Machine Learning on Source Code #MLonCode

Francesc Campoy, source{d}

Machine Learning on Source Code

- The why
- The way
- The what?

Francesc Campoy

VP of Developer Relations
@francesc
francesc@sourced.tech

Previously Google Cloud (ML + Go)

speakerdeck.com/campoy/machine-learning-on-source-code

The why

Why do we want to do this?

Deep Learning Revolutions

- Computer Vision: ImageNet
- Natural Language Processing: Siri, Alexa, Google Assistant,
- Go: AlphaGo

source: wikimedia

Machine Learning on Source Code

- Automated code review
- Source code similarity: clustering + search
- Translation:
  - source code to natural language
  - source code to source code
  - natural language to source code?

source: WOCinTech

The way

How are we tackling the challenge?

What is Source Code

```
package main

import “fmt”

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

'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

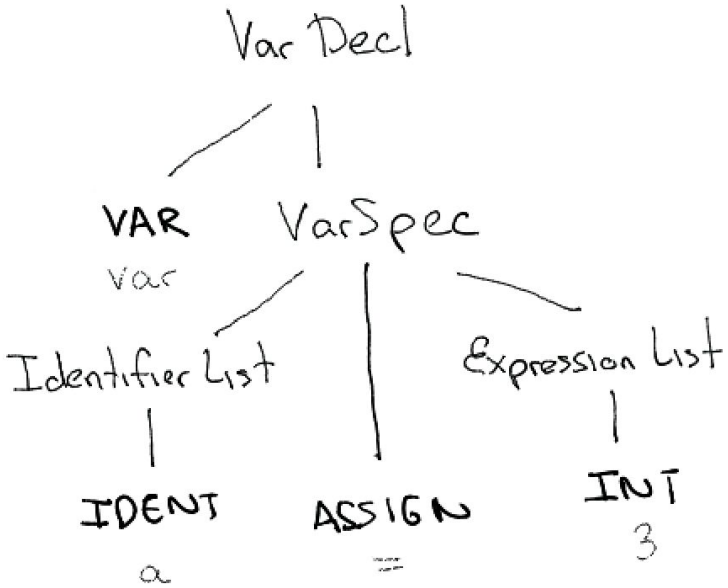
```
package main

import “fmt”

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

```
package package
IDENT main
;

import import
STRING "fmt"
;

func func
IDENT main
(
)
```

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

}
;
```

What is Source Code

```
package main

import “fmt”

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



Datasets

Data retrieval at source{d}

Rovers

- Crawls for git repos
- Supports:
  - GitHub
  - BitBucket
  - Cgit

github.com/src-d/rovers

Borges

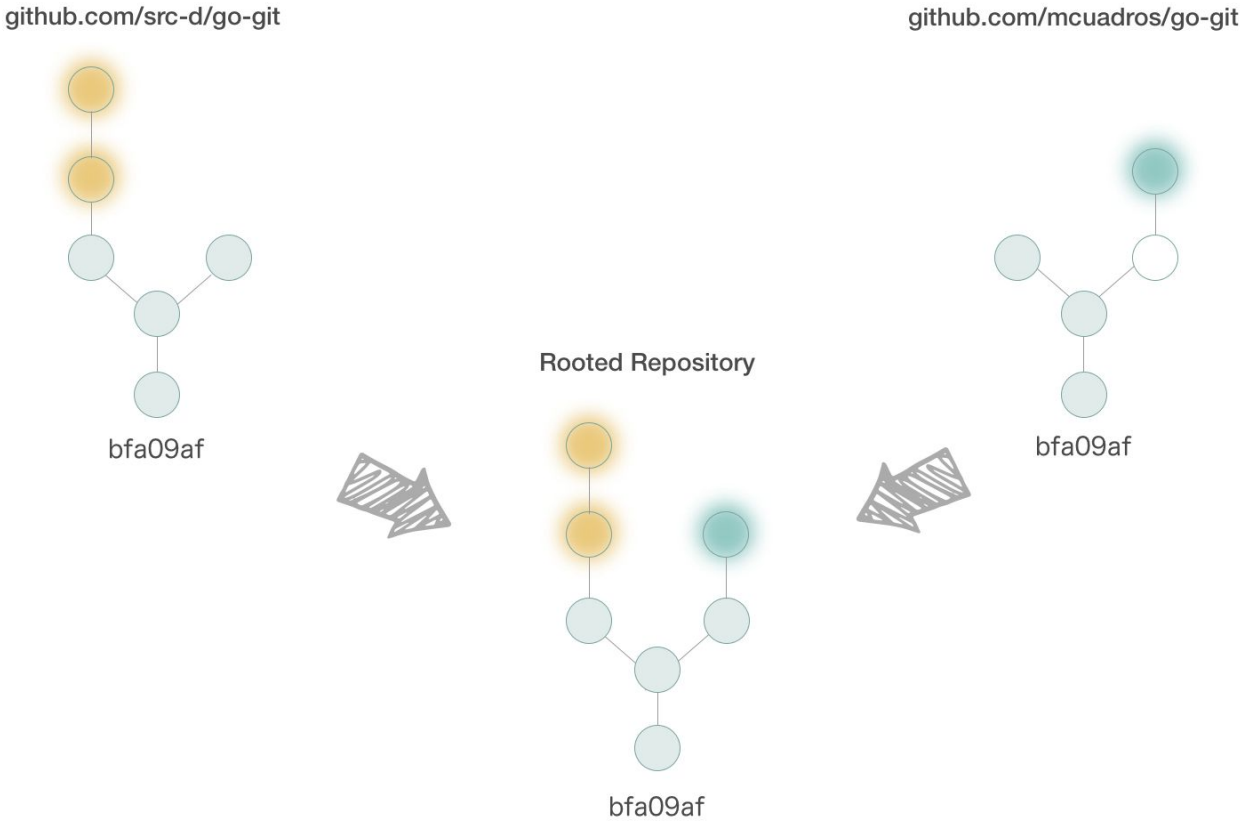
- Store git repositories
- Rooted repositories

github.com/src-d/borges

Siva

- Seekable Indexed Block Archiver
- Small, concatenable, constant access.

github.com/src-d/siva



Rooted repositories

Public Git Archive: a Big Code dataset for all

Vadim Markovtsev
source{d}
Madrid, Spain
vadim@sourced.tech

Waren Long
source{d}
Madrid, Spain
waren@sourced.tech

ABSTRACT

The number of open source software projects has been growing exponentially. The major online software repository host, GitHub, has accumulated tens of millions of publicly available Git version-controlled repositories. Although the research potential enabled by the available open source code is clearly substantial, no significant large-scale open source code datasets exist. In this paper, we present the Public Git Archive – dataset of 182,014 top-bookmarked Git repositories from GitHub. We describe the novel data retrieval pipeline to reproduce it. We also elaborate on the strategy for performing dataset updates and legal issues. The Public Git Archive occupies 3.0 TB on disk and is an order of magnitude larger than the current source code datasets. The dataset is made available through HTTP and provides the source code of the projects, the related metadata, and development history. The data retrieval pipeline employs an optimized worker queue model and an optimized archive format to efficiently store forked Git repositories, reducing the amount of data to download and persist. Public Git Archive aims to open a myriad of new opportunities for "Big Code" research.

control accessible, therefore universal. The next stage of the revolution is permitting the automatic analysis of source code at scale, to support data-driven language design, to infer best (and worst) practices, and to provide the raw data to data hungry machine learning techniques that will be the basis of the next generation of development tools [3, 15]. It requires source code archives that are both big and programmatically accessible for analysis.

The GHTorrent project [12] took first steps in this direction, focusing on metadata in order to be scalable. Current source code datasets typically contain tens of thousands of projects at most [3] and are dedicated to particular programming languages such as Java and JavaScript [25], thus lacking diversity and attracting critics [6]. Software Heritage [7] is a recent attempt to archive all the open source code ever written, however no public dataset has been published yet by them.

We present the Public Git Archive, the first big code dataset amenable to programmatic analysis at scale. It is by far the biggest curated archive of top-rated[1] repositories on GitHub, see Table 1 for comparison. The Public Git Archive targets large-scale quantitative research in the areas of source code analysis (SCA) and ma-

https://arxiv.org/abs/1803.10144

src-d / datasets

Watch 13 | Star 10 | Fork 7

Code | Issues 8 | Pull requests 0 | Insights

Branch: master ▾ datasets / PublicGitArchive / README.md

Find file | Copy path

vmarkovtsev Update PGA readme — b95e20d 23 days ago

3 contributors

112 lines (74 sloc) 4.04 KB

Raw | Blame | History

Public Git Archive

size 3.0TB

Paper (accepted to MSR'18).

This dataset consists of two parts:

- Siva files with Git repositories.
- Index file in CSV format.

github.com/src-d/datasets

Learning from Source Code

Learning from Code

Code is:

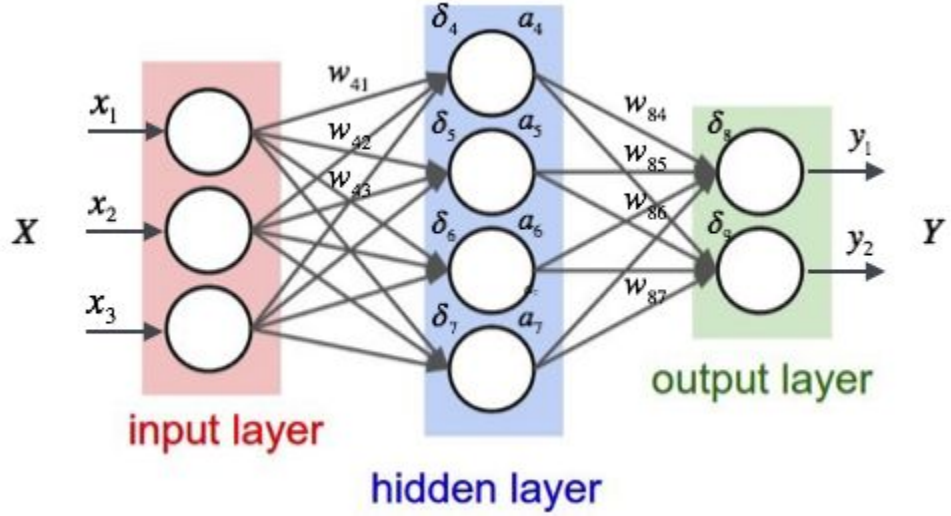
- a sequence of bytes
- a sequence of tokens
- a tree

Learning sequences

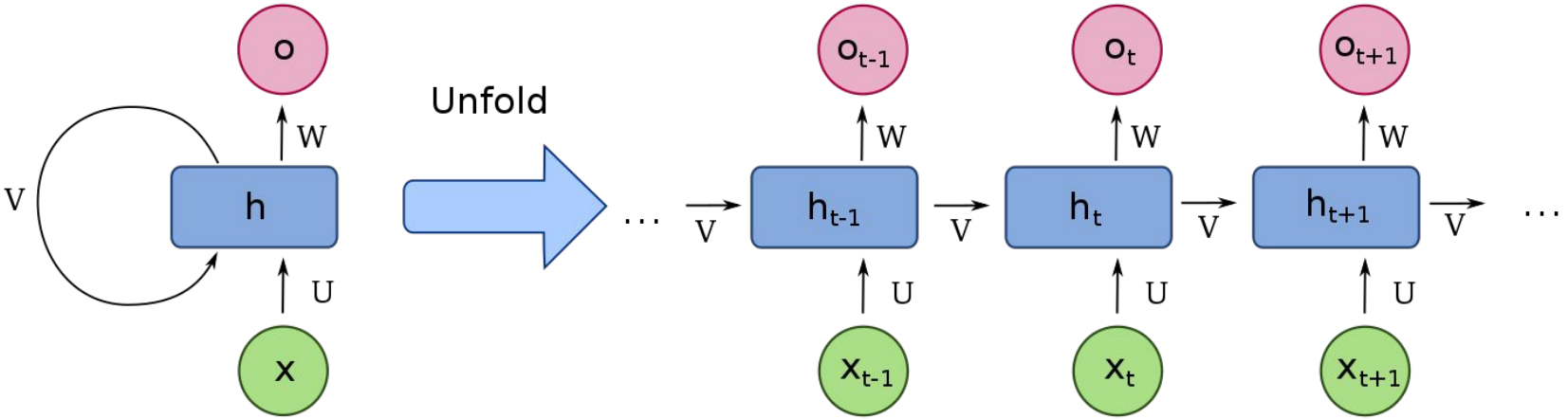
	Input	Output
Speech recognition:	audio clip	text
Sentiment analysis:	text	rating -1 to 1
Machine Translation:	Hello, everyone!	Hola a tothom!
Video recognition	Sequence of frames	text

Learning from Characters

Neural networks, right?



Recurrent Neural Networks



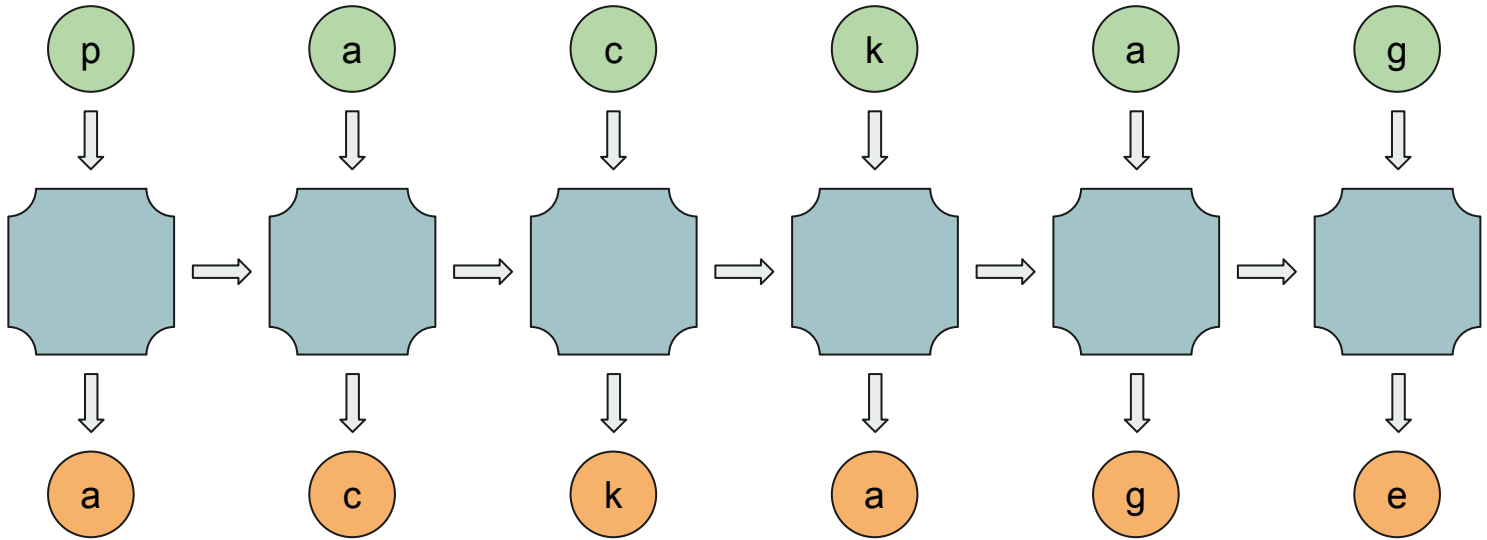
Recurrent Neural Networks

source: Martin Gorner

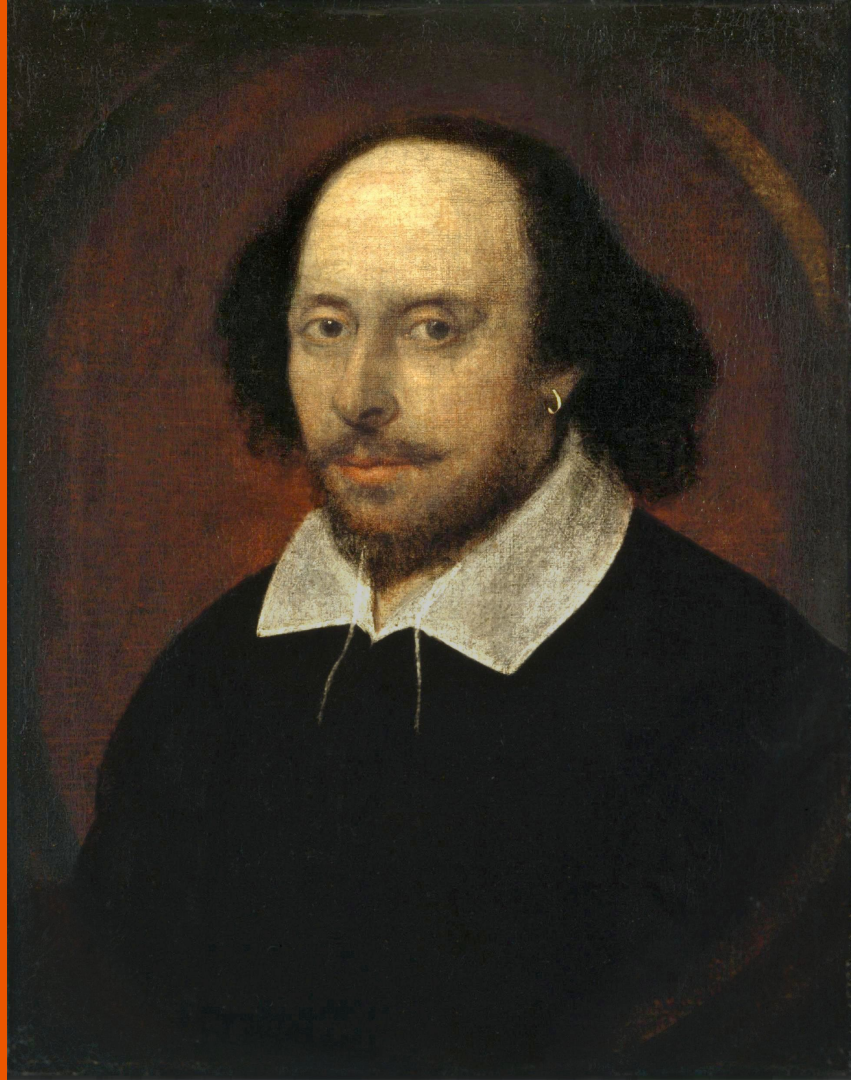
Predicting the next character

Input	Output	
‘p’ ‘a’, ‘c’, ‘k’, ‘a’, ‘g’:	‘e’	80%
	‘i’	15%
	...	

Training the neural network



Demo time!



More demo time!

Is this useful?

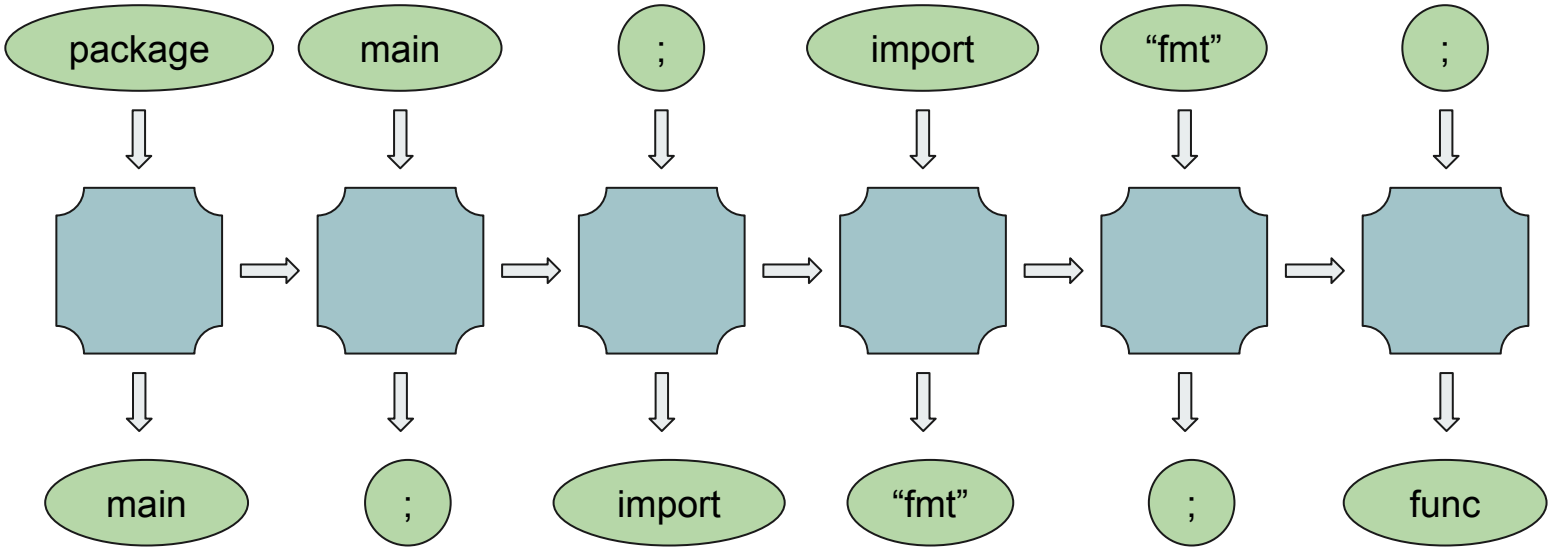
- Predicting implies understanding
- We can predict aspects of code:
  - Help with assisted editing
  - Detect anomalies

splittingidentifiersforthewin

(Paper coming soon)

Learning from Tokens

We could use the same mechanism!



But … one-hot encoding

Categorical (non continuous) variables require one-hot encoding:

Predict characters zero to 9:

0 -> [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1 -> [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
2 -> [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
3 -> [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
…
9 -> [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

But … one-hot encoding

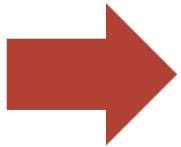
Possible characters in human text:	**hundreds**
Possible characters in computer programs:	**hundreds**
Possible words in English:	**171,476 (Oxford dictionary*)**

* urban dictionary not included

How many possible identifiers in X language?

Word embeddings

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

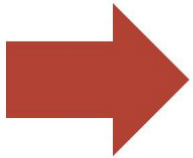


	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Each word gets a 1x9 vector representation

Try to build a lower dimensional embedding

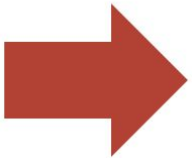
Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch



	Femininity	Youth	Royalty
Man			
Woman			
Boy			
Girl			
Prince			
Princess			
Queen			
King			
Monarch			

Try to build a lower dimensional embedding

Vocabulary:
Man, woman, boy, girl, prince, princess, queen, king, monarch

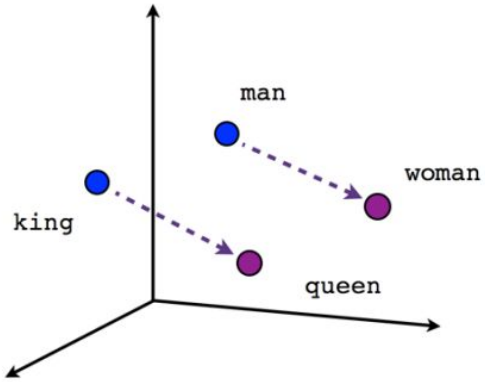


	Femininity	Youth	Royalty
Man	0	0	0
Woman	1	0	0
Boy	0	1	0
Girl	1	1	0
Prince	0	1	1
Princess	1	1	1
Queen	1	0	1
King	0	0	1
Monarch	0.5	0.5	1

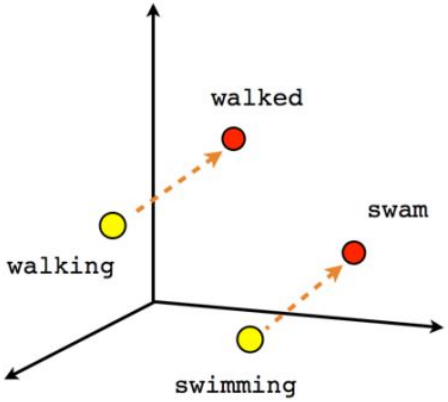
Each word gets a 1x3 vector

Similar words… similar vectors

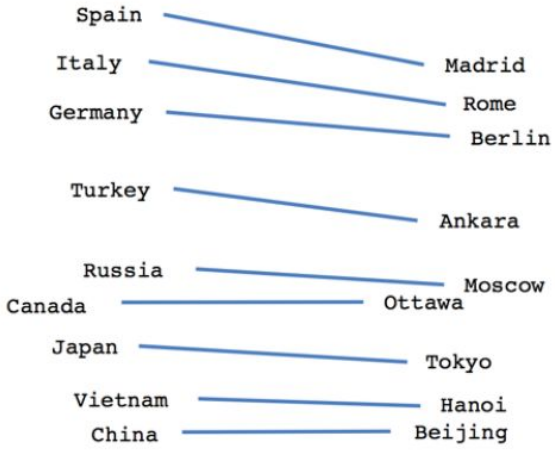
word2vec



Male-Female



Verb tense



Country-Capital

Splitting and stemming

allSeenElements

splitting (and we know how!)

all, seen, elements

stemming

all, see, element

id2vec

Vocabulary size: 500,000

Dataset: >100,000 most starred repos on GitHub

Result: https://github.com/src-d/ml

source code identifier embeddings

By **Vadim Markovtsev** 05 December 2017

This post is related to a talk we gave in Moscow in June at our Machine Learning on Source Code (MLoSC) conference and research we did at the beginning of this year: presentation and video.

Let's start with revising what "embeddings" are, then proceed with describing approaches to word2vec, then explain how this technique can be transferred from NLP to MLoSC, present some examples of the results and finish with instructions on how to reproduce this work.

embeddings

Suppose that we work with elements from an abstract N-dimensional space. In other words, each element is a series of numbers, the length of that series is N - that is, a vector of length N.

https://blog.sourced.tech/post/id2vec/

Receive is to send as read is to ...

___

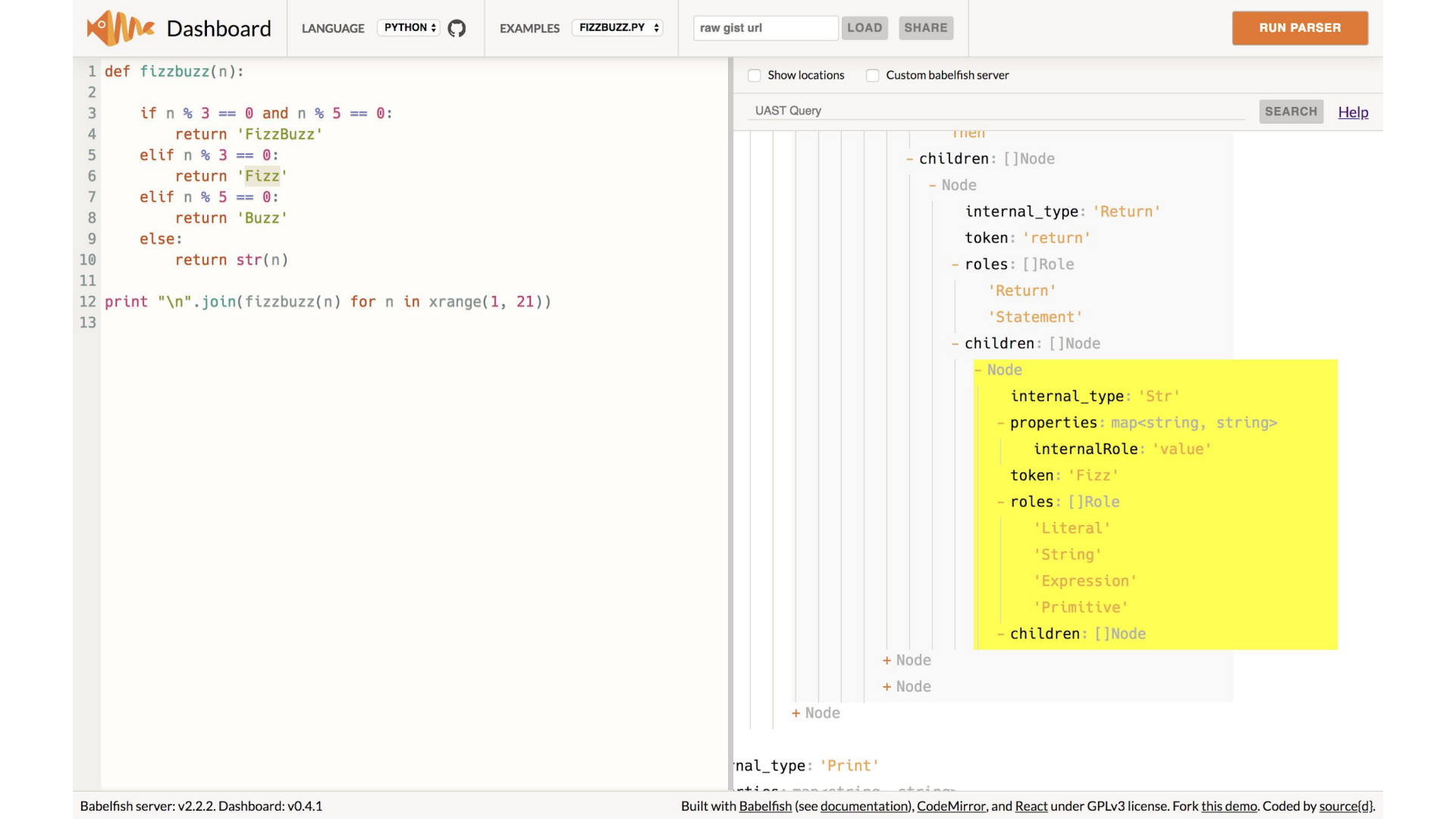
Learning from Trees

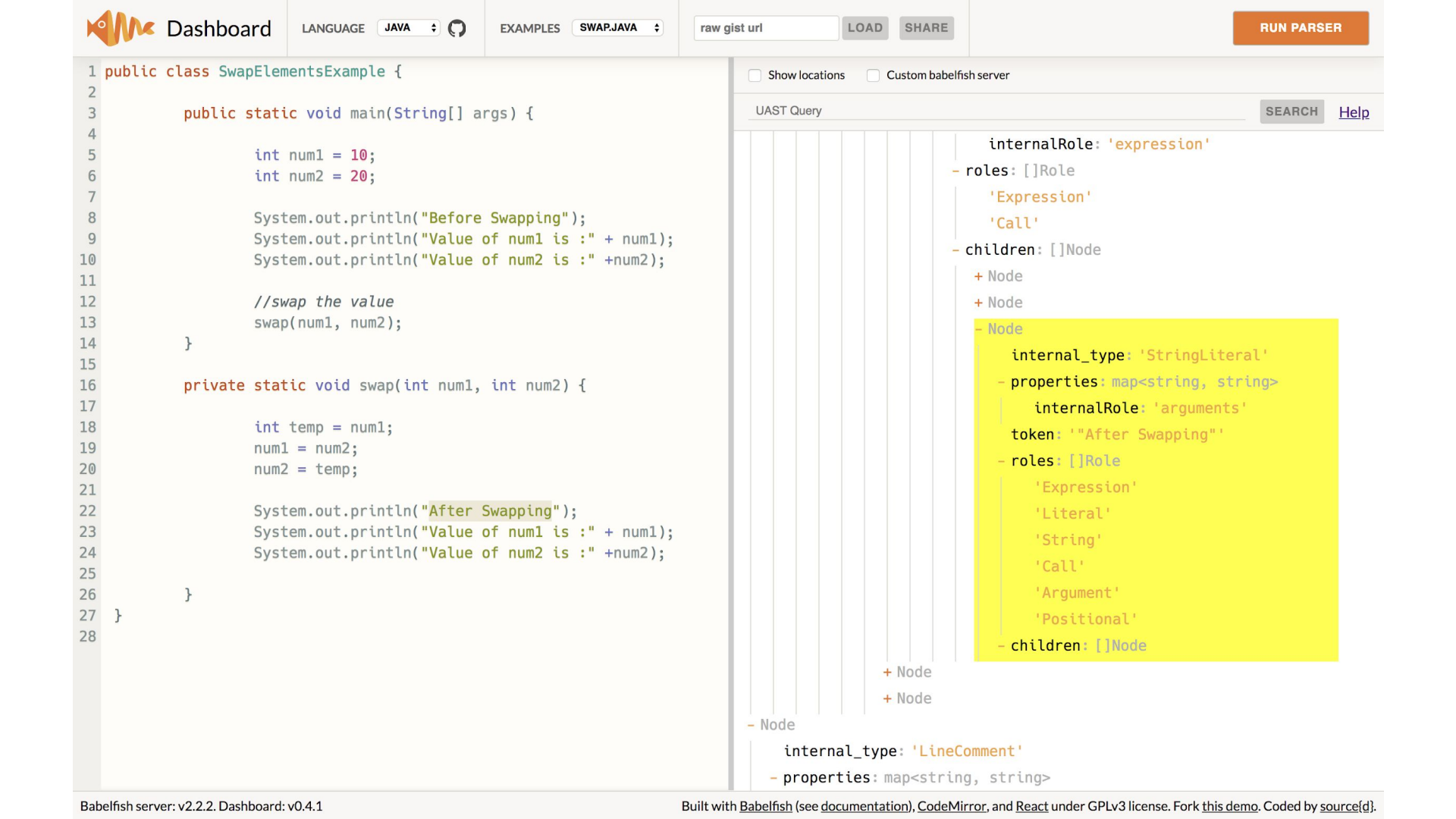
Abstract Syntax Trees

- Each programming language has its own grammar
- Each grammar generates slightly different ASTs
- We want to learn from *all* languages!

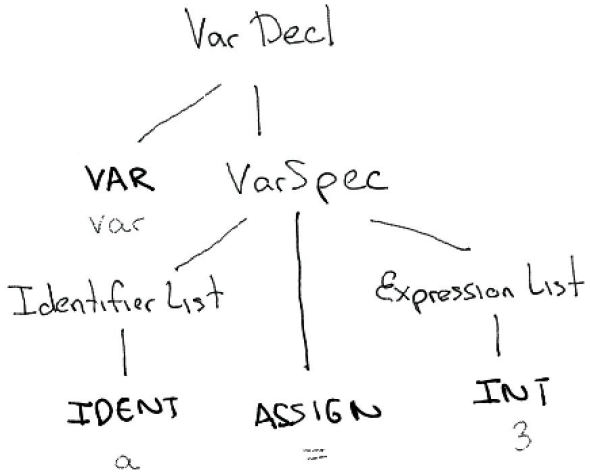
Universal Abstract Syntax Trees

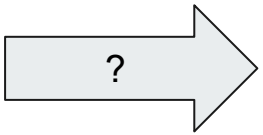
Babelfish (https://bblf.sh)

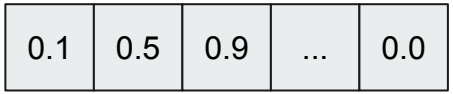

Dashboard
LANGUAGE
PYTHON
EXAMPLES
FIZZBUZZ.PY
raw gist url
LOAD
SHARE
RUN PARSER
1 def fizzbuzz(n):
2
3 if n % 3 == 0 and n % 5 == 0:
4 return 'FizzBuzz'
5 elif n % 3 == 0:
6 return 'Fizz'
7 elif n % 5 == 0:
8 return 'Buzz'
9 else:
10 return str(n)
11
12 print "\n".join(fizzbuzz(n) for n in xrange(1, 21))
13
Show locations
Custom babelfish server
UAST Query
SEARCH
Help
- children: []Node
- Node
internal_type: 'Return'
token: 'return'
- roles: []Role
'Return'
'Statement'
- children: []Node
- Node
internal_type: 'Str'
- properties: map<string, string>
internalRole: 'value'
token: 'Fizz'
- roles: []Role
'Literal'
'String'
'Expression'
'Primitive'
- children: []Node
+ Node
+ Node
+ Node
nal_type: 'Print'
Babelfish server: v2.2.2. Dashboard: v0.4.1
Built with Babelfish (see documentation), CodeMirror, and React under GPLv3 license. Fork this demo. Coded by source{d}.


Dashboard
LANGUAGE
JAVA
EXAMPLES
SWAP.JAVA
raw gist url
LOAD
SHARE
RUN PARSER
public class SwapElementsExample {
public static void main(String[] args) {
int num1 = 10;
int num2 = 20;
System.out.println("Before Swapping");
System.out.println("Value of num1 is :" + num1);
System.out.println("Value of num2 is :" +num2);
//swap the value
swap(num1, num2);
}
private static void swap(int num1, int num2) {
int temp = num1;
num1 = num2;
num2 = temp;
System.out.println("After Swapping");
System.out.println("Value of num1 is :" + num1);
System.out.println("Value of num2 is :" +num2);
}
}
Show locations
Custom babelfish server
UAST Query
SEARCH
Help
internalRole: 'expression'
- roles: []Role
'Expression'
'Call'
- children: []Node
+ Node
+ Node
- Node
internal_type: 'StringLiteral'
- properties: map<string, string>
internalRole: 'arguments'
token: '"After Swapping"'
- roles: []Role
'Expression'
'Literal'
'String'
'Call'
'Argument'
'Positional'
- children: []Node
+ Node
+ Node
- Node
internal_type: 'LineComment'
- properties: map<string, string>
Babelfish server: v2.2.2. Dashboard: v0.4.1
Built with Babelfish (see documentation), CodeMirror, and React under GPLv3 license. Fork this demo. Coded by source{d}.

Structural Embeddings







Structural Embedding of Syntactic Trees for Machine Comprehension

Rui Liu,* **Junjie Hu,*** **Wei Wei,*** **Zi Yang,*** **Eric Nyberg**
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh PA 15213, USA
{ruil, junjieh, weiwei, ziy, ehn}@cs.cmu.edu

Abstract

Deep neural networks for machine comprehension typically utilizes only word or character embeddings without explicitly taking advantage of structured linguistic information such as constituency trees and dependency trees. In this paper, we propose *structural embedding of syntactic trees* (SEST), an algorithm framework to

stituency tree and *dependency tree* into consideration. Such techniques have been proven to be useful in many natural language understanding tasks in the past and illustrated noticeable improvements such as the work by (Rajpurkar et al., 2016). In this paper, we adopt similar ideas but apply them to a neural attention model for question answering.

The constituency tree (Manning et al., 1999) of a sentence defines internal nodes and terminal nodes to represent phrase structure grammars

https://arxiv.org/abs/1703.00572

A General Path-Based Representation for Predicting Program Properties

Uri Alon
Technion
Haifa, Israel
urialon@cs.technion.ac.il

Meital Zilberstein
Technion
Haifa, Israel
mbs@cs.technion.ac.il

Omer Levy
University of Washington
Seattle, WA
omerlevy@cs.washington.edu

Eran Yahav
Technion
Haifa, Israel
yahave@cs.technion.ac.il

Abstract

Predicting program properties such as names or expression types has a wide range of applications. It can ease the task of programming, and increase programmer productivity. A major challenge when learning from programs is *how to represent programs in a way that facilitates effective learning*.

Our approach We present a novel program representation for learning from programs. Our approach uses different path-based abstractions of the program's abstract syntax tree. This family of path-based representations is natural, general, fully automatic, and works well across different tasks and programming languages.

https://arxiv.org/abs/1803.09544

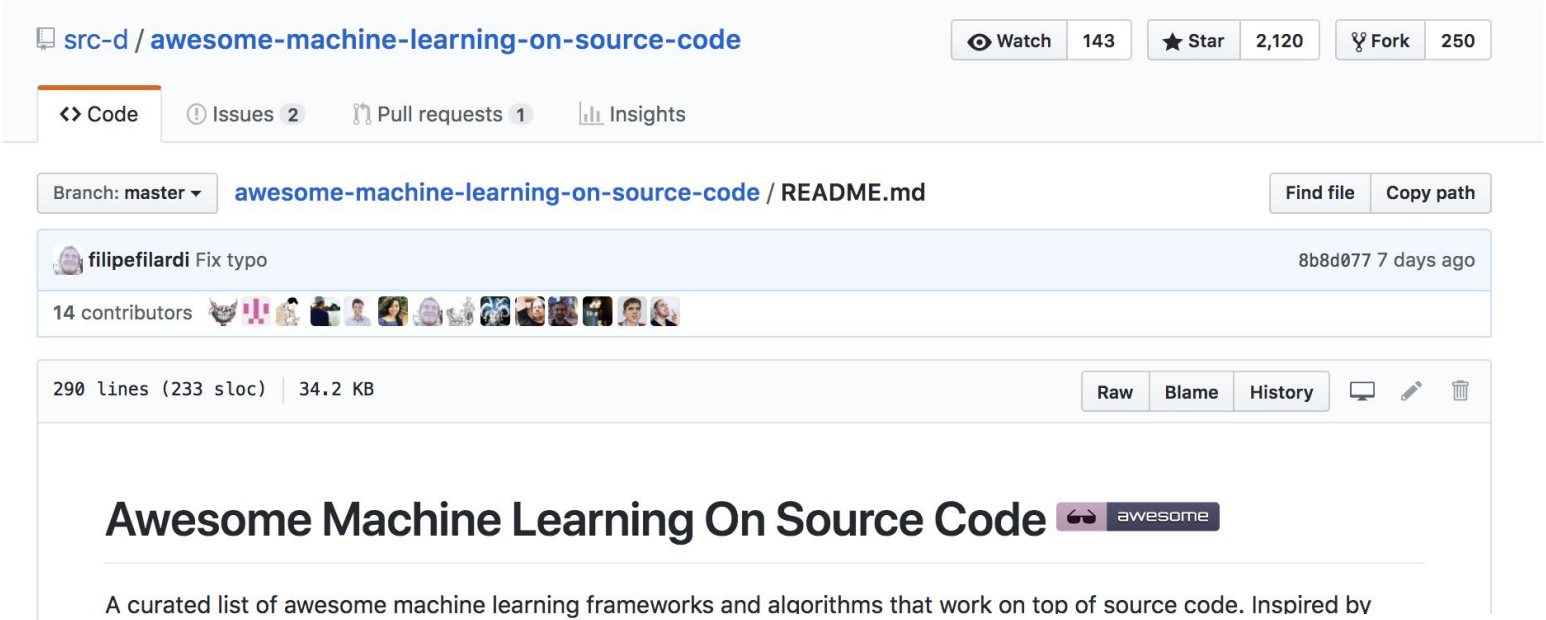
And more coming soon!

PS: we're hiring

Other projects

- Vecino: finding similar repositories
- Apollo: finding source code duplication at scale
- TMSC: topic modeling on source code repositories
- Snippet-Ranger: topic modeling on source code snippets

Awesome Machine Learning on Source Code



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

Want to know more?

@sourcedtech
sourced.tech
slack!

Thanks

Francesc Campoy
source{d}

 @francesc
 francesc@sourced.tech