Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-Class Learning

Muhammad Ikram*, Hassan Jameel Asghar, Mohamed Ali Kaafar, Anirban Mahanti, and Balachandar Krishnamurthy

Proceedings on Privacy Enhancing Technologies 2017 (1):79–99

COMS E6998 (Spring 2019)

Nathan Reitinger

Overview

(1) Background

- web privacy
    - canvas fingerprinting case study

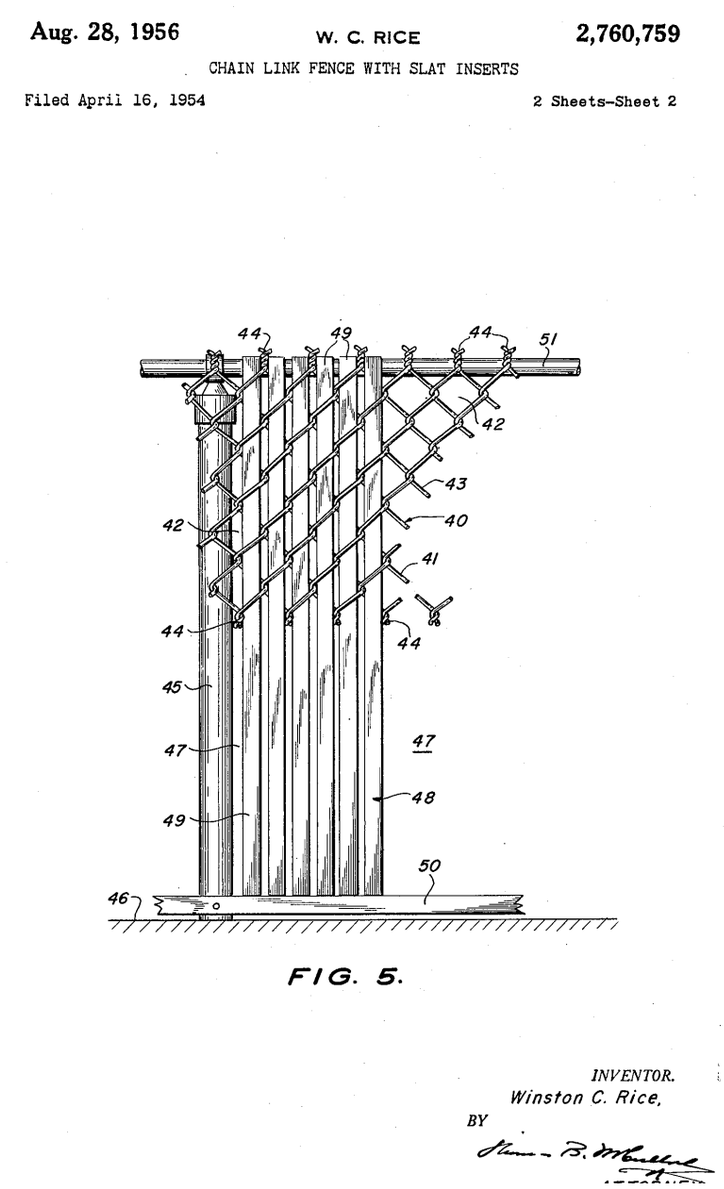
(2) Experiments

- selenium web scrape
- semantic similarity
    - *using web corpus statistics for program analysis* (i.e., obtaining canonical forms)
        - three address code
        - program dependency graph (PDG)
        - tf-idf based on PDG n-grams
- one-class SVM

(3) Evaluation

(4) Limitations

(5) Key Takeaways



Background

- privacy: who cares — is there a tracking problem?

You have zero privacy anyway, get over it. — Scott McNealy, Sun Microsystems 1999

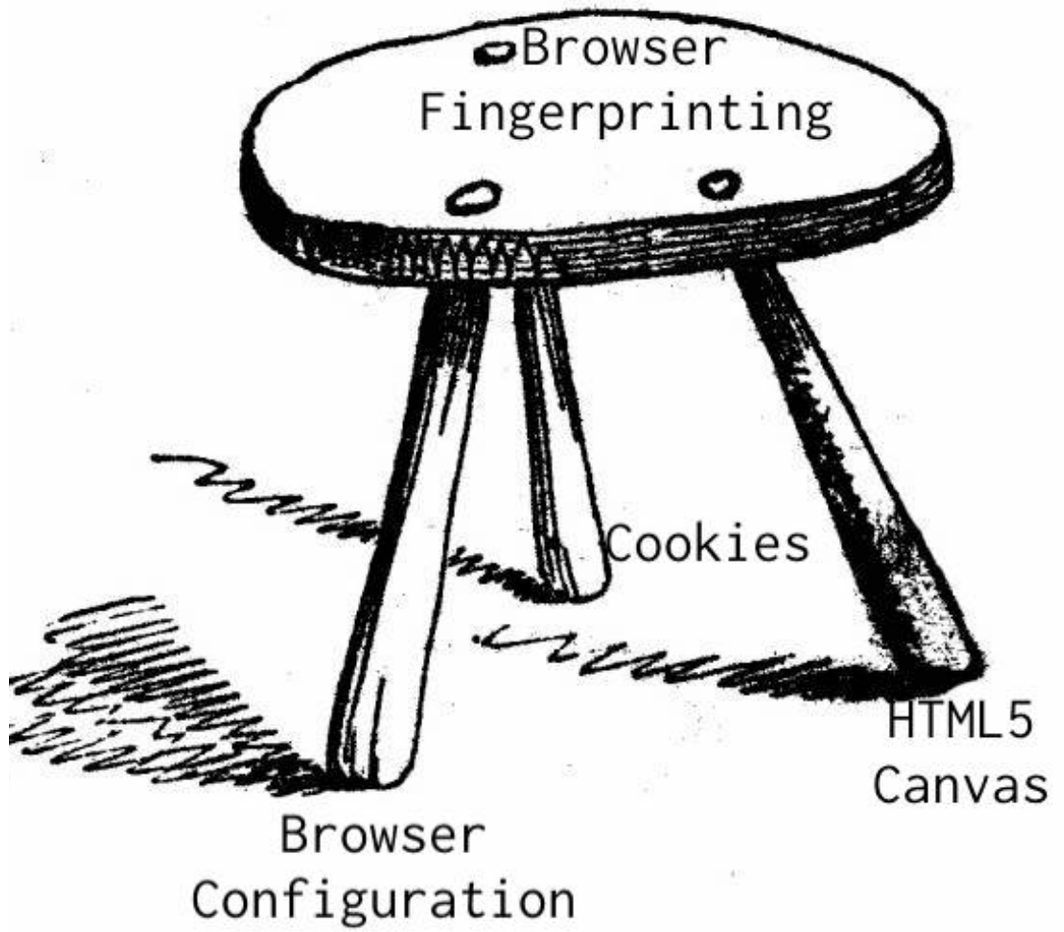
Background

privacy: is there a tracking problem?

- What is Apple talking about

- <u>browser configurations</u> to prevent fingerprinting

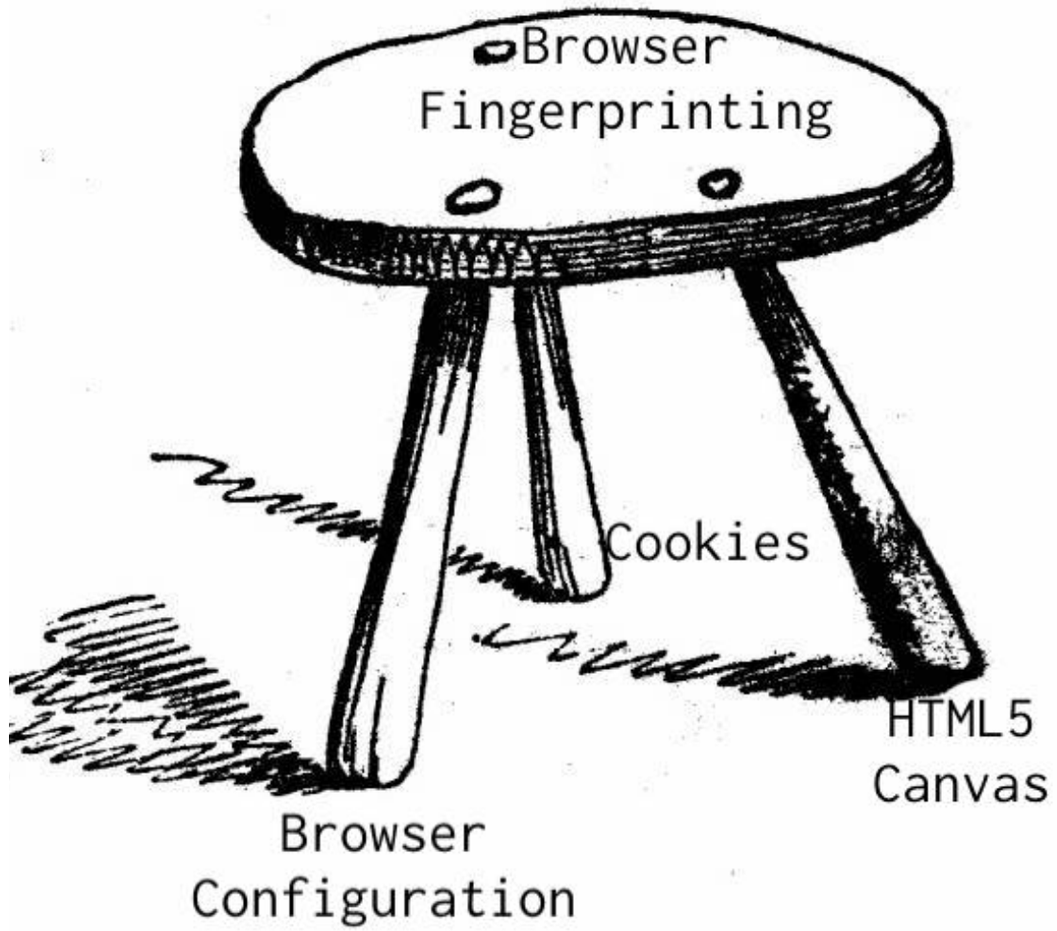
- <u>limited</u> data collection (and use of <u>differential privacy</u>)



Background

privacy: is there a tracking problem?

- Cookies
  - user consent ✓
- Browser Configurations
  - Tor, Firefox, Safari ✓
- Canvas
  - HTML5 X



Background

privacy: is there a tracking problem?

Pixel Perfect: Fingerprinting Canvas in HTML5

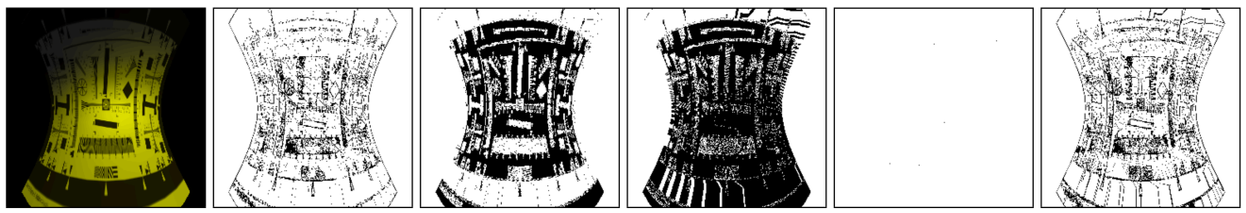
Keaton Mowery and Hovav Shacham

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California, USA

- computing systems vary widely from one to the other, both in hardware and software

- a repeated request to draw something unique on the canvas produces high entropy

- $$E = -\sum_{i=1}^{n} p(x_i) log_2(x_i)$$
  where $p(x_i)$ is the size of the ith group divided by the number of samples



(a) Original (Intel G41) (b) Group 1 (Radeon HD 2400) (c) Group 20 (Intel 82945G) (d) Group 23 (Intel G33/G31) (e) Group 25 (Intel HD Graphics) (f) Group 36 (GeForce 6200)

Figure 10: Original render and difference maps for Group 24

Background

privacy: is there a tracking problem?

Pixel Perfect: Fingerprinting Canvas in HTML5

Keaton Mowery and Hovav Shacham

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California, USA

- Panopticlick (EFF)

> Your browser fingerprint **appears to be unique** among the 216,541 tested in the past 45 days.
>
> Currently, we estimate that your browser has a fingerprint that conveys **at least 17.72 bits of identifying information.**

Background

privacy: is there a tracking problem?

Pixel Perfect: Fingerprinting Canvas in HTML5

Keaton Mowery and Hovav Shacham

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California, USA

Browser Characteristic	bits of identifying information	one in x browsers have this value	value
User Agent	7.58	190.79	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1 Safari/605.1.15
HTTP_ACCEPT Headers	4.65	25.02	text/html, */*; q=0.01 br, gzip, deflate en-us
Browser Plugin Details	9.49	719.41	Plugin 0: Shockwave Flash; Shockwave Flash 32.0 r0; Flash Player.plugin; (Shockwave Flash; application/x-shockwave-flash; swf) (FutureSplash Player; application/futuresplash; spl). Plugin 1: WebKit built-in PDF; ; ; (Portable Document Format; application/pdf; pdf) (Portable Document Format; text/pdf; pdf) (PostScript; application/postscript; ps).
Time Zone	3.54	11.6	240
Screen Size and Color Depth	5.45	43.69	2560x1440x24
System Fonts	17.72	216541.0	Al Bayan Bold, Al Bayan Plain, Al Nile, Al Nile Bold, Al Tarikh Regular, … Wingdings, Wingdings 2, Wingdings 3, Zapf Dingbats, Zapfino (via Flash)
Are Cookies Enabled?	0.25	1.19	Yes
Limited supercookie test	0.38	1.3	DOM localStorage: Yes, DOM sessionStorage: Yes, IE userData: No
Hash of canvas fingerprint	9.18	580.54	73ae61a5d5b43b7e350a02e59a437316
Hash of WebGL fingerprint	9.79	883.84	cb465998f4a380c5a57fabef97da7f8d
DNT Header Enabled?	1.22	2.33	FALSE
Language	0.95	1.94	en-US
Platform	3.27	9.63	MacIntel
Touch Support	0.75	1.68	Max touchpoints: 0; TouchEvent supported: false; onTouchStart supported: false

Background

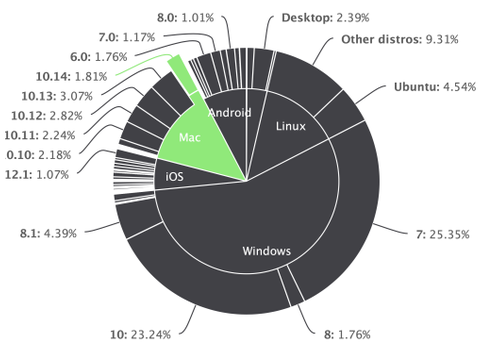
privacy: is there a tracking problem?

Pixel Perfect: Fingerprinting Canvas in HTML5

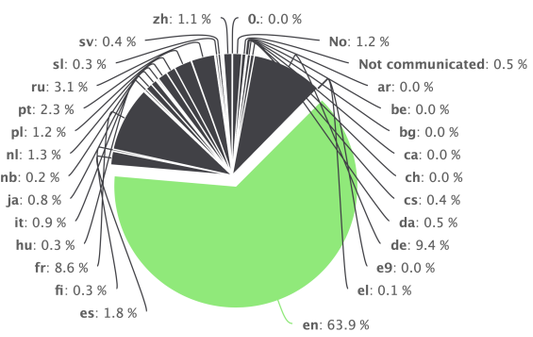
Keaton Mowery and Hovav Shacham

Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California, USA

Operating systems

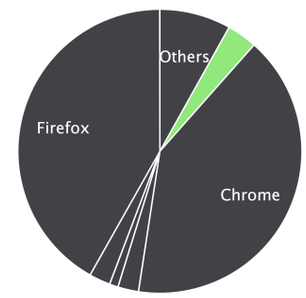


Languages

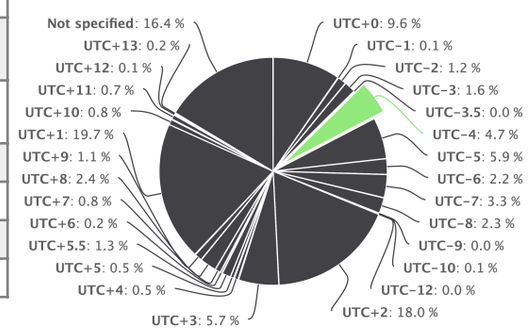


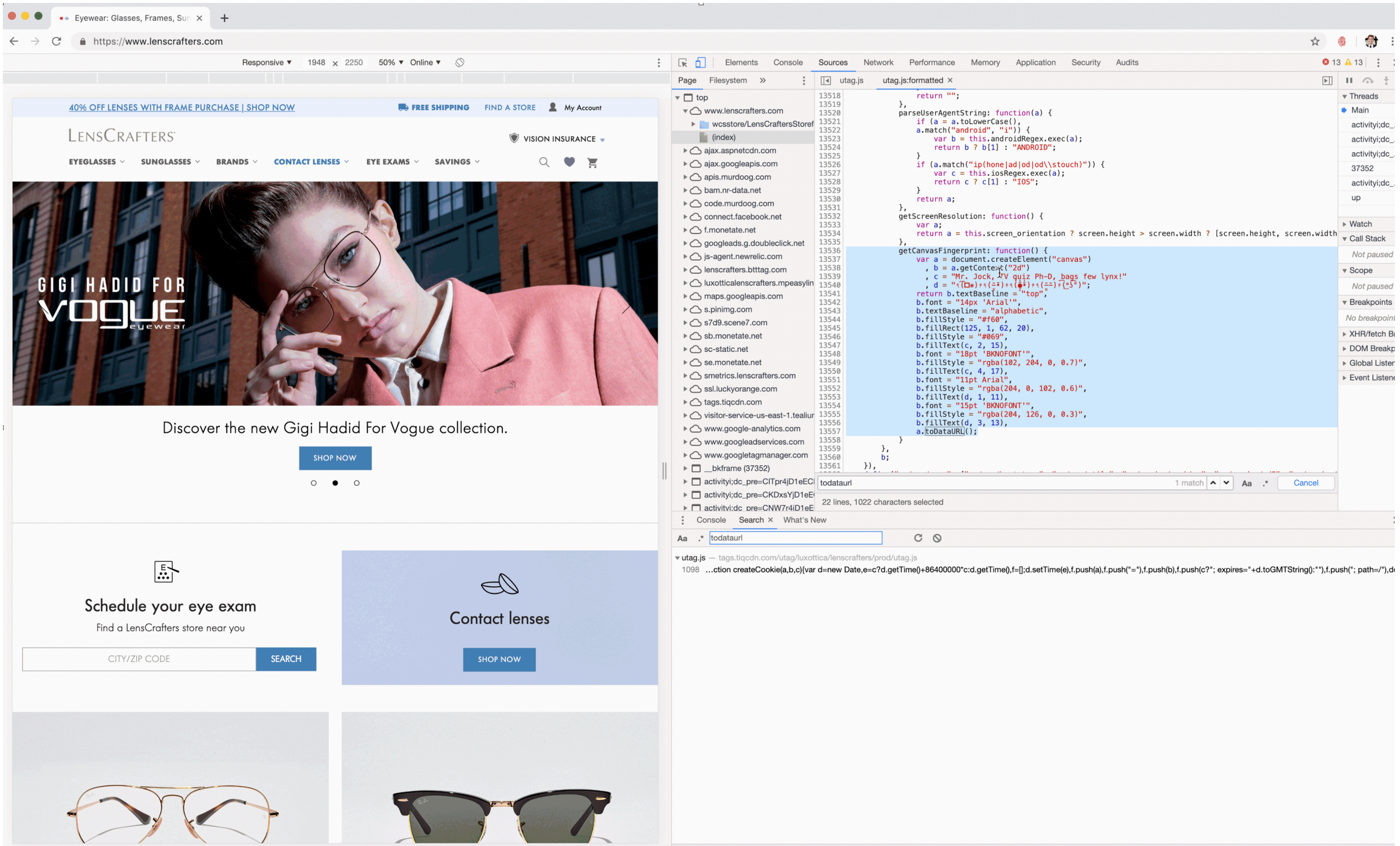
Browser Characteristic	bits of identifying information	one in *x* browsers have this value	value
User Agent	7.58	190.79	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/ 12.1 Safari/605.1.15
HTTP_ACCEPT Headers	4.65	25.02	text/html, */*; q=0.01 br, gzip, deflate en-us
Browser Plugin Details	9.49	719.41	Plugin 0: Shockwave Flash; Shockwave Flash 32.0 r0; Flash Player.plugin; (Shockwave Flash; application/x-shockwave-flash; swf) (FutureSplash Player; application/futuresplash; spl). Plugin 1: WebKit built-in PDF; ; ; (Portable Document Format; application/pdf; pdf) (Portable Document Format; text/pdf; pdf) (PostScript; application/postscript; ps).
Time Zone	3.54	11.6	240
Screen Size and Color Depth	5.45	43.69	2560x1440x24
System Fonts	17.72	216541.0	Al Bayan Bold, Al Bayan Plain, Al Nile, Al Nile Bold, Al Tarikh Regular, … Wingdings, Wingdings 2, Wingdings 3, Zapf Dingbats, Zapfino (via Flash)
Are Cookies Enabled?	0.25	1.19	Yes
Limited supercookie test	0.38	1.3	DOM localStorage: Yes, DOM sessionStorage: Yes, IE userData: No
Hash of canvas fingerprint	9.18	580.54	73ae61a5d5b43b7e350a02e59a437316
Hash of WebGL fingerprint	9.79	883.84	cb465998f4a380c5a57fabef97da7f8d
DNT Header Enabled?	1.22	2.33	FALSE
Language	0.95	1.94	en-US
Platform	3.27	9.63	MacIntel
Touch Support	0.75	1.68	Max touchpoints: 0; TouchEvent supported: false; onTouchStart supported: false

Browsers



Timezones





Mr. Jock, TV quiz Ph-D, bags few lynx! — an almost perfect pangram (26 letters of alphabet)

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

- guess what: this happens in the wild—and it's not limited to canvas fingerprinting
- January 2016 scrape of top 1 million sites (Alexa Top Sites)
  - battery (battery status API)
  - font suite (browser font list)
  - webRTC (in-browser voice and video)
  - audio (audioContext API)

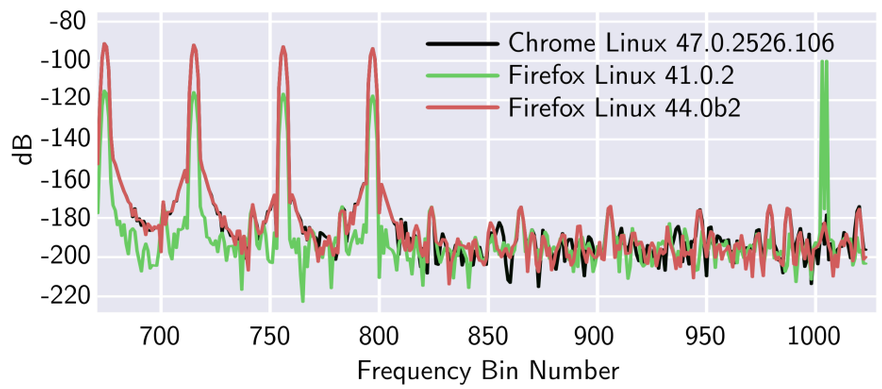


Figure 9: Visualization of processed `OscillatorNode` output from the fingerprinting script https://www.cdn-net.com/cc.js for three different browsers on the same machine. We found these values to remain constant for each browser after several checks.

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

- one of the most popular tools ===>
  - draws on a large amount of device-specific components

```
var components = [
  { key: 'userAgent', getData: UserAgent },
  { key: 'webdriver', getData: webdriver },
  { key: 'language', getData: languageKey },
  { key: 'colorDepth', getData: colorDepthKey },
  { key: 'deviceMemory', getData: deviceMemoryKey },
  { key: 'pixelRatio', getData: pixelRatioKey },
  { key: 'hardwareConcurrency', getData: hardwareConcurrencyKey },
  { key: 'screenResolution', getData: screenResolutionKey },
  { key: 'availableScreenResolution', getData: availableScreenResolutionKey },
  { key: 'timezoneOffset', getData: timezoneOffset },
  { key: 'timezone', getData: timezone },
  { key: 'sessionStorage', getData: sessionStorageKey },
  { key: 'localStorage', getData: localStorageKey },
  { key: 'indexedDb', getData: indexedDbKey },
  { key: 'addBehavior', getData: addBehaviorKey },
  { key: 'openDatabase', getData: openDatabaseKey },
  { key: 'cpuClass', getData: cpuClassKey },
  { key: 'platform', getData: platformKey },
  { key: 'doNotTrack', getData: doNotTrackKey },
  { key: 'plugins', getData: pluginsComponent },
  { key: 'canvas', getData: canvasKey },
  { key: 'webgl', getData: webglKey },
  { key: 'webglVendorAndRenderer', getData: webglVendorAndRendererKey },
  { key: 'adBlock', getData: adBlockKey },
  { key: 'hasLiedLanguages', getData: hasLiedLanguagesKey },
  { key: 'hasLiedResolution', getData: hasLiedResolutionKey },
  { key: 'hasLiedOs', getData: hasLiedOsKey },
  { key: 'hasLiedBrowser', getData: hasLiedBrowserKey },
  { key: 'touchSupport', getData: touchSupportKey },
  { key: 'fonts', getData: jsFontsKey, pauseBefore: true },
  { key: 'fontsFlash', getData: flashFontsKey, pauseBefore: true },
  { key: 'audio', getData: audioKey },
  { key: 'enumerateDevices', getData: enumerateDevicesKey }
]
```

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

- solutions?

```
- canvas element (javascript)
+
|
+----+ user-permission <naïve users (sorry Tor)>
|
|
+-------+ rules (what should the rules be) <false positives>
|
|
+----------+ block all (altered canvas) <degraded user experience>
```

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

- solutions?

```
- canvas element (javascript)


|
|
+-------+ rules (what should the rules be) <false positives>
                                             <stagnant>
```

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

- false positive problem: dual use technologies
  - canvas actions must be delineated ==========>

1. The canvas element's `height` and `width` properties must not be set below 16 px.[12]
2. Text must be written to canvas with least two colors or at least 10 distinct characters.
3. The script should not call the `save`, `restore`, or `addEventListener` methods of the rendering context.
4. The script extracts an image with `toDataURL` or with a single call to `getImageData` that specifies an area with a minimum size of 16px × 16px.

Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu

Online tracking: A 1-million-site measurement and analysis is the largest and most detailed measurement of online tracking to date. We measure stateful (cookie-based) and stateless (fingerprinting-based) tracking, the effect of browser privacy tools, and "cookie syncing".

This measurement is made possible by our web measurement tool OpenWPM, a mature platform that enables fully automated web crawls using a full-fledged and instrumented browser.

Read the paper » Go to Project Home »

Sites with canvas fingerprinting scripts

In a crawl conducted during January 2016, these websites were found to run scripts on their homepages that used the Canvas API to fingerprint users.

Show 100 entries Search: todayifound

Showing 1 to 2 of 2 entries (filtered from 15,089 total entries)

Previous Next

Alexa Rank	Site URL	Fingerprinting Domain
8917	http://todayifoundout.com	lijit.com
8917	http://todayifoundout.com	doubleverify.com

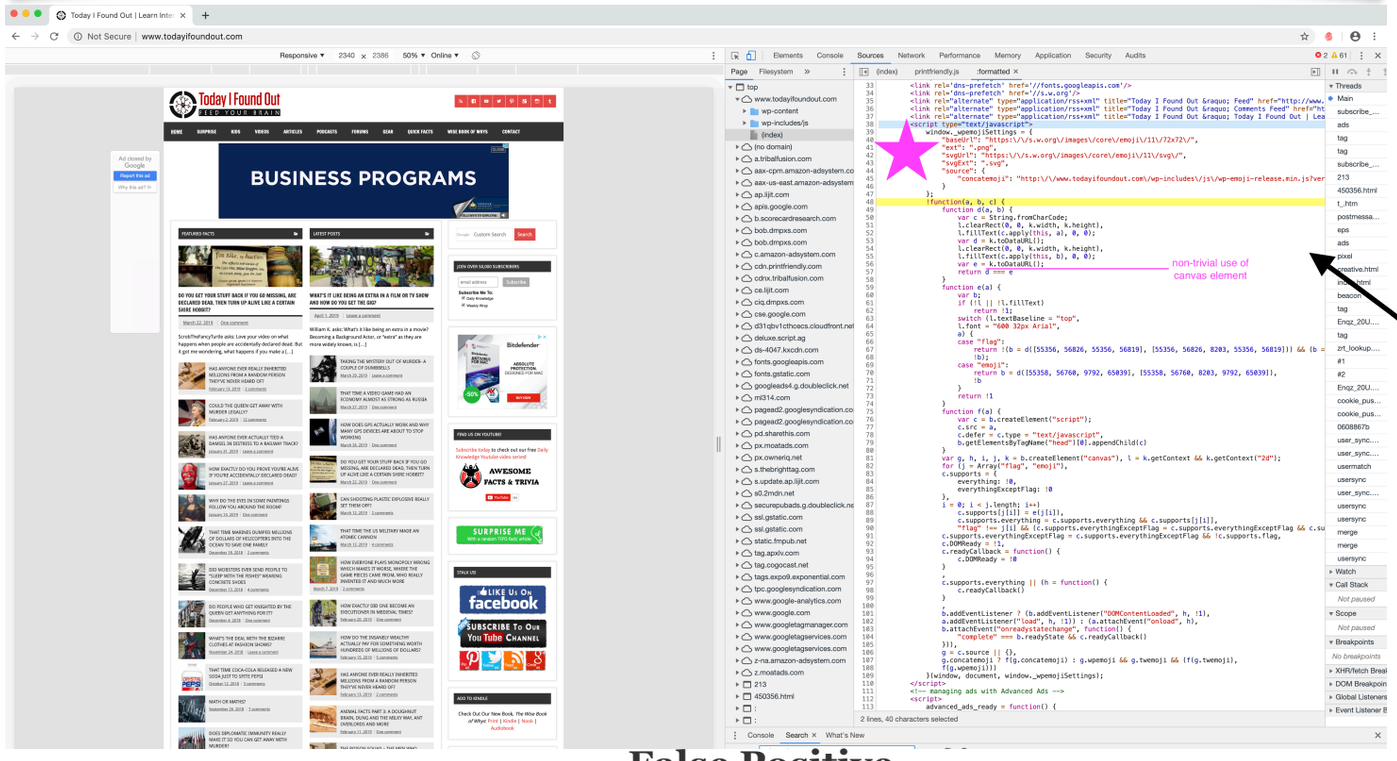
Background

privacy: is there a tracking problem?

Online Tracking:
A 1-million-site Measurement and Analysis

Steven Englehardt
Princeton University
ste@cs.princeton.edu

Arvind Narayanan
Princeton University
arvindn@cs.princeton.edu



False Positive

https://github.com/ghostwords/chameleon/issues/20
https://core.trac.wordpress.org/ticket/43264
https://www.thesafemac.com/tor-browser-false-positive/

Background

privacy: is there a tracking problem?

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2017 (1):79–99

Muhammad Ikram*, Hassan Jameel Asghar, Mohamed Ali Kaafar, Anirban Mahanti, and Balachandar Krishnamurthy

Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-class Learning

```
- canvas element (javascript)



|
|
+-------+ rules
```

let’s use ML to tune the rules!

Background

apples to apples

- Problem: Tracking
  - Solution: Turn off Javascript
    - Problem: Broken functionality
      - Solution: ad-block with regex-styled string matching
        (1) NoScript: default is to block javascript, Silverlight, flash (users may whitelist)
        (2) AdBlock Plus: blacklists, searches rendered DOM tree (HTML) with regex and blocks requests to download content per blacklist
        (3) Disconnect: blacklists, similar to AdBlock Plus
        (4) Ghostery: blacklist, similar to AdBlock Plus. Also disables cookies
        (5) Privacy Badger: blacklist, similar to AdBlock Plus. Also blocks code that attempts to read cookies (high entropy cookies)
        - Problem: Ineffective (false positives and broken functionality)

Background

apples to apples

- intuition: tracking code has similar structure

Tracker 1. Google Analytics Cookie Setting

```
var _gaq = _gaq || [];
_gaq.push(['_setAccount', 'UA-1627489-1']);
_gaq.push(['_setDomainName', 'geo.tv']);
_gaq.push(['_trackPageview']);
```

Tracker 2. Visual Revenue Cookie Setting

```
var _vrq = _vrq || [],
_vrqIsOnHP = (document.body.className ||
  '').search('pg-section') >=0 ? true : false;
_vrq.push(['id', 396]);
_vrq.push(['automate', _vrqIsOnHP]);
_vrq.push(['track', function() {}]);
```

The `_gaq` object is what makes the asynchronous syntax possible. It acts as a queue, which is a first-in,first-out data structure that collects API calls until `ga.js` is ready to execute them. To add something to the queue, use the `_gaq.push` method.

Background

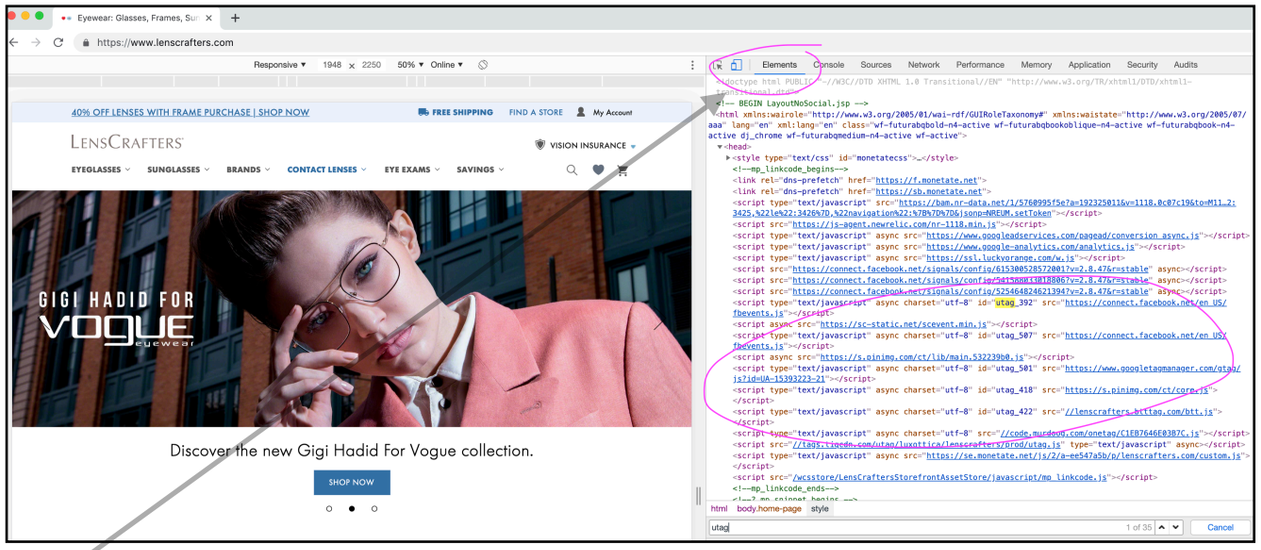
apples to apples

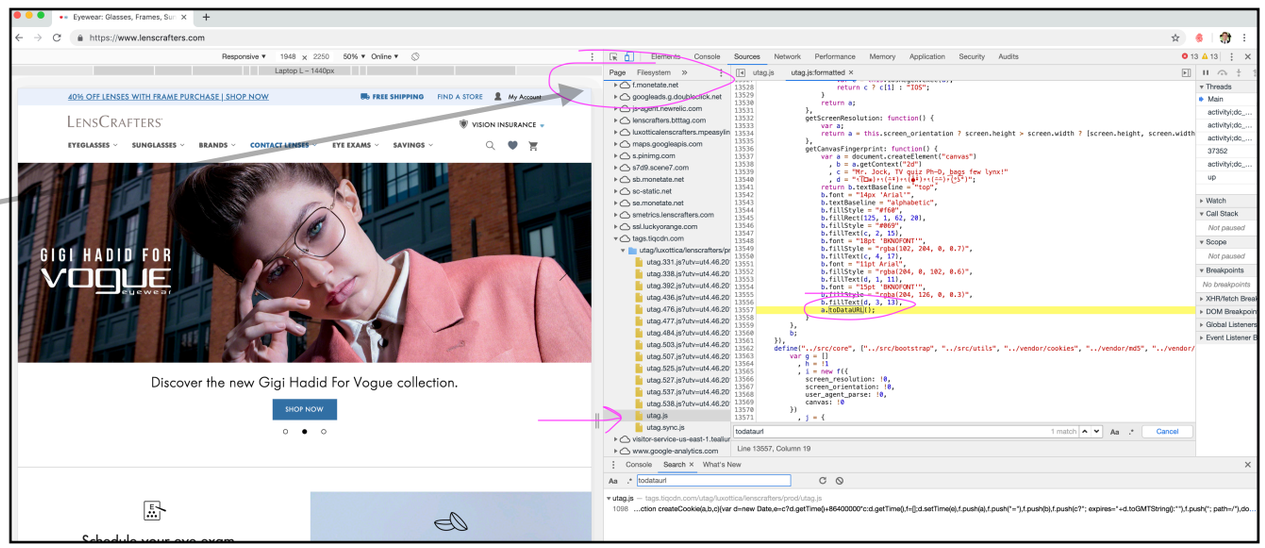
- Solution: Semantic Similarity
  - Main Intuition: “tracking” code is functionally and structurally similar to other tracking code, and different from non-tracking (“functional”) code

Experiments

the scrape

- Selenium (180 seconds per domain)
- *process*
  - visit 95 websites (2612 programs)
    50 Alexa Top Sites
    45 random
  - store DOM tree
    - parse script tags (in-page)
    - remote content (external)
  - repeat *process* with and without adblockers (set)





Experiments

manual labeling

- Defining a tracker
  - R7, R8 — useful functionality: “Facilitate access to contents and services related to the target (visited) webpage”
    - “web-pages contain JavaScript programs that enable search boxes, accessibility options, authentication services, shopping carts, prompts, navigation menu and breadcrumbs”
    - “we created a manual list of well-known third-party CDNs to differentiate them from other content providers”
- if tie == assume tracking

Rule	JS	#	Description
R1	✗	216	All JS that create panels and set margins for ads
R2	✗	115	All JS that access and display ads
R3	✗	45	All social media widgets
R4	✗	324	All in-page JS that include external JS from third-party analytics and advertisers
R5	✗	353	All external JS from third-party analytics and advertisers
R6	✗	180	All cookie enablers, readers or writers
R7	✓	542	All external JS that provide useful functionality such as navigation menus, search and login
R8	✓	509	All in-page JS that provide useful functionality
R9	✓	132	All JS that fetch content from first-party content domains or third-party CDNs
R10	✗	103	All JS in hidden iframe that belong to third-party analytics, advertisers and social media
R11	✗	40	All JS in hidden iframe that enable, read or modify cookies
R12	✓	53	All JS that track mouse or keyboard events

Table 1. Rules for labelling JavaScript programs - R stands for Rule; JS stands for JavaScript program; # denotes the number of JavaScript programs satisfying the corresponding rule in the labelled dataset; ✗ represents tracking JavaScript programs and ✓ represents functional JavaScript programs.

Experiments

manual labeling

- notable: a single expert evaluated all programs

- notable: interestingly good split between tracking and not tracking

overview

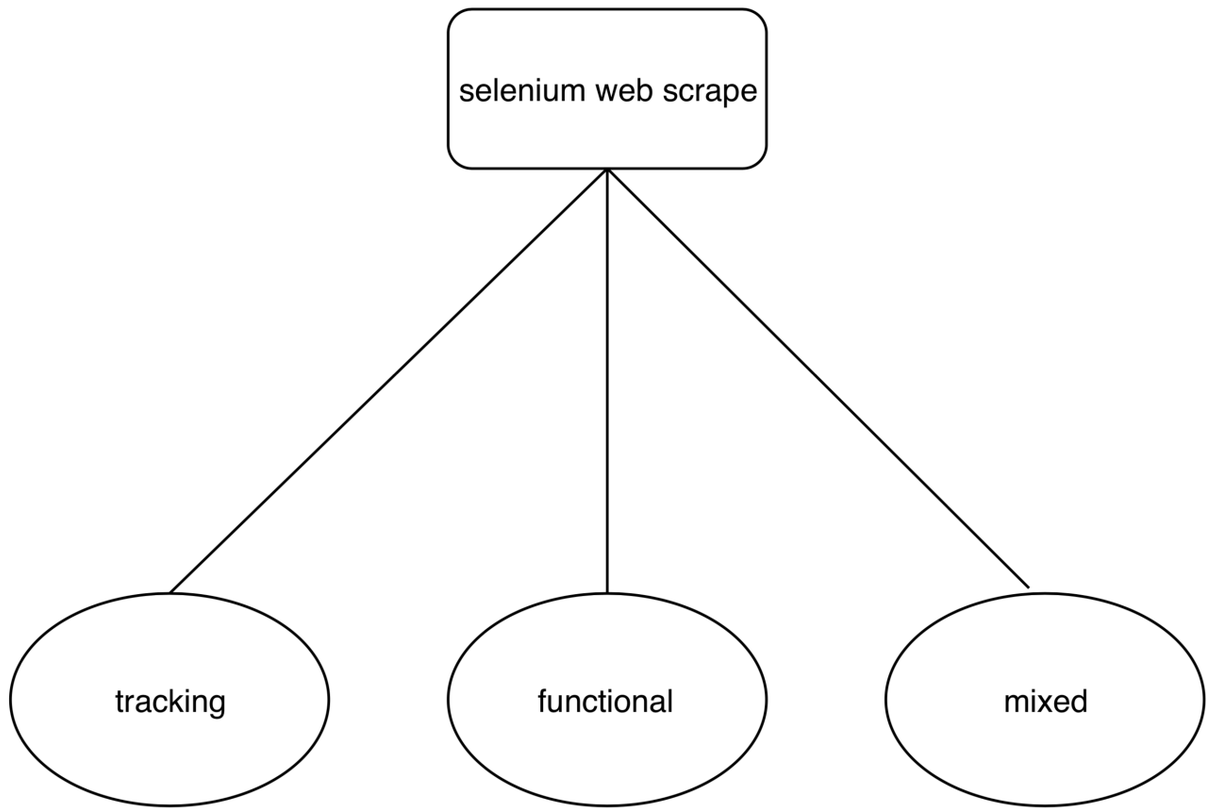
JavaScript programs					
External	In-page	Average	Total	Tracking	Functional
1,353	1,256	27.5	2,612	57%	43%

Table 2. Characteristics of JavaScript programs from 95 websites in our labelled dataset.

Experiments

what do we have?

- manually labeled groupings of tracking, functional, and mixed programs



Experiments

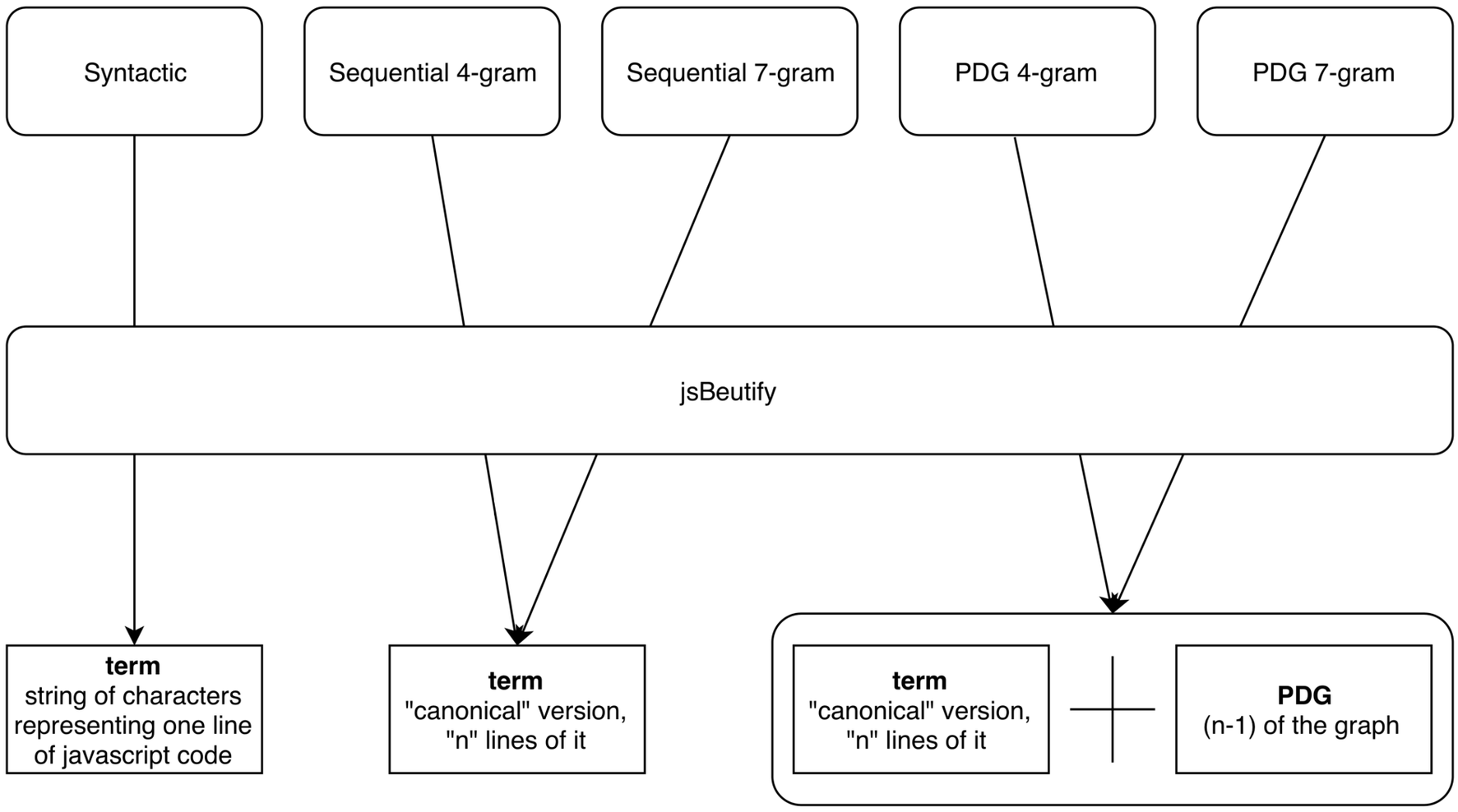
where are we going

- train a one-class support vector machine (SVM) to identify and predict these groupings, given that we have a small set of labeled data

Experiments

what do we need: semantic similarity

- how to represent text



Experiments

jsBeautify

```
function fingerprint() {
      var canvas = document.createElement('canvas');
      var ctx = canvas.getContext('2d');
      var txt = 'i9asdm..$#po((^@KbXrww!~cz';
      ctx.textBaseline = "top";
      ctx.font = "16px 'Arial'";
      ctx.textBaseline = "alphabetic";
      ctx.rotate(.07);
      ctx.fillStyle = "#f60";
      ctx.fillRect(125,1,62,20);
      ctx.fillStyle = "#069";
      ctx.fillText(txt, 2, 15);
      ctx.fillStyle = "rgba(102, 200, 0, 0.7)";
      ctx.fillText(txt, 4, 17);
      ctx.shadowBlur=10;
      ctx.shadowColor="blue";
      ctx.fillRect(-20,10,234,5);
      var strng=canvas.toDataURL();

document.body.appendChild(canvas);

}

$('#number').html(fingerprint());
/* $('#MimeType').html(navigator.mimeTypes[4].type);
$('#colorDepth').html(screen.colorDepth);
$('#pixelDepth').html(screen.pixelDepth);
 */
console.log(screen);
```

original

```
var _0xb786=
["\x63\x61\x6E\x76\x61\x73","\x63\x72\x65\x61\x74\x65\x45\x6C\x65\x6D\x65\x6E\x74","\x32\x64","\x67\x65\x74\x43\x6F\x6E\x74\x65\x78\x74","\x69\x39\x61\x73\x64\x6D\x2E\x2E\x24\x23\x70\x6F\x28\x28\x5E\x40\x4B\x62\x58\x72\x77\x77\x21\x7E\x63\x7A","\x74\x65\x78\x74\x42\x61\x73\x65\x6C\x69\x6E\x65","\x74\x6F\x70","\x66\x6F\x6E\x74","\x31\x36\x70\x78\x20\x27\x41\x72\x69\x61\x6C\x27","\x61\x6C\x70\x68\x61\x62\x65\x74\x69\x63","\x72\x6F\x74\x61\x74\x65","\x66\x69\x6C\x6C\x53\x74\x79\x6C\x65","\x23\x66\x36\x30","\x66\x69\x6C\x6C\x52\x65\x63\x74","\x23\x30\x36\x39","\x66\x69\x6C\x6C\x54\x65\x78\x74","\x72\x67\x62\x61\x28\x31\x30\x32\x2C\x20\x32\x30\x30\x2C\x20\x30\x2C\x20\x30\x2E\x37\x29","\x73\x68\x61\x64\x6F\x77\x42\x6C\x75\x72","\x73\x68\x61\x64\x6F\x77\x43\x6F\x6C\x6F\x72","\x62\x6C\x75\x65","\x74\x6F\x44\x61\x74\x61\x55\x52\x4C","\x61\x70\x70\x65\x6E\x64\x43\x68\x69\x6C\x64","\x62\x6F\x64\x79"];function fingerprint(){var _0xa3cfx2=document[_0xb786[1]](_0xb786[0]);var _0xa3cfx3=_0xa3cfx2[_0xb786[3]](_0xb786[2]);var _0xa3cfx4=_0xb786[4];_0xa3cfx3[_0xb786[5]]= _0xb786[6];_0xa3cfx3[_0xb786[7]]= _0xb786[8];_0xa3cfx3[_0xb786[5]]= _0xb786[9];_0xa3cfx3[_0xb786[10]](0.07);_0xa3cfx3[_0xb786[11]]= _0xb786[12];_0xa3cfx3[_0xb786[13]](125,1,62,20);_0xa3cfx3[_0xb786[11]]= _0xb786[14];_0xa3cfx3[_0xb786[15]](_0xa3cfx4,2,15);_0xa3cfx3[_0xb786[11]]= _0xb786[16];_0xa3cfx3[_0xb786[15]](_0xa3cfx4,4,17);_0xa3cfx3[_0xb786[17]]= 10;_0xa3cfx3[_0xb786[18]]= _0xb786[19];_0xa3cfx3[_0xb786[13]](-20,10,234,5);var _0xa3cfx5=_0xa3cfx2[_0xb786[20]]();document[_0xb786[22]][_0xb786[21]](_0xa3cfx2)}

$('#number').html(fingerprint());
/* $('#MimeType').html(navigator.mimeTypes[4].type);
$('#colorDepth').html(screen.colorDepth);
$('#pixelDepth').html(screen.pixelDepth);
 */
console.log(screen);
```

obfuscated

Experiments

jsBeautify

```
var _0xb786=
["\x63\x61\x6E\x76\x61\x73","\x63\x72\x65\x61\x74\x65\x45\x6C\x65\x6D\x65\x6E\x74","\x32\x64","\x67\x65\x74\x43\x6F\x6E\x74\x65\x78\x74","\x69\x39\x61\x73\x64\x6D\x2E\x2E\x24\x23\x70\x6F\x28\x28\x5E\x40\x4B\x62\x58\x72\x77\x77\x21\x7E\x63\x7A","\x74\x65\x78\x74\x42\x61\x73\x65\x6C\x69\x6E\x65","\x74\x6F\x70","\x66\x6F\x6E\x74","\x31\x36\x70\x78\x20\x27\x41\x72\x69\x61\x6C\x27","\x61\x6C\x70\x68\x61\x62\x65\x74\x69\x63","\x72\x6F\x74\x61\x74\x65","\x66\x69\x6C\x6C\x53\x74\x79\x6C\x65","\x23\x66\x36\x30","\x66\x69\x6C\x6C\x52\x65\x63\x74","\x23\x30\x36\x39","\x66\x69\x6C\x6C\x54\x65\x78\x74","\x72\x67\x62\x61\x28\x31\x30\x32\x2C\x20\x32\x30\x30\x2C\x20\x30\x2C\x20\x30\x2E\x37\x29","\x73\x68\x61\x64\x6F\x77\x42\x6C\x75\x72","\x73\x68\x61\x64\x6F\x77\x43\x6F\x6C\x6F\x72","\x62\x6C\x75\x65","\x74\x6F\x44\x61\x74\x61\x55\x52\x4C","\x61\x70\x70\x65\x6E\x64\x43\x68\x69\x6C\x64","\x62\x6F\x64\x79"];function fingerprint(){var _0xa3cfx2=document[_0xb786[1]](_0xb786[0]);var _0xa3cfx3=_0xa3cfx2[_0xb786[3]](_0xb786[2]);var _0xa3cfx4=_0xb786[4];_0xa3cfx3[_0xb786[5]]=_0xb786[6];_0xa3cfx3[_0xb786[7]]= _0xb786[8];_0xa3cfx3[_0xb786[5]]=_0xb786[9];_0xa3cfx3[_0xb786[10]](0.07);_0xa3cfx3[_0xb786[11]]=_0xb786[12];_0xa3cfx3[_0xb786[13]](125,1,62,20);_0xa3cfx3[_0xb786[11]]=_0xb786[14];_0xa3cfx3[_0xb786[15]](_0xa3cfx4,2,15);_0xa3cfx3[_0xb786[11]]=_0xb786[16];_0xa3cfx3[_0xb786[15]](_0xa3cfx4,4,17);_0xa3cfx3[_0xb786[17]]=10;_0xa3cfx3[_0xb786[18]]= _0xb786[19];_0xa3cfx3[_0xb786[13]](-20,10,234,5);var _0xa3cfx5=_0xa3cfx2[_0xb786[20]]();document[_0xb786[22]][_0xb786[21]](_0xa3cfx2)}

$('#number').html(fingerprint());
/* $('#MimeType').html(navigator.mimeTypes[4].type);
$('#colorDepth').html(screen.colorDepth);
$('#pixelDepth').html(screen.pixelDepth);
 */
console.log(screen);
```

obfuscated

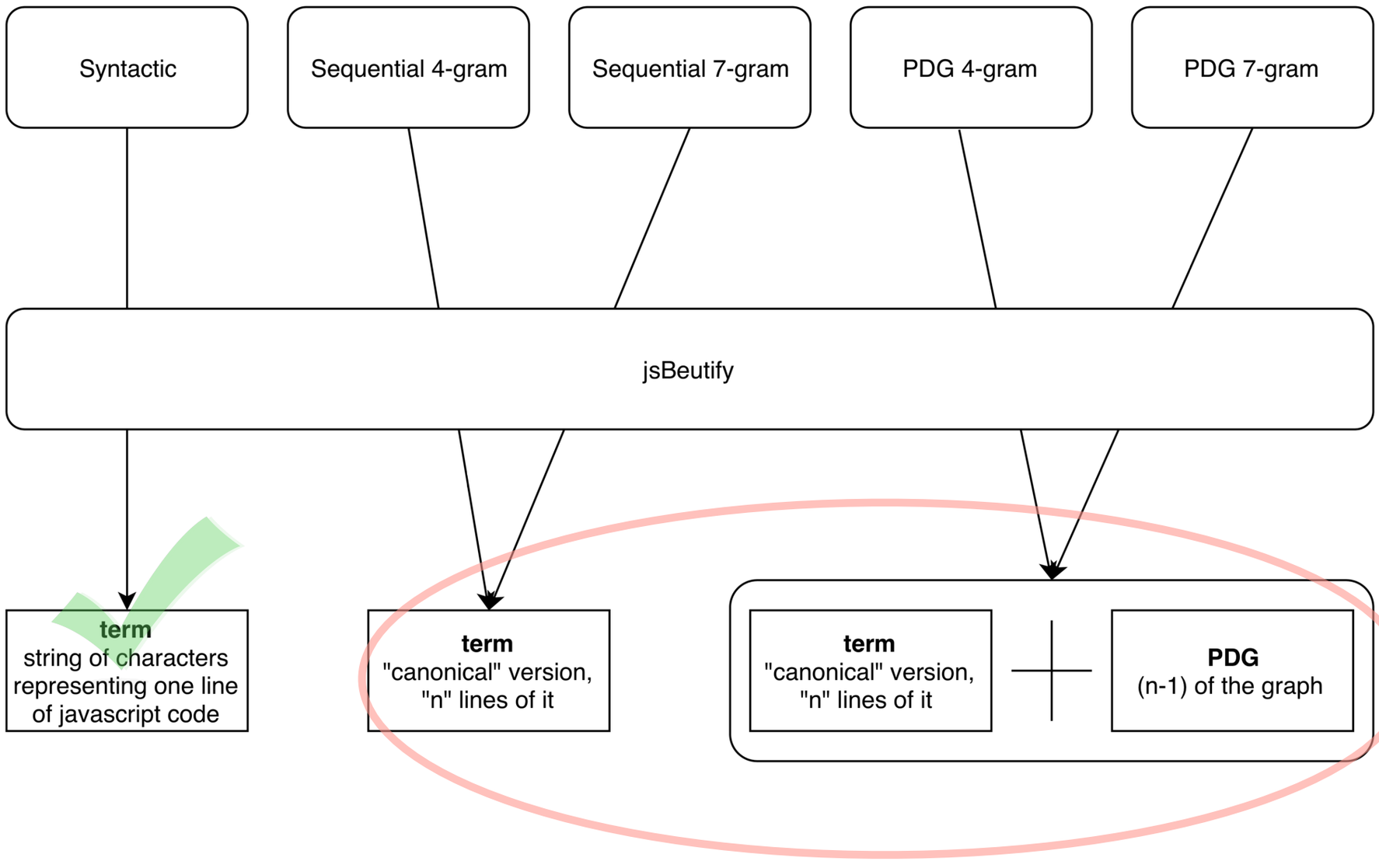
```
function fingerprint() {
    var _0xa3cfx2 = document['createElement']('canvas');
    var _0xa3cfx3 = _0xa3cfx2['getContext']('2d');
    var _0xa3cfx4 = 'i9asdm..$#po((^@KbXrww!~cz';
    _0xa3cfx3['textBaseline'] = 'top';
    _0xa3cfx3['font'] = '16px \'Arial\'';
    _0xa3cfx3['textBaseline'] = 'alphabetic';
    _0xa3cfx3['rotate'](0.07);
    _0xa3cfx3['fillStyle'] = '#f60';
    _0xa3cfx3['fillRect'](125, 1, 62, 20);
    _0xa3cfx3['fillStyle'] = '#069';
    _0xa3cfx3['fillText'](_0xa3cfx4, 2, 15);
    _0xa3cfx3['fillStyle'] = 'rgba(102, 200, 0, 0.7)';
    _0xa3cfx3['fillText'](_0xa3cfx4, 4, 17);
    _0xa3cfx3['shadowBlur'] = 10;
    _0xa3cfx3['shadowColor'] = 'blue';
    _0xa3cfx3['fillRect'](-20, 10, 234, 5);
    var _0xa3cfx5 = _0xa3cfx2['toDataURL']();
    document['body']['appendChild'](_0xa3cfx2)
}

$('#number').html(fingerprint());
/* $('#MimeType').html(navigator.mimeTypes[4].type);
$('#colorDepth').html(screen.colorDepth);
$('#pixelDepth').html(screen.pixelDepth);
 */
console.log(screen);
```

jsBeautify

Experiments

semantic similarity: likeness between programs

- how to represent text



Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

secondary paper

why does tf-idf work on text

- “The context of a token is reasonably captured by the preceding words, and the text tokens are different enough to have distinctive distributions, but common enough that a single text token can be observed multiple times.”

- programs do not behave like this

Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

why is this important

- finding “important” pieces of code is a non-trivial task—tf-idf does not work on code-as-text

- e.g., plagiarism false positives (a word-for-word copy of a trivial section of code should not be considered plagiarism)

Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

example

- when Oracle sued Google back in 2010 for copyright violations (asking for 8.8 billion in damages) the case, in part, came down to 9 lines of code—out of 2.86 million lines—which were copied verbatim

Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

example

```
1 private static void rangeCheck(int arrayLen, int fromIndex, int toIndex {
2       if (fromIndex > toIndex)
3             throw new IllegalArgumentException("fromIndex(" + fromIndex +
4                   ") > toIndex(" + toIndex+")");
5       if (fromIndex < 0)
6             throw new ArrayIndexOutOfBoundsException(fromIndex);
7       if (toIndex > arrayLen)
8             throw new ArrayIndexOutOfBoundsException(toIndex);
9   }
```

Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

example

3. Has Oracle proven that Google's conceded use of the following was infringing,

- the only issue being whether such use was de minimis:

	Yes (Infringing)	No (Not Infringing)
A. The rangeCheck method in TimSort.java and ComparableTimSort.Java	✓	
B. Source code in seven "Impl.java" files and the one "ACL" file		✓
C. The English-language comments in CodeSourceTest.java and CollectionCertStoreParameters Test.java		✓

Semantic Similarity

likeness between programs

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

tf-idf for programs

- 1) canonical form
- 2) program dependency graph (PDG)
- 3) tf-idf with “tokens” of n-gram PDGs

Semantic Similarity

1) canonical form: original text

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- original program
  - is val in array

```
function inArray(a, val) {
  var i;
  for (i = 0; i < a.length; i++) {
    if (a[i] === val) {
      return true;
    }
  }
  return false;
}
```

Semantic Similarity

1) canonical form: three-address code

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- **three address code**
  - used by compilers
  - expression is assignment or binary operator or combination of assignment and binary operator

GivenExpression:

```
a := (-c * b) + (-c * d)
```

Three-address code is as follows:

```
t1 := -c
t2 := b*t1
t3 := -c
t4 := d * t3
t5 := t2 + t4
a := t5
```

t is used as registers in the target program.

Semantic Similarity

1) canonical form: three-address code

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
function inArray(a, val) {
  var i;
  for (i = 0; i < a.length; i++) {
    if (a[i] === val) {
      return true;
    }
  }
  return false;
}
```

→

```
function inArray(a, val) {
1   begin;
2   i = 0;
3   $0 = a.length;
4   $1 = i < $0;
5   while ($1) {
6     $2 = a[i];
7     $3 = $2 === val;
8     if ($3) {
9       return true;
      }
10    i = i + 1;
11    $4 = a.length;
12    $1 = i < $4;
    }
13  return false;
14  end;
}
```

Semantic Similarity

1) canonical form: formal specification

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

The canonical transformation function χ takes a JavaScript statement or an expression as input, and transforms it into a pair $(val, stmt^*)$, where $stmt^*$ is a list of canonical statements that describes the functionality of the input statement or expression, and val holds the result of the statement or expression.

$$
\begin{aligned}
func &\rightarrow \texttt{function}\ id^?\ \texttt{(}\ var^*\ \texttt{)}\ \{\ \texttt{begin;}\ stmt^*\ \texttt{end;}\ \}\\
stmt &\rightarrow assign\ \texttt{;} \mid \texttt{break;} \mid \texttt{continue;} \mid \texttt{return}\ val^?\ \texttt{;}\\
&\mid \texttt{if (}\ val\ \texttt{)}\ \{\ stmt^*\ \}\ \texttt{else}\ \{\ stmt^*\ \}\\
&\mid \texttt{while (}\ val\ \texttt{)}\ \{\ stmt^*\ \}\\
&\mid \texttt{for (}\ var\ \texttt{in}\ val\ \texttt{)}\ \{\ stmt^*\ \}\\
&\mid \texttt{switch (}\ val\ \texttt{)}\ \{\\
&\quad (\texttt{case}\ val\texttt{:}\ stmt^*)^*\ (\texttt{default:}\ stmt^*)^?\}\\
&\mid \texttt{with (}\ val\ \texttt{)}\ \{\ stmt^*\ \}\\
assign &\rightarrow var\ \texttt{=}\ val \mid var\ \texttt{=}\ op_{\text{unary}}\ val \mid var\ \texttt{=}\ val\ op_{\text{binary}}\ val\\
&\mid var\ \texttt{=}\ val\ \texttt{?}\ val\ \texttt{:}\ val \mid var\ \texttt{=}\ val\texttt{.}identifier\\
&\mid var\ \texttt{=}\ val\texttt{[}val\texttt{]} \mid var\ \texttt{=}\ identifier\texttt{(}val^*\texttt{)}\\
&\mid var\ \texttt{=}\ \texttt{(}func\texttt{)(}val^*\texttt{)} \mid var\ \texttt{=}\ val\texttt{.}identifier\texttt{(}val^*\texttt{)}\\
&\mid var\texttt{.}identifier\ \texttt{=}\ val \mid var\texttt{[}val\texttt{]}\ \texttt{=}\ val\\
val &\rightarrow var \mid literal \mid func
\end{aligned}
$$

Figure 3: The canonical form's formal definition. S^* means that S appears at 0 or more times, and $S^?$ means that S appears at most once.

Semantic Similarity

1) canonical form: formal specification

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

$$\frac{\begin{array}{c} expr \rightarrow expr_1 \ op \ expr_2 \\ (val_1, S_1) = \chi(expr_1) \\ (val_2, S_2) = \chi(expr_2) \\ var = \texttt{NewTemp()} \end{array}}{\chi(expr) = (var, \langle S_1, S_2, var \texttt{ = } val_1 \ op \ val_2 \texttt{;} \rangle)}$$

The first line specifies the context-free reduction rule that is used to parse the expression. In this case, it says that the above rule is applied when the input expression is a binary operation. The remaining equations above the horizontal line are the preconditions for the rule, and post-condition of the transformation rule is listed below the line. So the above rule states that if $expr_1$ is transformed into (val_1, S_1) and $expr_2$ is transformed into (val_2, S_2), and if we create a temporary variable var through the `NewTemp()` special function, then the resulting canonical statement consists of S_1 and S_2, followed by the statement that assigns the results of $expr_1 \, op \, expr_2$ into var.

Semantic Similarity

2) PDG

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- all edges signify either control or data dependency (lecture 2-7-19)
  - data dependency: value *a* affects value *b* (e.g., line 6 to line 2)
  - control dependency: “if” or “while” statements (e.g., line 7 to line 5)

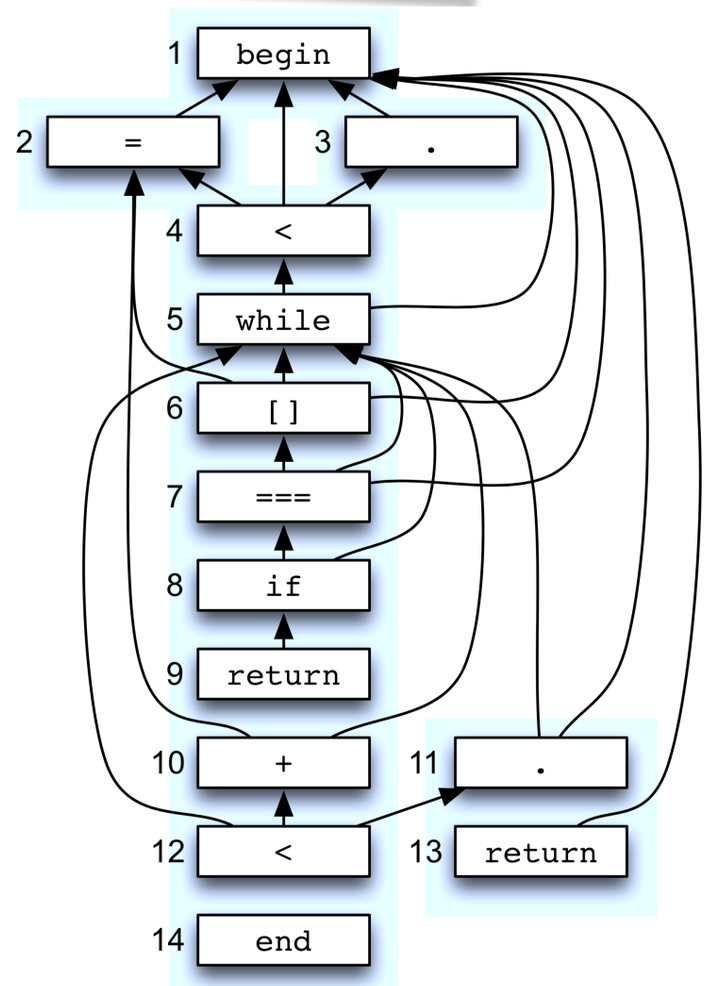
Semantic Similarity

2) PDG

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
  function inArray(a, val) {
1   begin;
2   i = 0;
3   $0 = a.length;
4   $1 = i < $0;
5   while ($1) {
6     $2 = a[i];
7     $3 = $2 === val;
8     if ($3) {
9       return true;
      }
10    i = i + 1;
11    $4 = a.length;
12    $1 = i < $4;
    }
13  return false;
14  end;
  }
```



Semantic Similarity

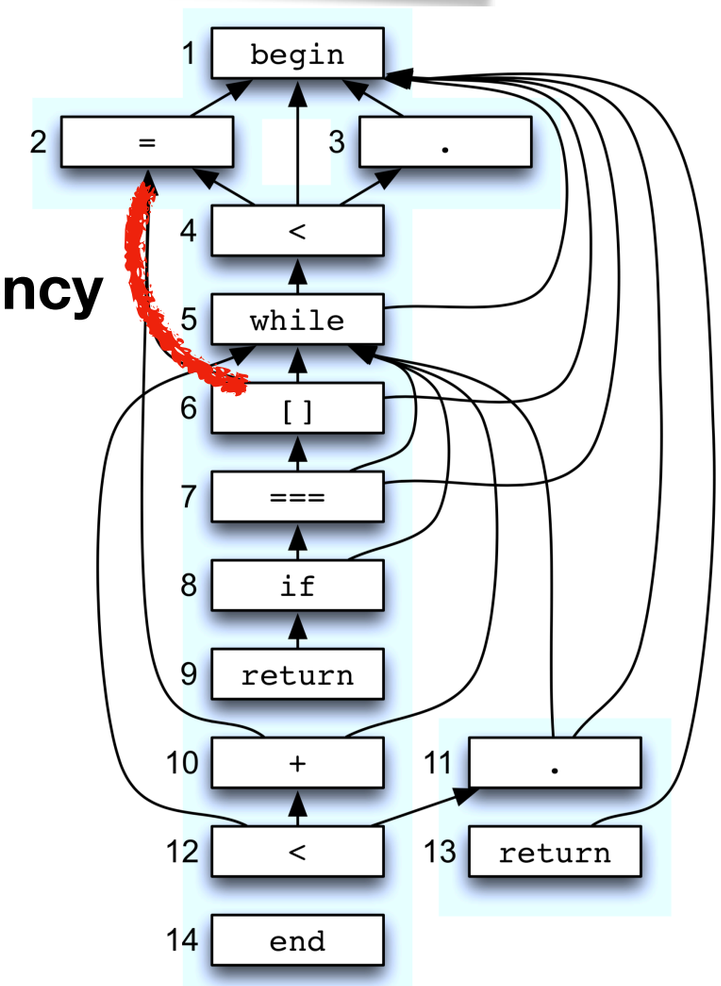
2) PDG

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
  function inArray(a, val) {
1   begin;
2   i = 0;
3   $0 = a.length;
4   $1 = i < $0;
5   while ($1) {
6     $2 = a[i];
7     $3 = $2 === val;
8     if ($3) {
9       return true;
      }
10    i = i + 1;
11    $4 = a.length;
12    $1 = i < $4;
    }
13  return false;
14  end;
  }
```

data dependency



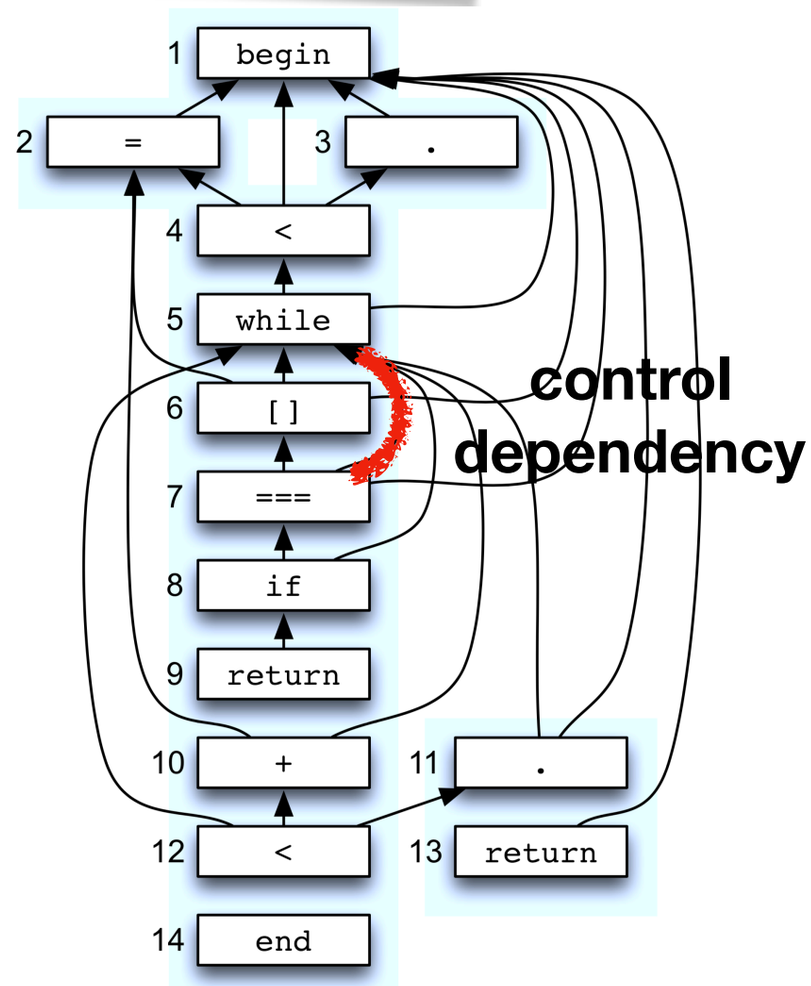
Semantic Similarity

2) PDG

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
  function inArray(a, val) {
1   begin;
2   i = 0;
3   $0 = a.length;
4   $1 = i < $0;
5   while ($1) {
6     $2 = a[i];
7     $3 = $2 === val;
8     if ($3) {
9       return true;
      }
10    i = i + 1;
11    $4 = a.length;
12    $1 = i < $4;
    }
13  return false;
14  end;
  }
```

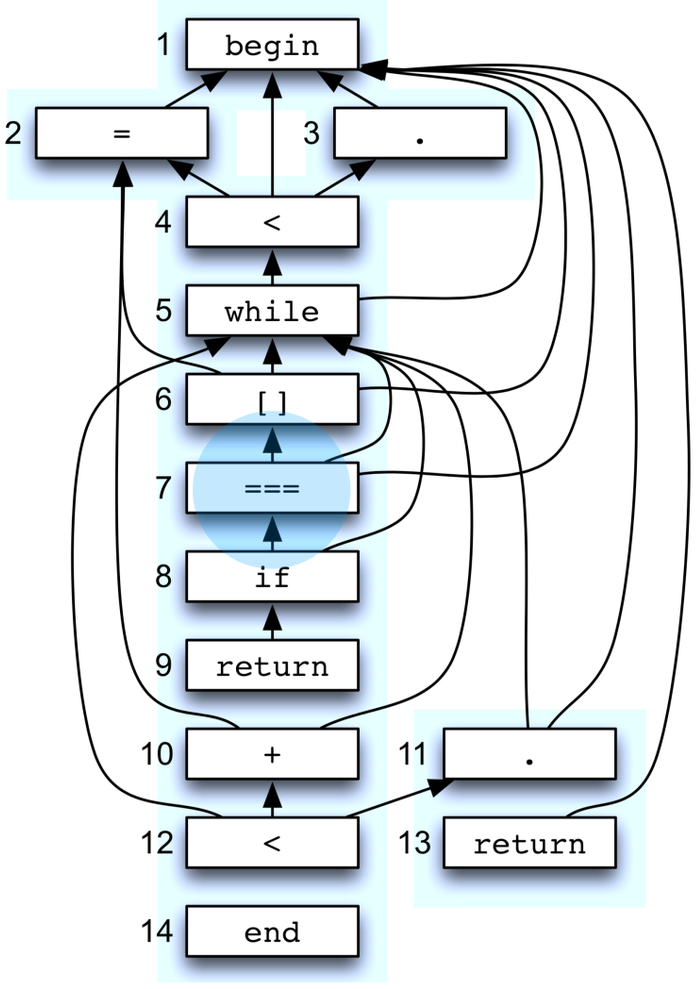


Semantic Similarity

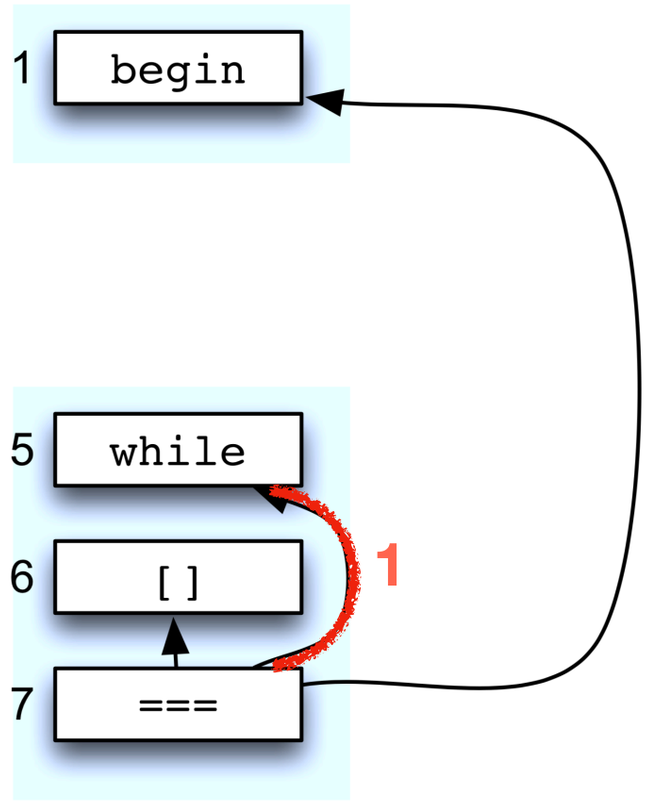
2) PDG "n" grams

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu



2 gram

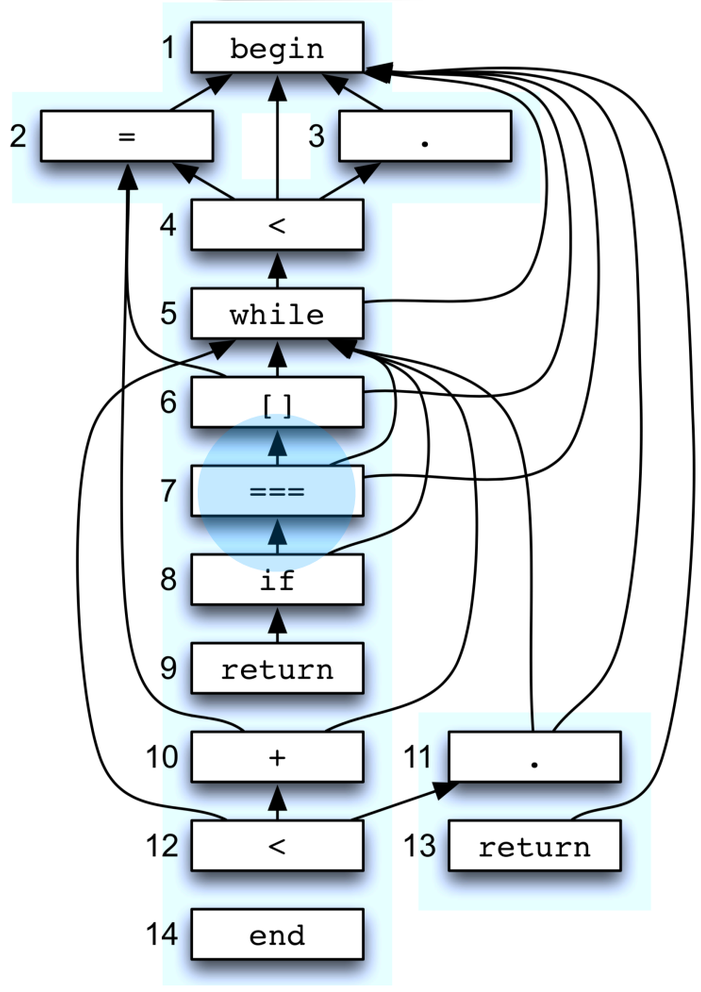


Semantic Similarity

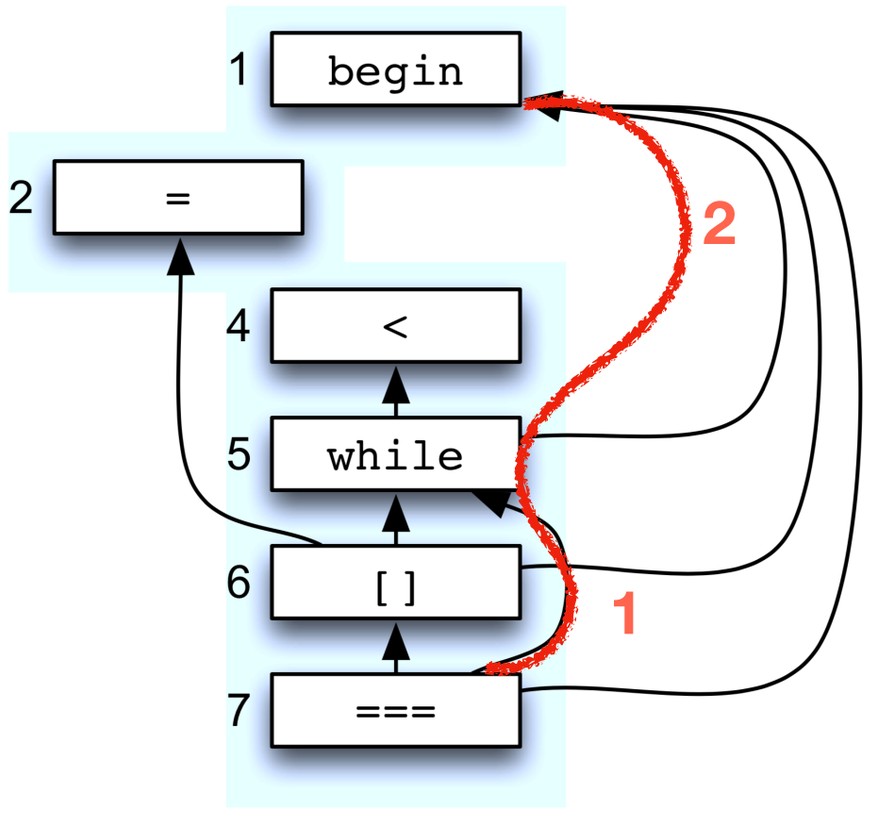
2) PDG "n" grams

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu



3 gram



Semantic Similarity

2) PDG "n" grams

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- n-gram is a labeled sub-graph of the program dependency graph constructed over the canonical form

- subgraph consists of all paths of length (n-1) starting from a specific statement

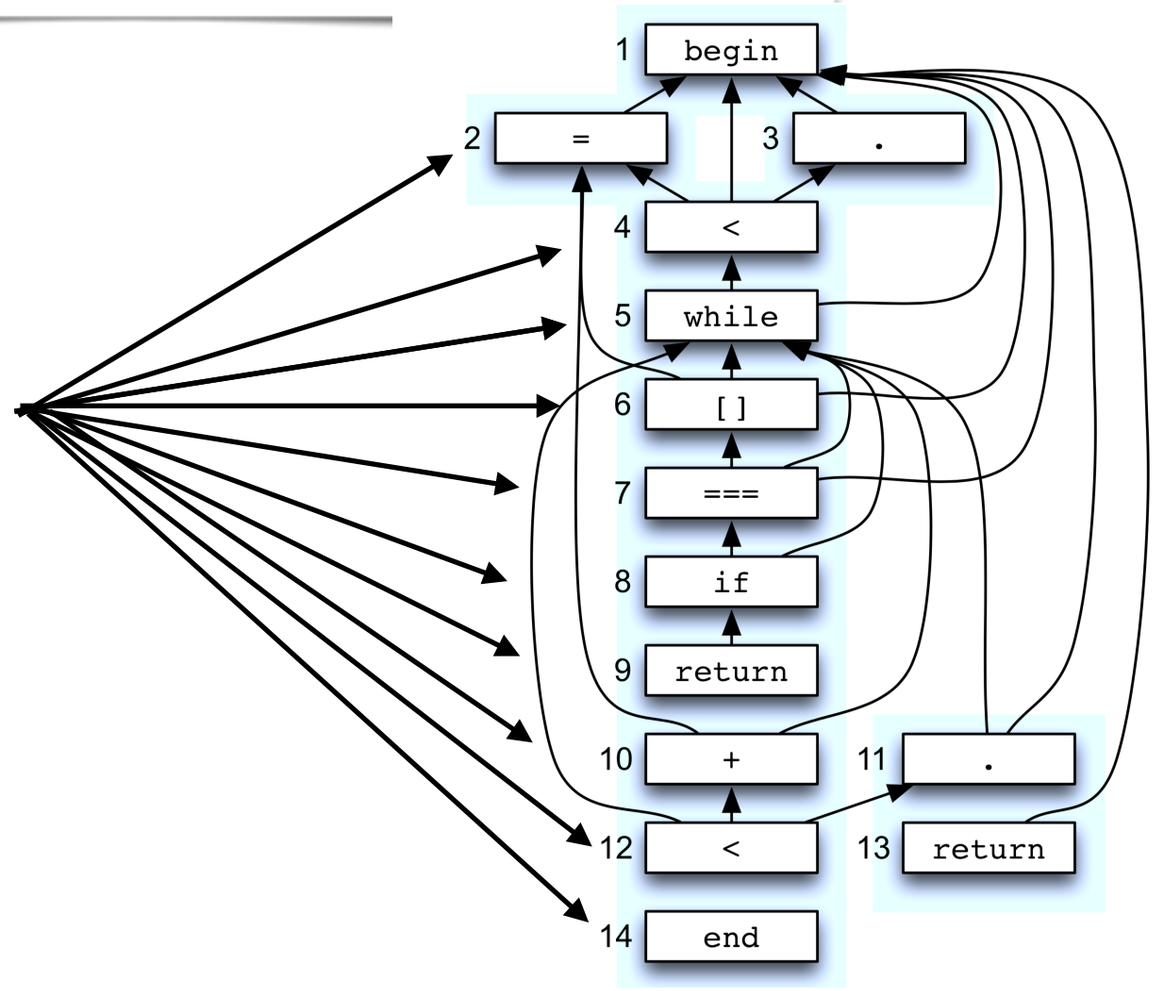
Semantic Similarity

3) tf-idf: importance

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- problem: but how do we know which line to use in the n-gram



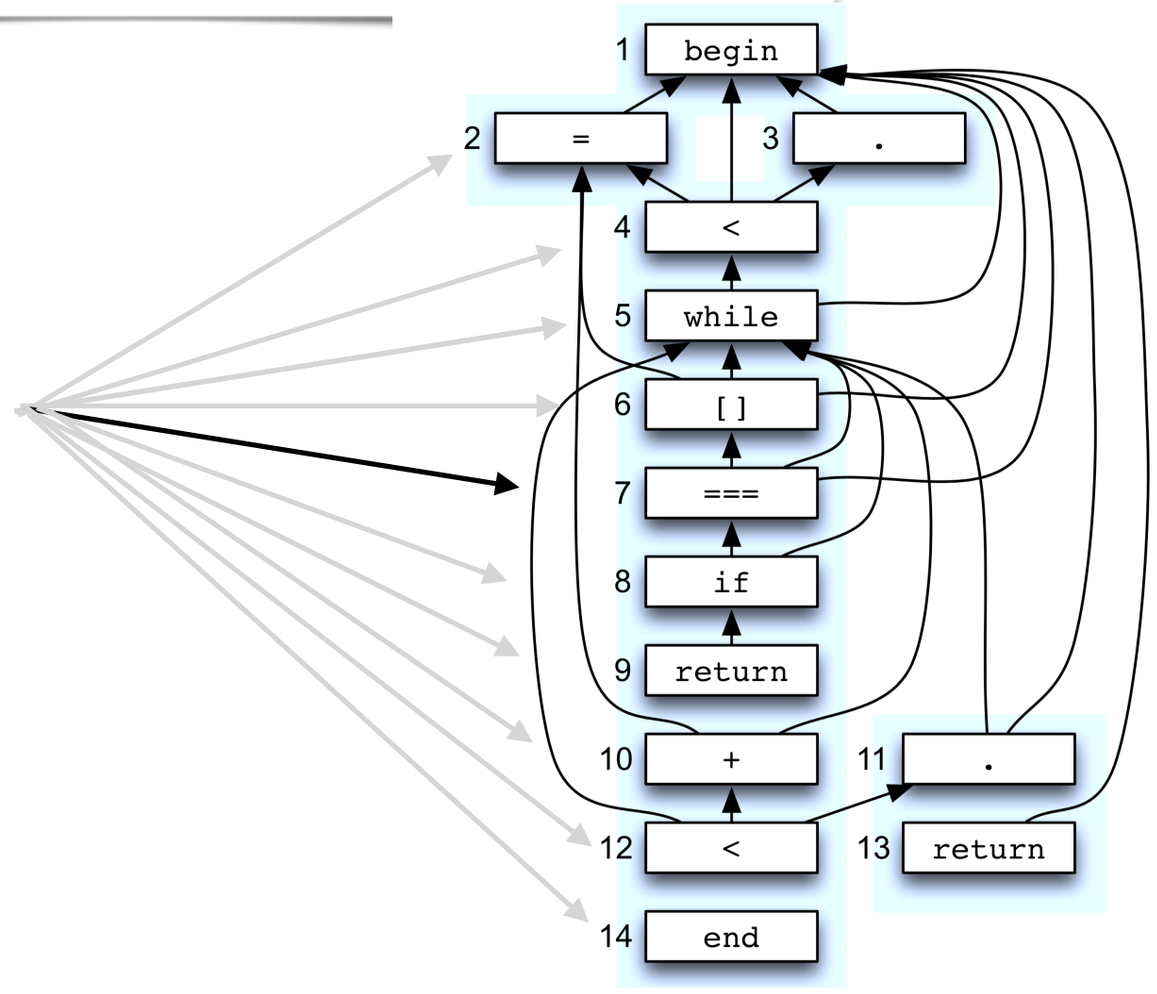
Semantic Similarity

3) tf-idf: importance

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- answer: tf-idf



Semantic Similarity

3) tf-idf: boolean frequency

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

(1) **tf** is the boolean frequency

(a) $tf(x,P) = \begin{cases} 1, & \text{if } x \ in \ P \\ 0, & \text{otherwise} \end{cases}$

- if x in P, very simple, where P is the program

(2) **idf** is couched in the overall program

(a) $idf(x,\Pi) = log \frac{|\Pi|}{|\{P \in \ \Pi : x \in P\}|}$

- program corpus Π
- n-gram x
- measures important of x *in* Π

(3) **tf-idf** is then $tf(x,P) * idf(x,\Pi)$

(a) if x is in P then tf-idf of x is the same as its idf value

traditional tf-idf

Sentence 1 : The car is driven on the road.

Sentence 2: The truck is driven on the highway.

In this example, each sentence is a separate document.

We will now calculate the TF-IDF for the above two documents, which represent our corpus.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Semantic Similarity

3) tf-idf

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

(1) **tf** is the boolean frequency

(a) $tf(x, P) = \begin{cases} 1, & \text{if } x \ in \ P \\ 0, & \text{otherwise} \end{cases}$

- if x in P, very simple, where P is the program

(2) **idf** is couched in the overall program

(a) $idf(x, \Pi) = log \frac{|\Pi|}{|\{P \in \Pi : x \in P\}|}$

- program corpus Π
- n-gram x
- measures important of x *in* Π

(3) **tf-idf** is then $tf(x, P) * idf(x, \Pi)$

(a) if x is in P then tf-idf of x is the same as its idf value

```
   function inArray(a, val) {
1    begin;
2    i = 0;
3    $0 = a.length;
4    $1 = i < $0;
5    while ($1) {
6      $2 = a[i];
7      $3 = $2 === val;
8      if ($3) {
9        return true;
       }
10     i = i + 1;
11     $4 = a.length;
12     $1 = i < $4;
     }
13   return false;
14   end;
   }
```

2-gram *tf-idf*	3-gram *tf-idf*
0.000	0.000
1.017	1.017
0.969	0.969
2.238	2.876
1.641	2.368
3.035	3.590
4.767	6.704
4.560	5.296
1.699	5.911
1.934	2.232
2.024	2.312
1.564	3.846
1.857	1.857

Semantic Similarity

3) tf-idf

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
function inArray(a, val) {
  var i;
  for (i = 0; i < a.length; i++) {
    if (a[i] === val) {
      return true;
    }
  }
  return false;
}
```

essential component

```
   function inArray(a, val) {
1    begin;                          0.000   0.000
2    i = 0;                          1.017   1.017
3    $0 = a.length;                  0.969   0.969
4    $1 = i < $0;                    2.238   2.876
5    while ($1) {                    1.641   2.368
6      $2 = a[i];                    3.035   3.590
7      $3 = $2 === val;              4.767   6.704
8      if ($3) {                     4.560   5.296
9        return true;                1.699   5.911
       }
10     i = i + 1;                    1.934   2.232
11     $4 = a.length;                2.024   2.312
12     $1 = i < $4;                  1.564   3.846
     }
13   return false;                   1.857   1.857
14   end;
   }
```

2-gram *tf-idf* | 3-gram *tf-idf*

Semantic Similarity

3) tf-idf

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

```
function inArray(a, val) {
  var i;
  for (i = 0; i < a.length; i++) {
    if (a[i] === val) {
      return true;
    }
  }
  return false;
}
```

not just equality
but if equality break

		2-gram *tf-idf*	3-gram *tf-idf*
	`function inArray(a, val) {`		
1	`  begin;`	0.000	0.000
2	`  i = 0;`	1.017	1.017
3	`  $0 = a.length;`	0.969	0.969
4	`  $1 = i < $0;`	2.238	2.876
5	`  while ($1) {`	1.641	2.368
6	`    $2 = a[i];`	3.035	3.590
7	`    $3 = $2 === val;`	4.767	6.704
8	`    if ($3) {`	4.560	5.296
9	`      return true;`	1.699	5.911
	`    }`		
10	`    i = i + 1;`	1.934	2.232
11	`    $4 = a.length;`	2.024	2.312
12	`    $1 = i < $4;`	1.564	3.846
	`  }`		
13	`  return false;`	1.857	1.857
14	`  end;`		
	`}`		

Semantic Similarity

3) tf-idf: efficiency

Using Web Corpus Statistics for Program Analysis

Chun-Hung Hsiao Michael Cafarella Satish Narayanasamy
University of Michigan
{chhsiao,michjc,nsatish}@umich.edu

- take each statement in canonical form
- find the statement's PDG
- run `NGramBFS` with a depth limit of n-1 to find corresponding nodes

Algorithm 1 The n-gram extraction algorithm.

function EXTRACTNGRAMS(n, P)
  $P' \leftarrow \chi(P)$ ▷ Canonical form of P
  $G \leftarrow \text{PDG}(P')$ ▷ Program dependence graph of P'
  $\Gamma \leftarrow \emptyset$ ▷ The set of all n-grams in P
  for $p \in P'$ **do**
    $\Gamma \leftarrow \Gamma \cup \{\text{NGRAMBFS}(G, p, n)\}$
  return Γ

function NGRAMBFS(G, v, n)
  $V \leftarrow \{v\}$ ▷ The set of vertices with distance $\leq n-1$
  $E \leftarrow \emptyset$ ▷ The set of edges with distance $\leq n-1$
  $d[v] \leftarrow 0$
  $Q \leftarrow \emptyset$
  ENQUEUE(Q, v)
  while $Q \neq \emptyset$ **do** ▷ Breadth-first search with depth $\leq n-1$
    $v \leftarrow$ DEQUEUE(Q)
    for $(v, u) \in G$ **do**
      $E \leftarrow E \cup \{(v, u)\}$
      if $u \notin V$ **then**
        $V \leftarrow V \cup \{u\}$
        $d[u] \leftarrow d[u] + 1$
        if $d[u] < n-1$ **then**
          ENQUEUE(Q, u)
  return (V, E)

Experiments

canonical form

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies ; 2017 (1):79–99

back to the main paper

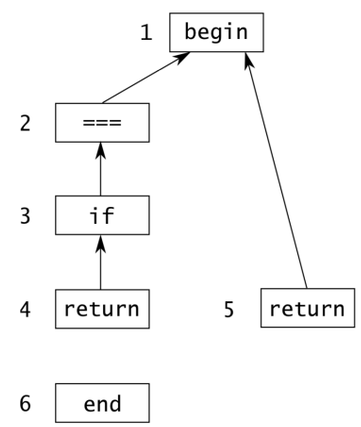
Muhammad Ikram*, Hassan Jameel Asghar, Mohamed Ali Kaafar, Anirban Mahanti, and Balachandar Krishnamurthy

Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-class Learning

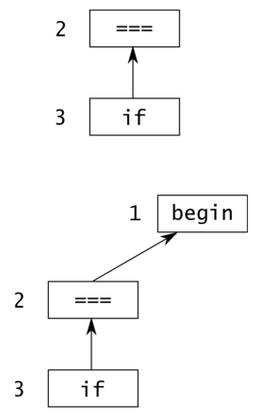
JavaScript Program → Canonical Form → PDGs n-grams

```
function equalTest(a, b){
  if(a == b){
    return true;}
  return false;}
```

```
function equalTest(a, b){
1:    begin;
2:    $0 = a === b;
3:    if($0){
4:      return true;}
5:    return false;
6:    end;}
```



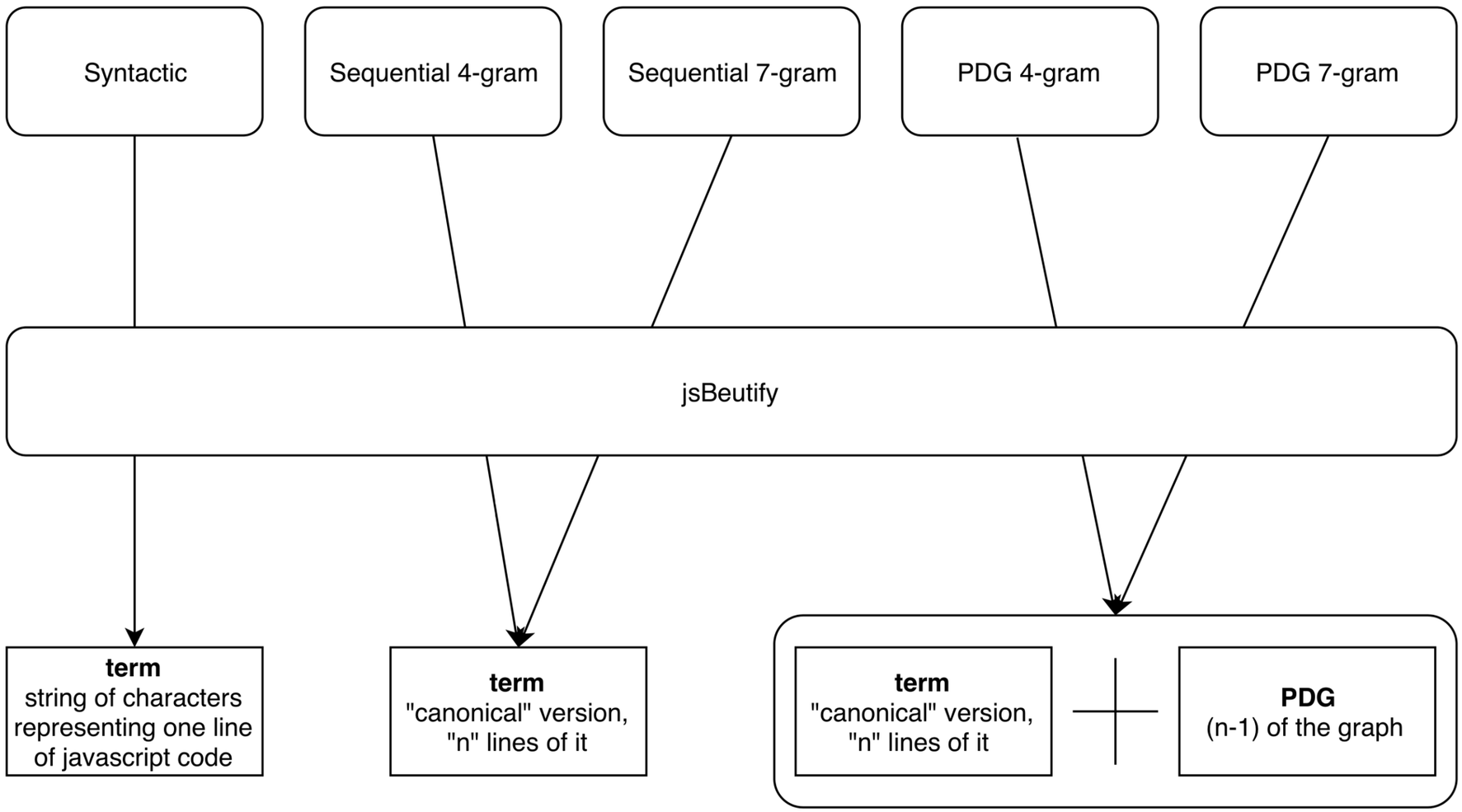
(a) 1-gram



(b) 2-gram

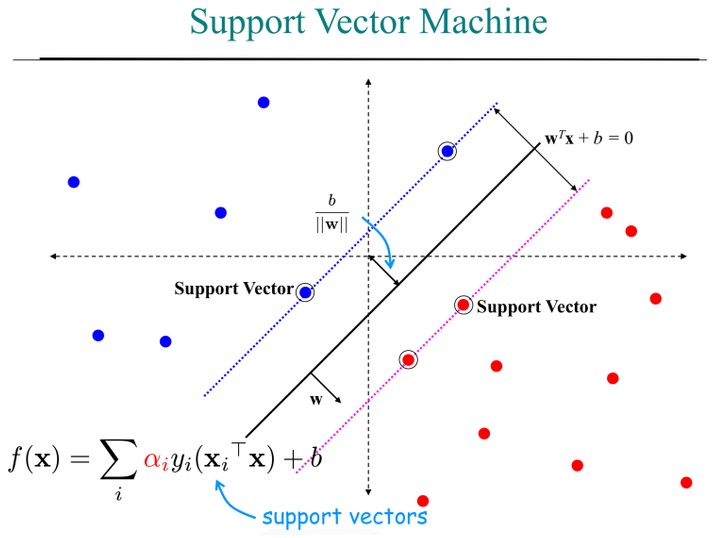
Experiments

semantic similarity: likeness between programs



Experiments

features

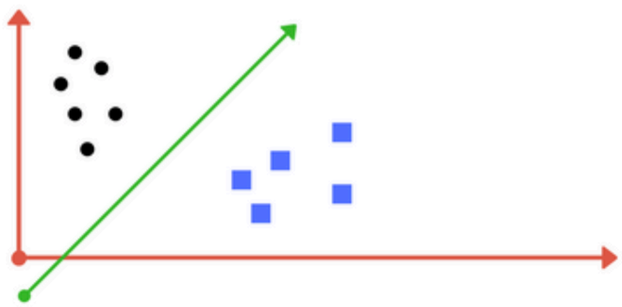


- **Support Vector Machine (SSVM)**
  - traditional, two-class
  - used as a baseline
- **one-class SVM (OCSVM)**
  - maps the feature vectors to a higher dimensional space through an appropriate kernel (radial basis function), then finds hyperplane whose margin from the origin is maximized
  - similar to two class, but without negative examples
- **positive and unlabeled learning (PU)**
  - (1) identify a set of reliable negative documents from unlabeled set; (2) build a set of classifiers by iteratively applying a classification algorithm and then selecting a good classifier from the set
  - “These two steps together can be seen as an iterative method of increasing the number of unlabeled examples that are classified as negative while maintaining the positive examples correctly classified.” Building Text Classifiers Using Positive and Unlabeled Examples

Experiments

SVM background

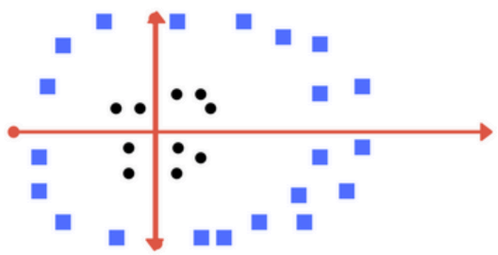
- **Support Vector Machine (SSVM)**
  - with two dimensional data, you simply find a demarcating line (like a perceptron)



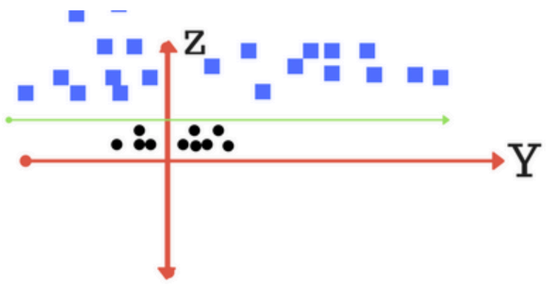
Experiments

SVM background

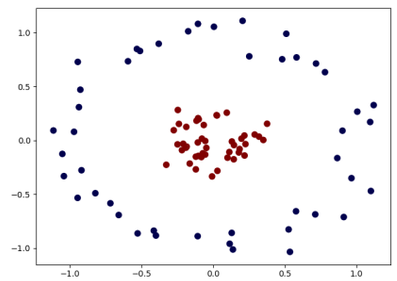
- **Support Vector Machine (SSVM)**
  - but with multi-dimensional data, you need to apply a kernel to transform the data and find a hyperplane

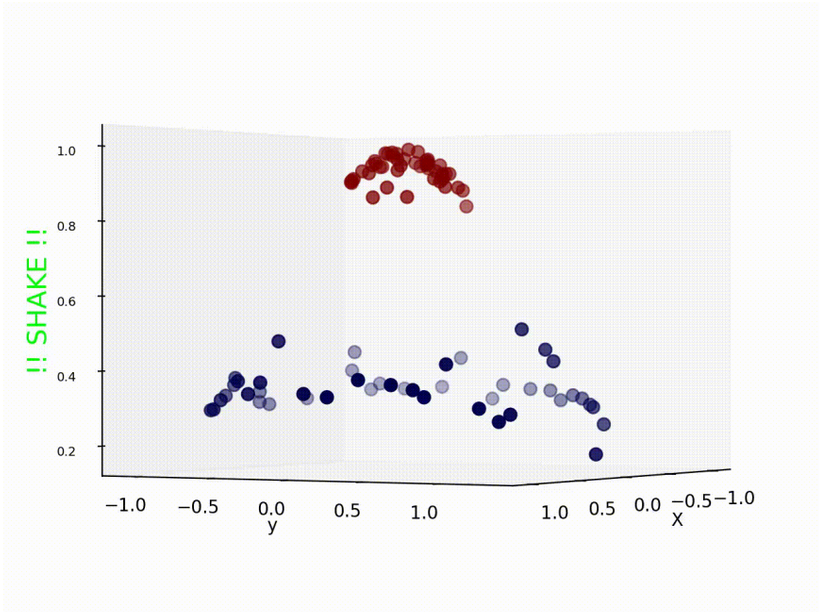


Can you draw a separating line in this plane?



plot of zy axis. A separation can be made here.

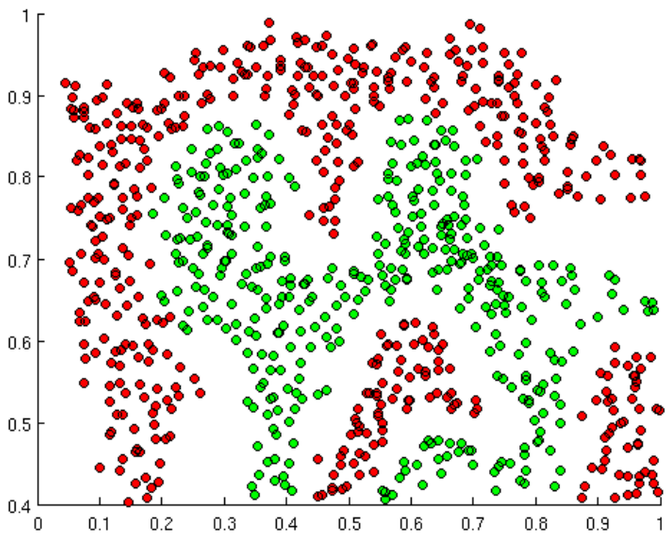




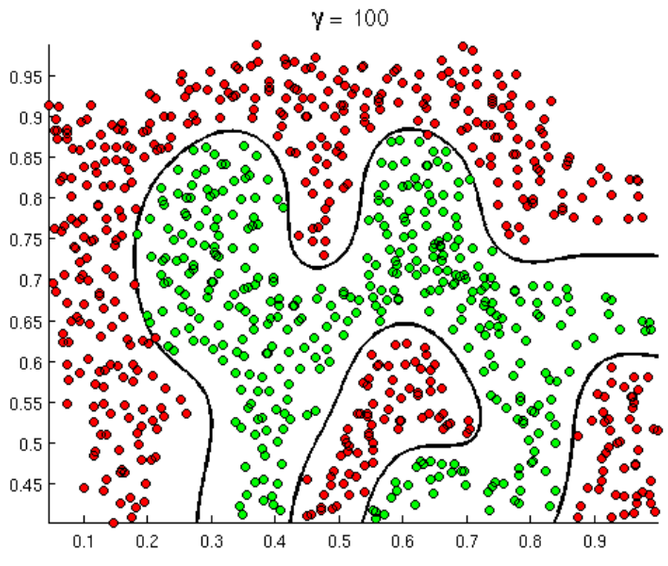
Experiments

SVM background

- **Support Vector Machine (SSVM)**
  - and to find the hyperplane, you use a kernel—here, the radial basis function
    - like RBF NNs—you can think of the RBG as a transformer that to generates new features by measuring the distance between all other dots to a specific dot/dots—centers (link *for source*))
- OVERALL: used to handle non-linearly separable data—which is then transformed into linearly separable data (i.e., the “kernel trick”) on a higher dimension



$$K\left(x^{(i)}, x^{(j)}\right) = \phi(x^{(i)})^T \phi(x^{(j)})$$
$$= \exp\left(-\gamma \left\| x^{(i)} - x^{(j)} \right\|^2\right), \qquad \gamma > 0$$



Evaluation

classification

Feature Model	Classifier	Tracking		Functional	
		Blocked	Allowed	Blocked	Allowed
Syntactic	SSVM	0.93	0.07	0.01	0.99
	OCSVM	0.88	0.12	0.02	0.98
	PU	0.86	0.14	0.02	0.98
PDG 4-gram	SSVM	0.96	0.04	0.03	0.97
	OCSVM	0.95	0.05	0.03	0.97
	PU	0.93	0.07	0.04	0.96
Sequential 4-gram	SSVM	0.98	0.02	0.01	0.99
	OCSVM	0.98	0.02	0.02	0.98
	PU	0.96	0.04	0.03	0.97
PDG 7-gram	SSVM	0.99	0.01	0.01	0.99
	OCSVM	0.99	0.01	0.01	0.99
	PU	0.98	0.02	0.02	0.98
Sequential 7-gram	SSVM	0.99	0.01	0.01	0.99
	OCSVM	0.99	0.01	0.01	0.99
	PU	0.98	0.02	0.02	0.98

Table 5. Performance of the classifiers against the labelled dataset of tracking and functional JavaScript programs. true positives and negatives, false positives and negatives.

- similar performance across all models
- notable: these results come from the manually labeled set

Evaluation

adblockers on manually labeled set

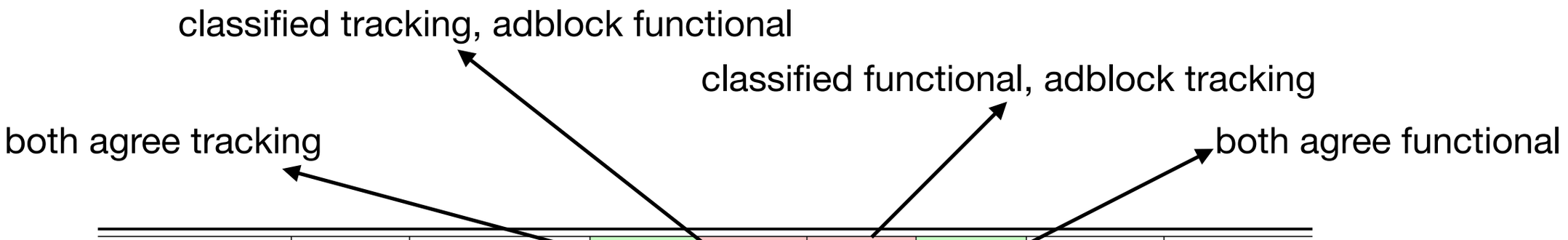
PP-Tool	Tracking		Functional	
	Blocked	Allowed	Blocked	Allowed
NoScript	0.78	0.22	0.21	0.79
Ghostery	0.65	0.35	0.08	0.92
Adblock Plus	0.44	0.56	0.06	0.94
Disconnect	0.40	0.60	0.06	0.94
Privacy Badger	0.37	0.63	0.06	0.94

Table 4. Comparison of the output of PP-Tools against our labelled set of tracking and functional JavaScript programs. ■ true positives and negatives, ■ false positives and negatives.

- more blocking == more false positives
- NoScript blocks the most (both tracking and functional)
- Disconnect, adblock Plus, and Privacy Bader allowed the most

Evaluation

classifiers and adblockers in the wild (4084 domains)



Feature Model	Classifier	PP-Tool	$T_c \cap T_p$	$T_c \cap F_p$	$F_c \cap T_p$	$F_c \cap F_p$	Agreement	Disagreement
Syntactic	OCSVM	NoScript	0.56	0.10	0.29	0.05	0.61	0.39
		Ghostery	0.54	0.13	0.27	0.06	0.60	0.40
		Adblock Plus	0.47	0.20	0.25	0.09	0.56	0.44
		Privacy Badger	0.23	0.44	0.11	0.22	0.45	0.55
		Disconnect	0.17	0.50	0.08	0.25	0.42	0.58
Sequential 7-gram	OCSVM	NoScript	0.71	0.06	0.14	0.09	0.80	0.20
		Ghostery	0.67	0.10	0.15	0.08	0.75	0.25
		Adblock Plus	0.62	0.15	0.11	0.13	0.75	0.25
		Privacy Badger	0.27	0.5	0.07	0.16	0.43	0.57
		Disconnect	0.19	0.58	0.06	0.17	0.36	0.64
Syntactic	PU	NoScript	0.50	0.07	0.36	0.07	0.57	0.43
		Ghostery	0.47	0.10	0.35	0.08	0.55	0.45
		Adblock Plus	0.43	0.14	0.30	0.13	0.56	0.44
		Privacy Badger	0.18	0.38	0.15	0.28	0.46	0.54
		Disconnect	0.13	0.44	0.12	0.31	0.44	0.56
Sequential 7-gram	PU	NoScript	0.70	0.05	0.16	0.09	0.79	0.21
		Ghostery	0.65	0.10	0.16	0.09	0.74	0.26
		Adblock Plus	0.61	0.14	0.12	0.13	0.74	0.26
		Privacy Badger	0.18	0.57	0.07	0.18	0.36	0.64
		Disconnect	0.26	0.49	0.07	0.18	0.44	0.56

agreement on tracking = $T_c \cap T_p / J$

agreement on functional = $F_c \cap F_p / J$

Table 7. Agreement and disagreement in classification of tracking and functional JavaScript programs between our classifiers and PP-Tools on the wild dataset; agreement, disagreement; T_p and F_p represent JavaScript programs classified as tracking and functional, respectively, by the PP-Tool p, and T_c and F_c represent JavaScript programs classified as tracking and functional, respectively, by the classifier c.

Evaluation

disagreements

- **disagreements**
  - classified as tracking or functional in disagreement with adblockers
  - 4610 programs in total
    - 100 programs were randomly picked for manual inspection
- those 100 programs were labeled

Disagreement	Total	Sample	Manual Labelling	
			Tracking	Functional
$T_c \cap_p F_p$	4,610	100	75	25
$F_c \cap_p T_p$	4,461	100	19	81

Evaluation

disagreements

- **classified as tracking, manually classified as functional** *(all adblockers thought functional)*
  - 75/100 of the tracking-classified programs were “correct” according to human review
    - adblockers do not run regex-styled matching on the body of programs, only domains
    - you can’t block what you don’t know about
- **classified as functional, manually classified as tracking** *(all adblockers thought tracking)*
  - 81/100 of the functional-classified programs were “correct” according to human review

Disagreement	Total	Sample	Manual Labelling	
			Tracking	Functional
$T_c \cap_p F_p$	4,610	100	75	25
$F_c \cap_p T_p$	4,461	100	19	81

Limitations

- **the scrape**
    - 180 seconds? more time = more programs?
- **manually labeled dataset**
    - difficult to replicate
        - response: crowd-sourced training using “tech savvy workers”
- **“tracking” versus “functional” == dueling definitions**
    - what does functionality really mean?
    - using rules (human generated) to block likeness (machine generated)—but how good are those rules in the first place

Limitations

- **obfuscation**
  - adblockers susceptible to new domains (this approach is not)
    - this approach susceptible to features not found in dataset
      - response: retraining—but time consuming
- **features**
  - jsBeautify could be stripping important detail

Key Takeaways

- tracking scripts are similar to each other
- similar enough to train an ML model with only a small set of labeled programs
- detecting similarity is most accurate when PDG n-grams are used
  - canonical form —> PDG —> n-grams

References

- Towards Seamless Tracking-Free Web: Improved Detection of Trackers via One-class Learning, Muhammad Ikram et al., **PoPETS** 2017

- Using Web Corpus Statistics for Program Analysis, Chun-Hung Hsiao, Michael Cafarella, Satish Narayanasamy, **OOPSLA** 2014

- Pixel Perfect: Fingerprinting Canvas in HTML5, Keaton Mowery, Hovav Shacham, **W2SP** 2012

- Online Tracking: A 1-million-site Measurement and Analysis, Steven Englehardt, Arvind Narayanan, ACM **CCS**, 2016