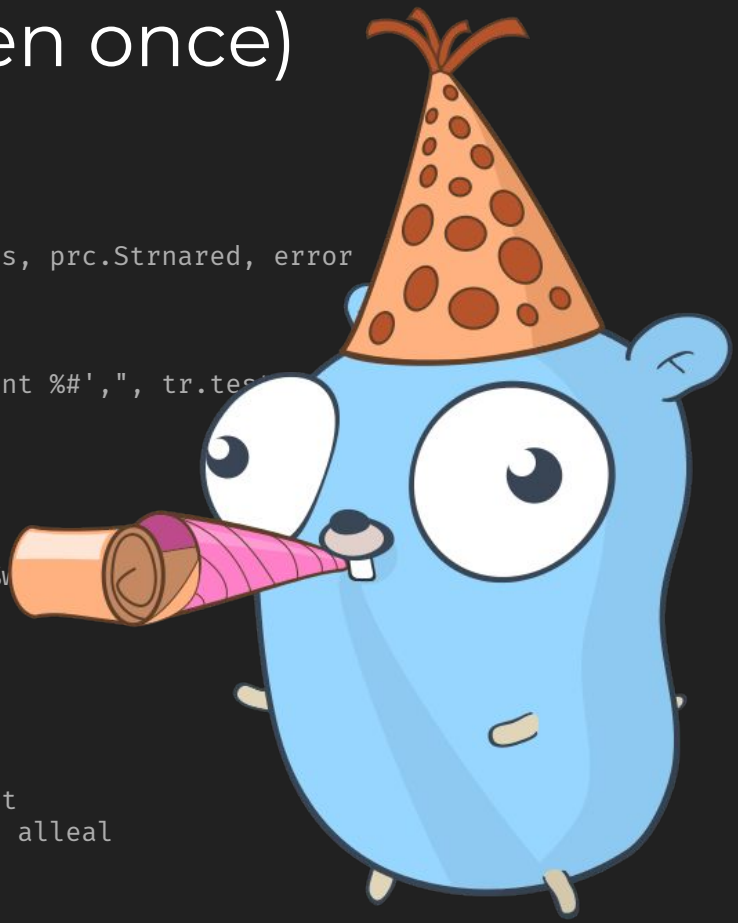
# Before training

r t,

kp0t@pp kpktp 0p000 xS%%%?ttk?^@p0rk^@%ppp@ta#p^@ #pp}}%p^@?P%^@@k#%@P}}ta S?@}^@t%@% %%aNt i ^@SSt@@ pyikkp?%y ?t k L P0L t% ^@i%yy ^@p i? %L%LL tyLP?a ?L@Ly?tkk^@ @^@ykk^@i#P^@iL@??@%1tt%^@tPTta L ^@LL%% %i1::yyy^@^@t tP @?@a#Patt 1^@@ k^@k ? yt%L1^@tP%k1?k? % ^@i ^@ta1?1taktt1P?a^@^@Pkt?#^@t^@##1?## #^@t11#:^@%??t%1^@a 1?a at1P ^@^@Pt #%^@^@ ^@aaak^@#a#?P1Pa^@tt%?^@kt?#akP ?#^@i%%aa ^@1%t tt?a?% t^@k^@^@k^@ a : ^@1 P# % ^@^@#t% :% kkP ^@#?P: t^@a ?%##?kkPaP^@ #a k?t?? ai?i%PPk taP% P^@ k^@iiai#?^@# #t ?# P?P^@ i^@ttPt # 1%11 ti a^@k P^@k ^@kt %^@%y?#a a#% @? kt ^@t%k? ^@PtttkL tkLa1 ^@iaay?p1% Pta tt ik?ty k^@kpt%^@tktpkryyp^@?pP# %kt?ki? i @t^@k^@%#P} ?at}akP##Pa11%^@i% ^@?ia ia%##%tki % }i%%}} a ay^@%yt }%t ^@tU%a% t}yi^@ ^@ @t yt%? aP @% ^@??^@%? ^@??k#% kk#%t?a: P}^@t :#^@#1t^@#: w^@P#%w:Pt t # t%aa%i@ak@@^@ka@^@a # y}^@# ^@? % tP i? ?tk ktPPt a tpprrpt? a^@ pP pt %p ? k? ^@^@ kP^@%%?tk a Pt^@# tP? P kkP1L1tP a%? t1P%PPti^@?%ytk %#%%t?@?^@ty^@iyk%1#^@@^@1#t a t@P^@^@ P@^@1P^@%%#@P:^@%^@ t 1:#P#@LtL#@L L1 %%dt??^@L ^@iBt yTk%p ^@i

# After one epoch (dataset seen once)

```
 if testingValuesIntering() {
        t.SetCaterCleen(time.SewsallSetrive(true)
        if weq := nil {
                t.Errorf("eshould: wont %v", touts anverals, prc.Strnared, error
        }
        t, err := ntr.Soare(cueper(err, err)
        if err != nil {
                t.Errorf("preveth dime il resetests:%d; want %#',", tr.tes
        }
        if err != nil {
        return
        }
        if err == nel {
                t.Errorf("LoconserrSathe foot %q::%q: %s;%w
        },
        defarenContateFule(temt.Canses)
}
if err != nil {
        return err
}
// Treters and restives of the sesconse stmpeletatareservet
// This no to the result digares wheckader. Constate bytes alleal
```



# After two epochs

```
        if !ok {
                t.Errorf("%d: %v not %v", i, err)
        }
        if !ot.Close()
        if enr != nil {
                t.Fatal(err)
        }
        if !ers != nil {
                t.Fatal(err)
        }
        if err != nil {
                t.Fatal(err)
        }
        if err != nil {
                t.Errorf("error %q: %s not %v", i, err)
        }
        return nil
}
```

# After many epochs

```
if got := t.struct(); !ok {
        t.Fatalf("Got %q: %q, %v, want %q", test, true
}
if !strings.Connig(t) {
        t.Fatalf("Got %q: %q", want %q", t, err)
}
if !ot {
        t.Errorf("%s < %v", x, y)
        }
        if !ok {
                t.Errorf("%d <= %d", err)
        }
        if !stricgs(); !ot {
                t.Errorf("!(%d <= %v", x, e)
        }
}
if !ot != nil {
        return ""
}
```

# Learning to Represent Programs with Graphs

*Miltiadis Allamanis, Marc Brockschmidt, Mahmoud Khademi*

https://arxiv.org/abs/1711.00740

**The *VARMISUSE* Task:**

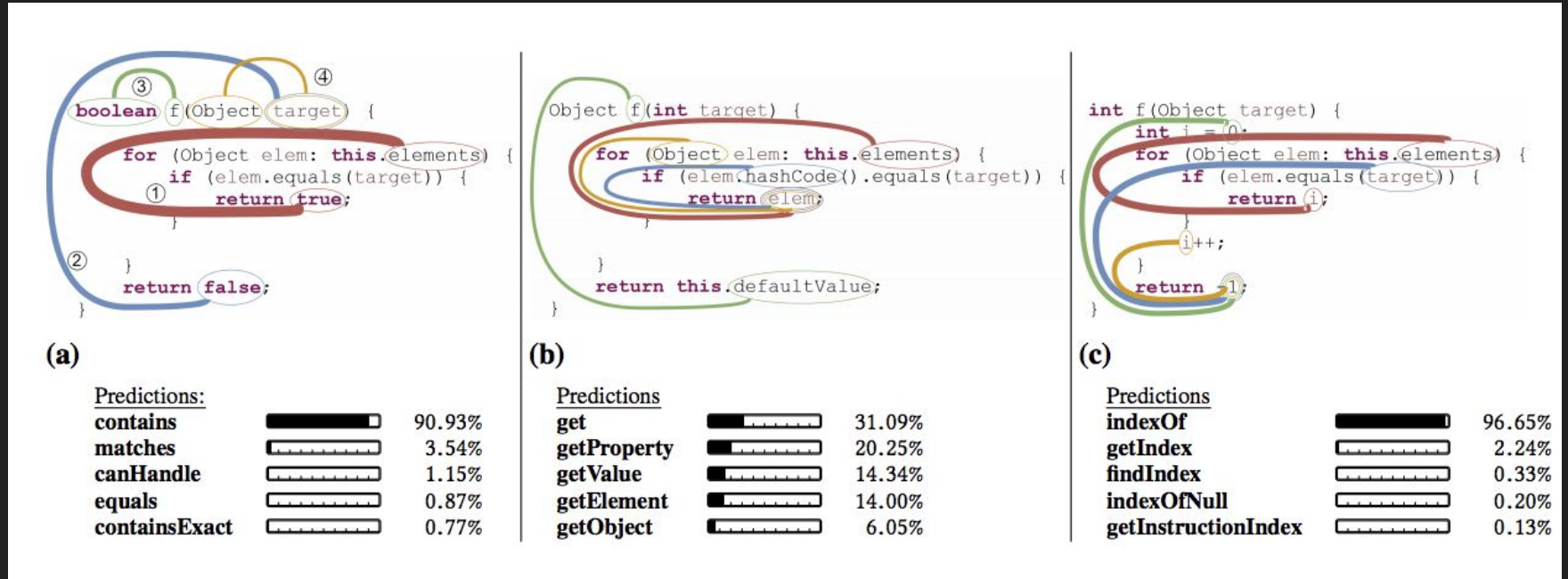
Given a program and a gap in it, predict what variable is missing.

```
from, err := os.Open("a.txt")
if err != nil {
     log.Fatal(err)
}
defer from.Close()

to, err := os.Open("b.txt")
if err != nil {
     log.Fatal(err)
}
defer ???.Close()

io.Copy(to, from)
```

# code2vec: Learning Distributed Representations of Code

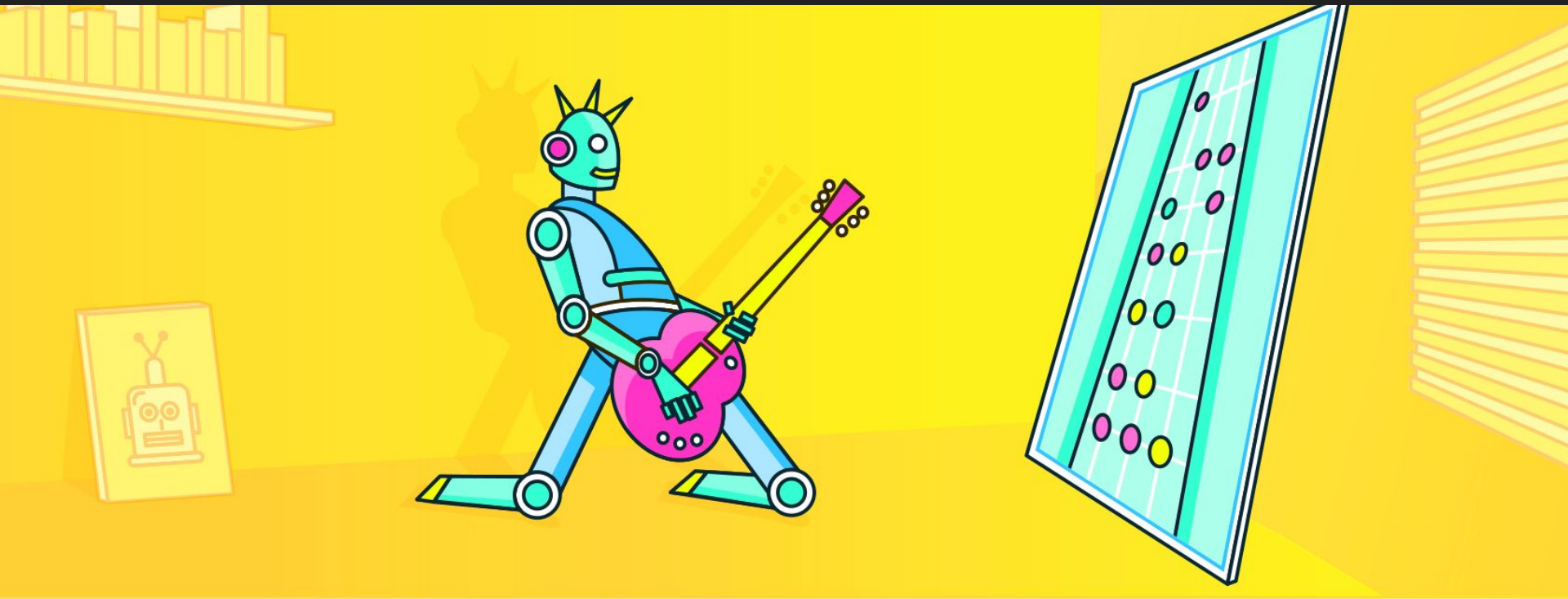
*Uri Alon, Meital Zilberstein, Omer Levy, Eran Yahav*

https://arxiv.org/abs/1803.09473 | https://code2vec.org/



# Much more research

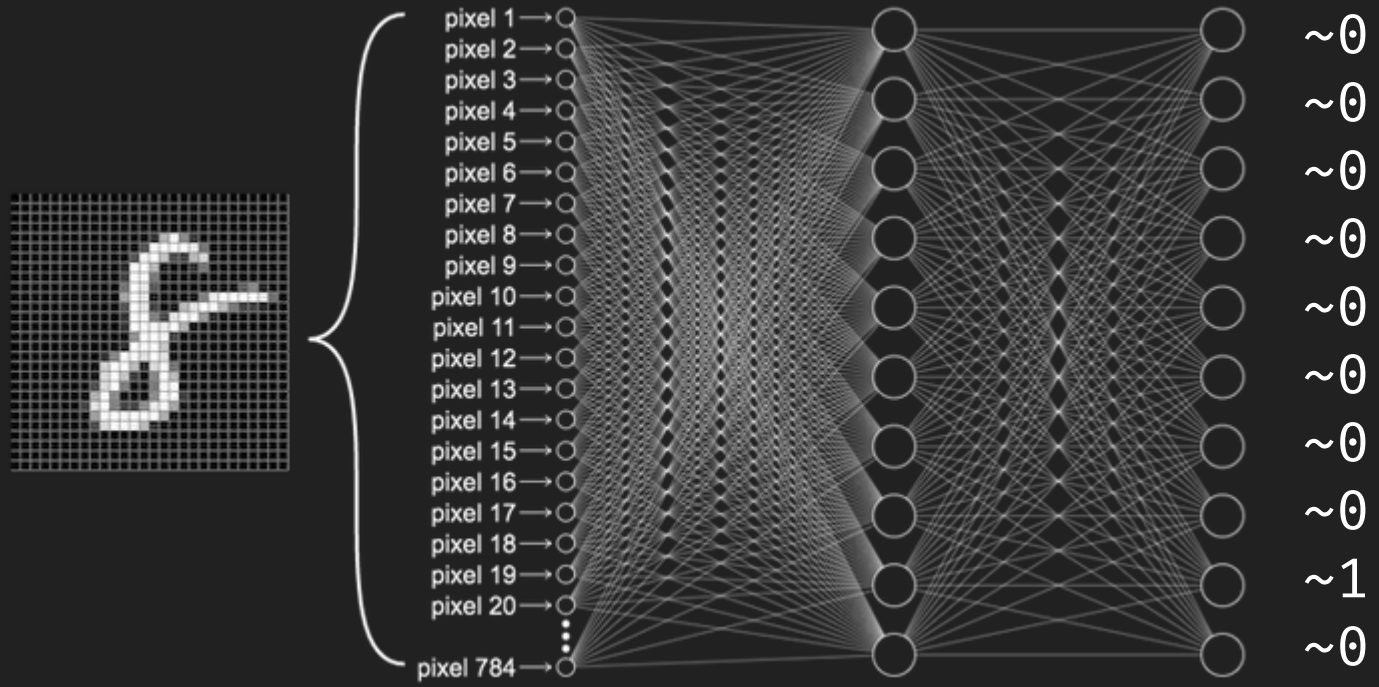
github.com/src-d/awesome-machine-learning-on-source-code



# Challenge #4

What can we build?

# Predictable vs Predicted



# An attention model for code reviews.

# misc/wasm: add polyfill for TextEncoder/TextDecoder for Edge support #27296

New issue

Open silbinarywolf wants to merge 1 commit into golang:master from silbinarywolf:fix-wasm-exec-for-microsoft-edge

Conversation 2 · Commits 1 · Checks 0 · Files changed 1

Changes from all commits · Jump to... · +76 −0 · Diff settings

76 misc/wasm/wasm_exec.js · View File (base) · View File (head) · View

```
@@ -27,6 +27,82 @@
                global.TextEncoder = util.TextEncoder;
                global.TextDecoder = util.TextDecoder;
        } else {
+               // Add polyfill for TextEncoder and TextDecoder for Microsoft Edge 17/18 support
+               // https://caniuse.com/#feat=textencoder
+               if (!window.TextEncoder) {
+                       TextEncoder = function(){}
+                       TextEncoder.prototype.encode = function (string) {
+                               var octets = [];
+                               var length = string.length;
+                               var i = 0;
+                               while (i < length) {
+                                       var codePoint = string.codePointAt(i);
+                                       var c = 0;
+                                       var bits = 0;
+                                       if (codePoint <= 0x0000007F) {
+                                               c = 0;
+                                               bits = 0x00;
+                                       } else if (codePoint <= 0x000007FF) {
+                                               c = 6;
+                                               bits = 0xC0;
+                                       } else if (codePoint <= 0x0000FFFF) {
+                                               c = 12;
+                                               bits = 0xE0;
+                                       } else if (codePoint <= 0x001FFFFF) {
+                                               c = 18;
+                                               bits = 0xF0;
+                                       }
+                                       octets.push(bits | (codePoint >> c));
+                                       c -= 6;
+                                       while (c >= 0) {
+                                               octets.push(0x80 | ((codePoint >> c) & 0x3F));
+                                               c -= 6;
+                                       }
+                                       i += codePoint >= 0x10000 ? 2 : 1;
+                               }
+                               return octets;
+                       };
+               }
+               if (!window.TextDecoder) {
+                       TextDecoder = function(){}
+                       TextDecoder.prototype.decode = function (octets) {
+                               var string = "";
+                               var i = 0;
+                               while (i < octets.length) {
+                                       var octet = octets[i];
+                                       var bytesNeeded = 0;
+                                       var codePoint = 0;
+                                       if (octet <= 0x7F) {
+                                               bytesNeeded = 0;
+                                               codePoint = octet & 0xFF;
+                                       } else if (octet <= 0xDF) {
+                                               bytesNeeded = 1;
+                                               codePoint = octet & 0x1F;
+                                       } else if (octet <= 0xEF) {
+                                               bytesNeeded = 2;
+                                               codePoint = octet & 0x0F;
+                                       } else if (octet <= 0xF4) {
+                                               bytesNeeded = 3;
+                                               codePoint = octet & 0x07;
+                                       }
+                                       if (octets.length - i - bytesNeeded > 0) {
+                                               var k = 0;
+                                               while (k < bytesNeeded) {
+                                                       octet = octets[i + k + 1];
+                                                       codePoint = (codePoint << 6) | (octet & 0x3F);
+                                                       k += 1;
+                                               }
+                                       } else {
+                                               codePoint = 0xFFFD;
+                                               bytesNeeded = octets.length - i;
+                                       }
+                                       string += String.fromCodePoint(codePoint);
+                                       i += bytesNeeded + 1;
+                               }
+                               return string
+                       };
+               }
+
                if (typeof window !== "undefined") {
                        window.global = window;
                } else if (typeof self !== "undefined") {
```

# misc/wasm: add polyfill for TextEncoder/TextDecoder for Edge support #27296

New issue

Open **silbinarywolf** wants to merge 1 commit into `golang:master` from `silbinarywolf:fix-wasm-exec-for-microsoft-edge`

Conversation 2 · Commits 1 · Checks 0 · Files changed 1

Changes from all commits · Jump to... · +76 −0 · Diff settings

76 misc/wasm/wasm_exec.js — View File (base) · View File (head) · View

```
@@ -27,6 +27,82 @@
                global.TextEncoder = util.TextEncoder;
                global.TextDecoder = util.TextDecoder;
        } else {
+               // Add polyfill for TextEncoder and TextDecoder for Microsoft Edge 17/18 support
+               // https://caniuse.com/#feat=textencoder
+               if (!window.TextEncoder) {
+                       TextEncoder = function(){}
+                       TextEncoder.prototype.encode = function (string) {
+                               var octets = [];
+                               var length = string.length;
+                               var i = 0;
+                               while (i < length) {
+                                       var codePoint = string.codePointAt(i);
+                                       var c = 0;
+                                       var bits = 0;
+                                       if (codePoint <= 0x0000007F) {
+                                               c = 0;
+                                               bits = 0x00;
+                                       } else if (codePoint <= 0x000007FF) {
+                                               c = 6;
+                                               bits = 0xC0;
+                                       } else if (codePoint <= 0x0000FFFF) {
+                                               c = 12;
+                                               bits = 0xE0;
+                                       } else if (codePoint <= 0x001FFFFF) {
+                                               c = 18;
+                                               bits = 0xF0;
+                                       }
+                                       octets.push(bits | (codePoint >> c));
+                                       c -= 6;
+                                       while (c >= 0) {
+                                               octets.push(0x80 | ((codePoint >> c) & 0x3F));
+                                               c -= 6;
+                                       }
+                                       i += codePoint >= 0x10000 ? 2 : 1;
+                               }
+                               return octets;
+                       };
+               }
+               if (!window.TextDecoder) {
+                       TextDecoder = function(){}
+                       TextDecoder.prototype.decode = function (octets) {
+                               var string = "";
+                               var i = 0;
+                               while (i < octets.length) {
+                                       var octet = octets[i];
+                                       var bytesNeeded = 0;
+                                       var codePoint = 0;
+                                       if (octet <= 0x7F) {
+                                               bytesNeeded = 0;
+                                               codePoint = octet & 0xFF;
+                                       } else if (octet <= 0xDF) {
+                                               bytesNeeded = 1;
+                                               codePoint = octet & 0x1F;
+                                       } else if (octet <= 0xEF) {
+                                               bytesNeeded = 2;
+                                               codePoint = octet & 0x0F;
+                                       } else if (octet <= 0xF4) {
+                                               bytesNeeded = 3;
+                                               codePoint = octet & 0x07;
+                                       }
+                                       if (octets.length - i - bytesNeeded > 0) {
+                                               var k = 0;
+                                               while (k < bytesNeeded) {
+                                                       octet = octets[i + k + 1];
+                                                       codePoint = (codePoint << 6) | (octet & 0x3F);
+                                                       k += 1;
+                                               }
+                                       } else {
+                                               codePoint = 0xFFFD;
+                                               bytesNeeded = octets.length - i;
+                                       }
+                                       string += String.fromCodePoint(codePoint);
+                                       i += bytesNeeded + 1;
+                               }
+                               return string
+                       };
+               }
+
        }

        if (typeof window !== "undefined") {
                window.global = window;
        } else if (typeof self !== "undefined") {
```

# Prediction vs Expectation

Can you see the mistake?

```
for i := 0; i < 10; i-- {
    if i %2 == 0 {
        fmt.Println("where's the mistake?")
    }
}
```

# Prediction vs Expectation

Can you see the mistake?

```
for i := 0; i < 10; i-- {
    if i %2 == 0 {
        fmt.Println("where's the mistake?")
    }
}
```

# VARMISUSE

Can you see the mistake?

```
from, err := os.Open("a.txt")
if err != nil {
    log.Fatal(err)
}
defer from.Close()

to, err := os.Open("b.txt")
if err != nil {
    log.Fatal(err)
}
defer from.Close()

io.Copy(to, from)
```

# VARMISUSE

Can you see the mistake?

```
from, err := os.Open("a.txt")
if err != nil {
    log.Fatal(err)
}
defer from.Close()

to, err := os.Open("b.txt")
if err != nil {
    log.Fatal(err)
}
defer from.Close()    ← s/from/to/

io.Copy(to, from)
```

# code2vec: Learning Distributed Representations of Code

Is this a good name?

```
func XXX(list []string, text string) bool {
     for _, s := range list {
          if s == text {
               return true
          }
     }
     return false
}
```

Suggestions:

- Contains
- Has

```
func XXX(list []string, text string) int {
     for i, s := range list {
          if s == text {
               return i
          }
     }
     return -1
}
```

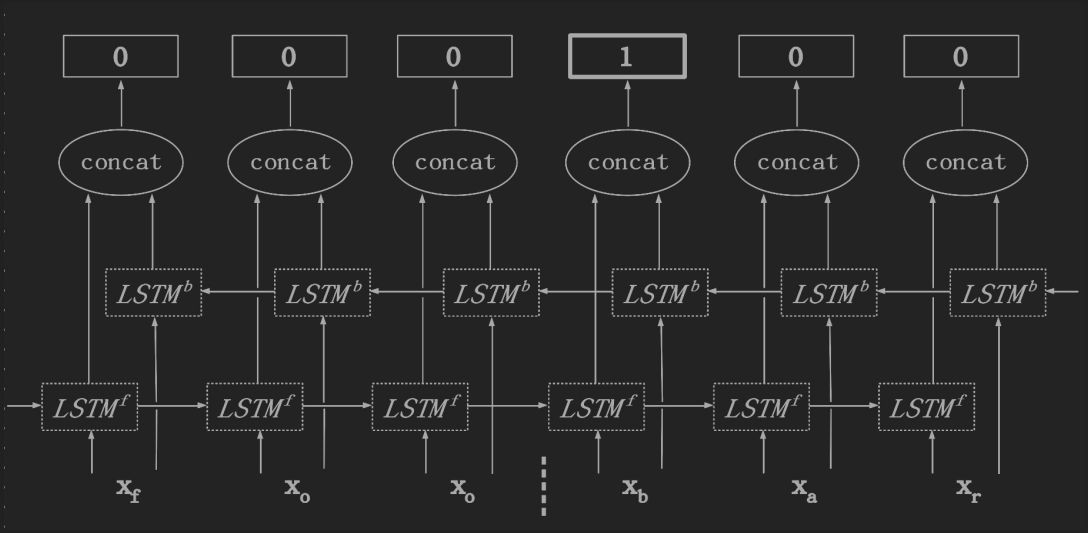
Suggestions:

- Find
- Index

# Splitting millions of identifiers with Deep Learning

https://blog.sourced.tech/post/idsplit/

isthisCorrect? → is this correct? → isThisCorrect?



Demo time!

learning Go

code2vec.org

neural splitter

# Assisted code review!

src-d/lookout


source: WOCinTech

# putting everything together

lookout

# And so much more

Coming up soon:

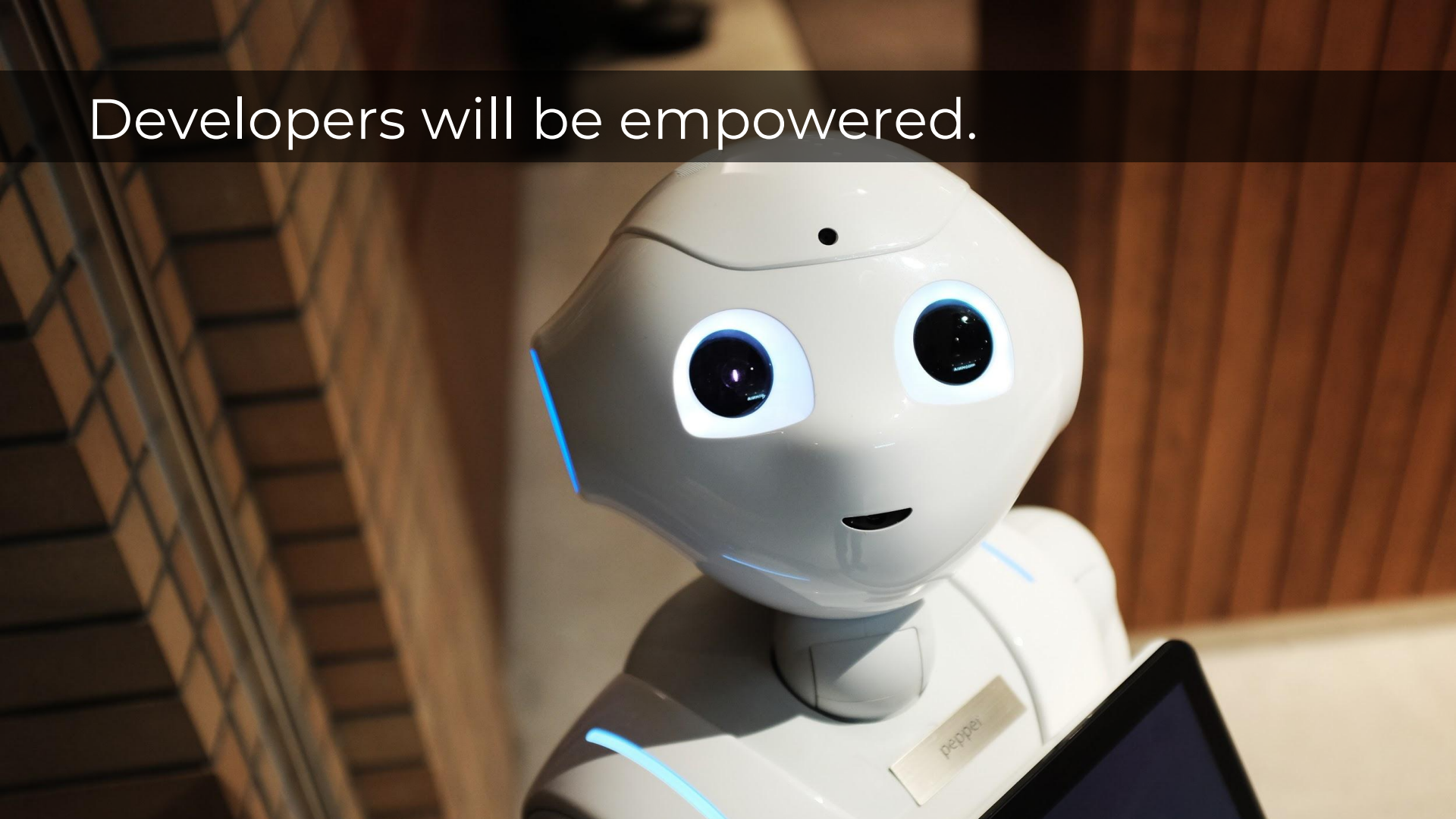
- Automated Style Guide Enforcing
- Bug Prediction

Coming ... later:

- ***Automated*** Code Review
- Code Generation: from unit tests, specification, natural language description.
- Natural Analysis: code description and conversational analysis.
- Education

# Will developers be replaced?

# Developers will be empowered.



# Want to know more?

- sourced.tech (pssh, we're hiring)
- bit.ly/awesome-mloncode
- francesc@sourced.tech
- come say hi, I have stickers

# Thanks

francesc

goto; copenhagen

GOTO Copenhagen 2018
Conference Nov. 19 - 21


Please

# Remember to rate this session

Thank you!

Follow us @gotocph