Why would robots read your code?

Alexey Smirnov & Konstantin Tyapochkin

PiterPy, 2019


CODE SCORING
IT Risk Management
When AI is on a guard of solutions Quality
www.codescoring.com
Made with ❤️ by Profiscope Team

CodeMining, что ты такое?

Вы не поверите!

- Все разработчики пишут код!

Вы не поверите!

- Все разработчики пишут код!
- Но не многие его читают ¯_(ツ)_/¯

Анализ исходных кодов. Обычное дело (уже и теперь)

1. **Автокомплит**. В вашем любимом IDE он 99% есть
2. **Автокорректор** (linter). Для тех, кому нужен порядок
3. **Рефакторинг**. Есть вероятность, что вы пользуетесь
4. **Код ревью**. На что «никогда не хватает времени»
5. **Статический и динамический анализ** кода - для сильных мира сего

Анализ кода и артефактов — что (будет) популярно

1. Суммаризация и генерация исходных кодов и доков
2. Анализ похожести (clone detection) и заимствований
3. Оценка качества исходных кодов

   Безопасность, потенциальные ошибки, читаемость
4. Оценка совместимости разработчиков
5. Оценка мнений разработчиков: о коде и о коллегах
6. Классификация, структуризация исходников
7. Анализ код-артефактов

   Полнота документации, совместимость лицензий

Анализ кода и артефактов — рассмотрим сегодня

1. **Суммаризация и генерация исходных кодов и доков**
2. Анализ похожести (clone detection) и заимствований
3. Оценка качества исходных кодов

   Безопасность, потенциальные ошибки, читаемость
4. Оценка совместимости разработчиков
5. **Оценка мнений разработчиков: о коде и о коллегах**
6. Классификация, структуризация исходников
7. **Анализ код-артефактов**

   Полнота документации, совместимость лицензий

Анализ артефактов разработки

Анализ артефактов разработки

Виды код-артефактов:

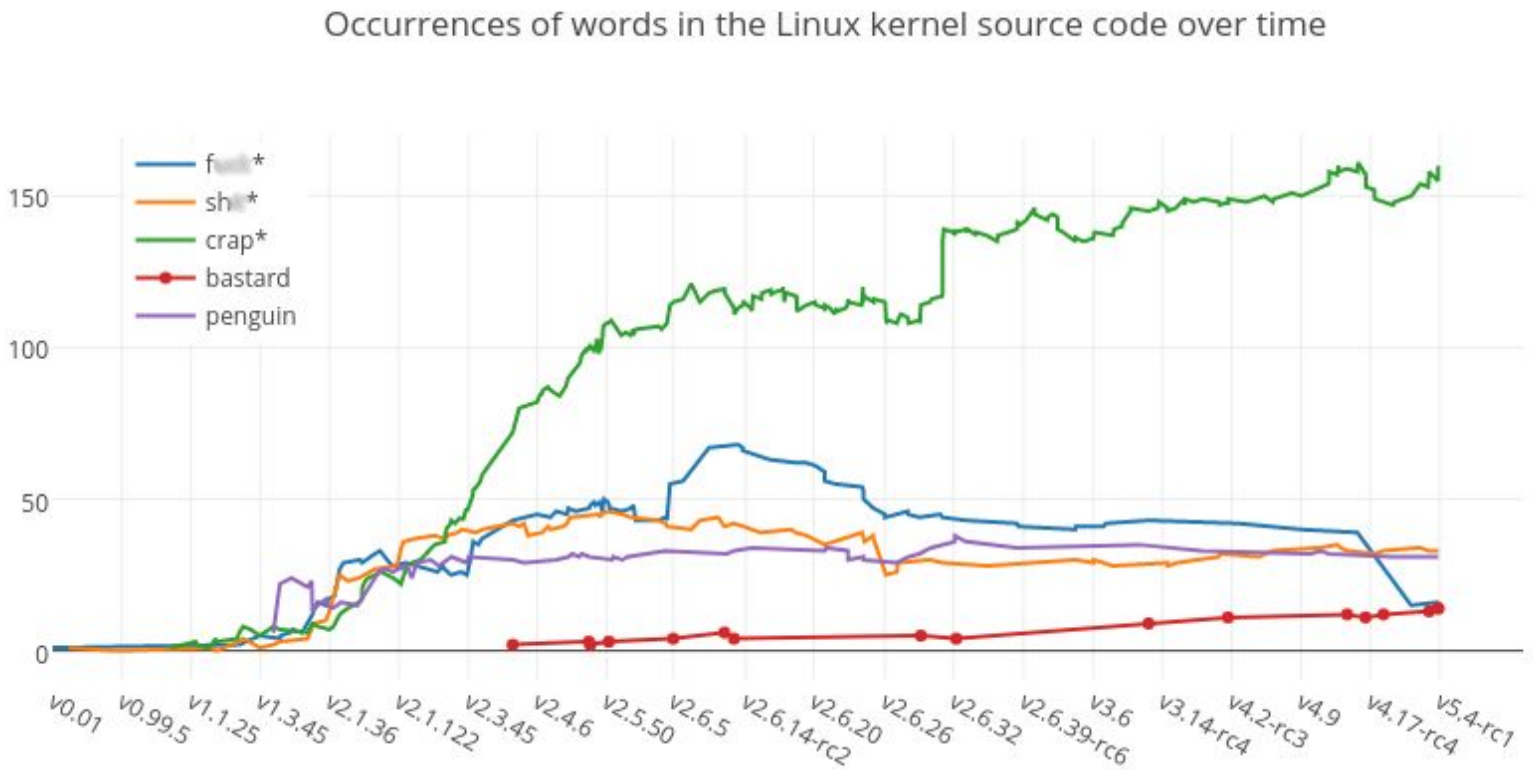
- Комментарии в коде
- Коммит-сообщения
- Код-ревью
- Тикет / его мета-данные
- Текст лицензии
- Листы рассылки
- Код в статьях (paperswithcode.com)
- Код в обучающих видео / слайдах
- QA (StackOverflow) и smell code (govnokod.ru).

Анализ артефактов разработки

Виды код-артефактов:

- **Комментарии в коде**
- **Коммит-сообщения**
- **Код-ревью**
- Тикет / его мета-данные
- Текст лицензии
- Листы рассылки
- Код в статьях (paperswithcode.com)
- Код в обучающих видео / слайдах
- QA (StackOverflow) и smell code (govnokod.ru).

Анализ встречаемости слов

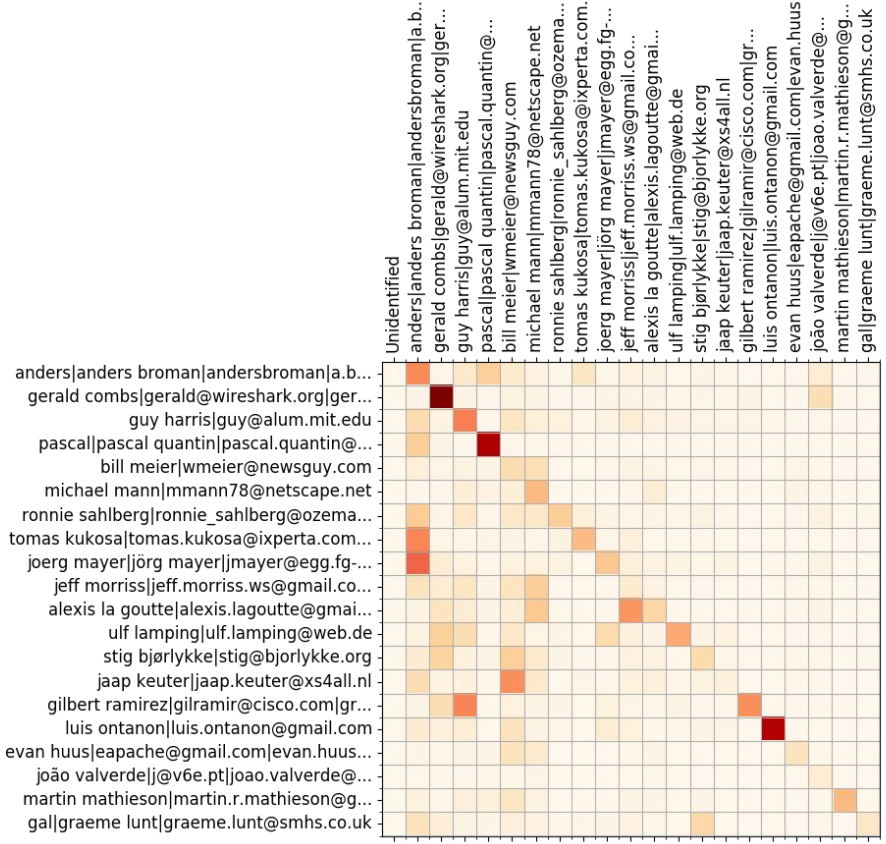


Поиграться можно тут: vildarholen.net/contents/wordcount

В попытках что-нибудь разглядеть

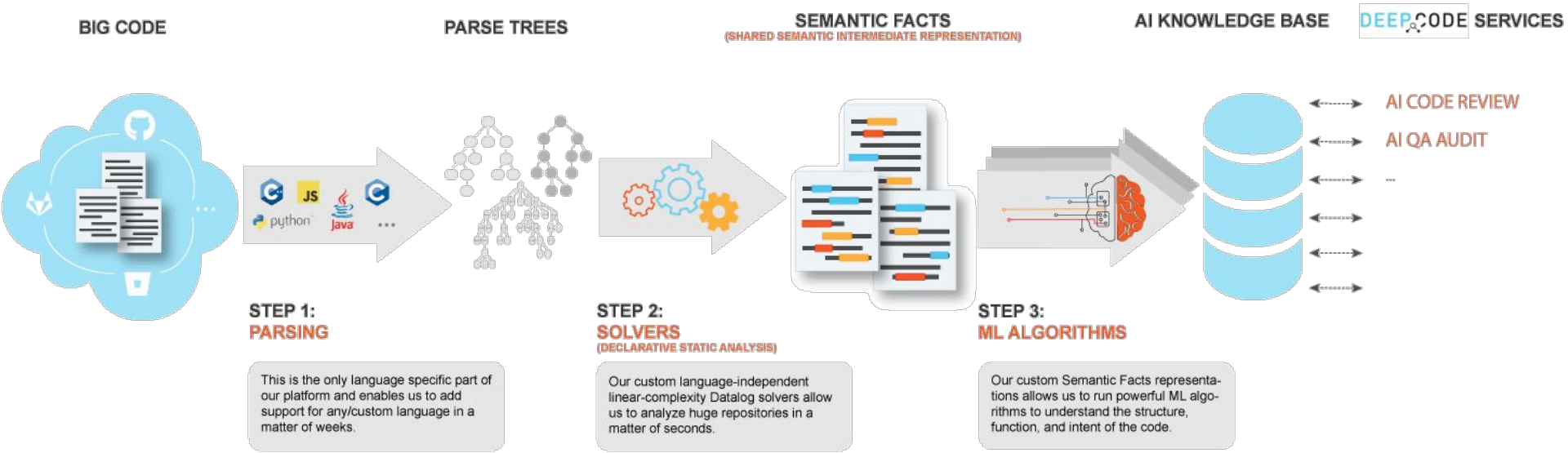
```
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 290         })
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 291
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 292         if err != nil {
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 293                 return err
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 294         }
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 295
501a9727 Máximo Cuadros   2016-08-19 17:42:13 +0200 296         return r.createReferences(head)
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 297 }
86a411c1 Máximo Cuadros   2016-08-15 21:43:33 +0200 298
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 299 // Commit return the commit with the given hash
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 300 func (r *Repository) Commit(h plumbing.Hash) (*Commit, error) {
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 301         commit, err := r.Object(plumbing.CommitObject, h)
c4a05c0a Joshua Sjoding   2016-02-16 03:28:21 -0800 302         if err != nil {
c4a05c0a Joshua Sjoding   2016-02-16 03:28:21 -0800 303                 return nil, err
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 304         }
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 305
0d670160 Máximo Cuadros   2016-09-12 02:22:08 +0200 306         return commit.(*Commit), nil
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 307 }
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 308
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 309 // Commits decode the objects into commits
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 310 func (r *Repository) Commits() (*CommitIter, error) {
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 311         iter, err := r.s.IterObjects(plumbing.CommitObject)
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 312         if err != nil {
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 313                 return nil, err
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 314         }
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 315
635c77e0 Alberto Cortés   2016-07-04 17:09:22 +0200 316         return NewCommitIter(r, iter), nil
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 317 }
2d94e011 Máximo Cuadros   2015-10-27 01:49:58 +0100 318
9ec9b0db Joshua Sjoding   2016-02-19 15:42:00 -0800 319 // Tree return the tree with the given hash
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 320 func (r *Repository) Tree(h plumbing.Hash) (*Tree, error) {
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 321         tree, err := r.Object(plumbing.TreeObject, h)
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 322         if err != nil {
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 323                 return nil, err
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 324         }
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 325
0d670160 Máximo Cuadros   2016-09-12 02:22:08 +0200 326         return tree.(*Tree), nil
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 327 }
aa78803b Joshua Sjoding   2016-02-18 23:28:06 -0800 328
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 329 // Trees decodes the objects into trees
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 330 func (r *Repository) Trees() (*TreeIter, error) {
f49096ea Máximo Cuadros   2016-11-08 23:46:38 +0100 331         iter, err := r.s.IterObjects(plumbing.TreeObject)
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 332         if err != nil {
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 333                 return nil, err
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 334         }
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 335
d025b0e1 Santiago M. Mola 2016-11-04 16:12:01 +0100 336         return NewTreeIter(r, iter), nil
```

Code Wars


Unidentified
anders|anders broman|andersbroman|a.b...
gerald combs|gerald@wireshark.org|ger...
guy harris|guy@alum.mit.edu
pascal|pascal quantin|pascal.quantin@...
bill meier|wmeier@newsguy.com
michael mann|mmann78@netscape.net
ronnie sahlberg|ronnie_sahlberg@ozema...
tomas kukosa|tomas.kukosa@ixperta.com...
joerg mayer|jörg mayer|jmayer@egg.fg-...
jeff morriss|jeff.morriss.ws@gmail.co...
alexis la goutte|alexis.lagoutte@gmai...
ulf lamping|ulf.lamping@web.de
stig bjørlykke|stig@bjorlykke.org
jaap keuter|jaap.keuter@xs4all.nl
gilbert ramirez|gilramir@cisco.com|gr...
luis ontanon|luis.ontanon@gmail.com
evan huus|eapache@gmail.com|evan.huus...
joão valverde|j@v6e.pt|joao.valverde@...
martin mathieson|martin.r.mathieson@g...
gal|graeme lunt|graeme.lunt@smhs.co.uk

Code Wars / Решение проблемы

DeepCode.ai — автоматизация код-ревью, которая частично снимает проблемы стоимости процесса и адекватности/коллизий участников ;).



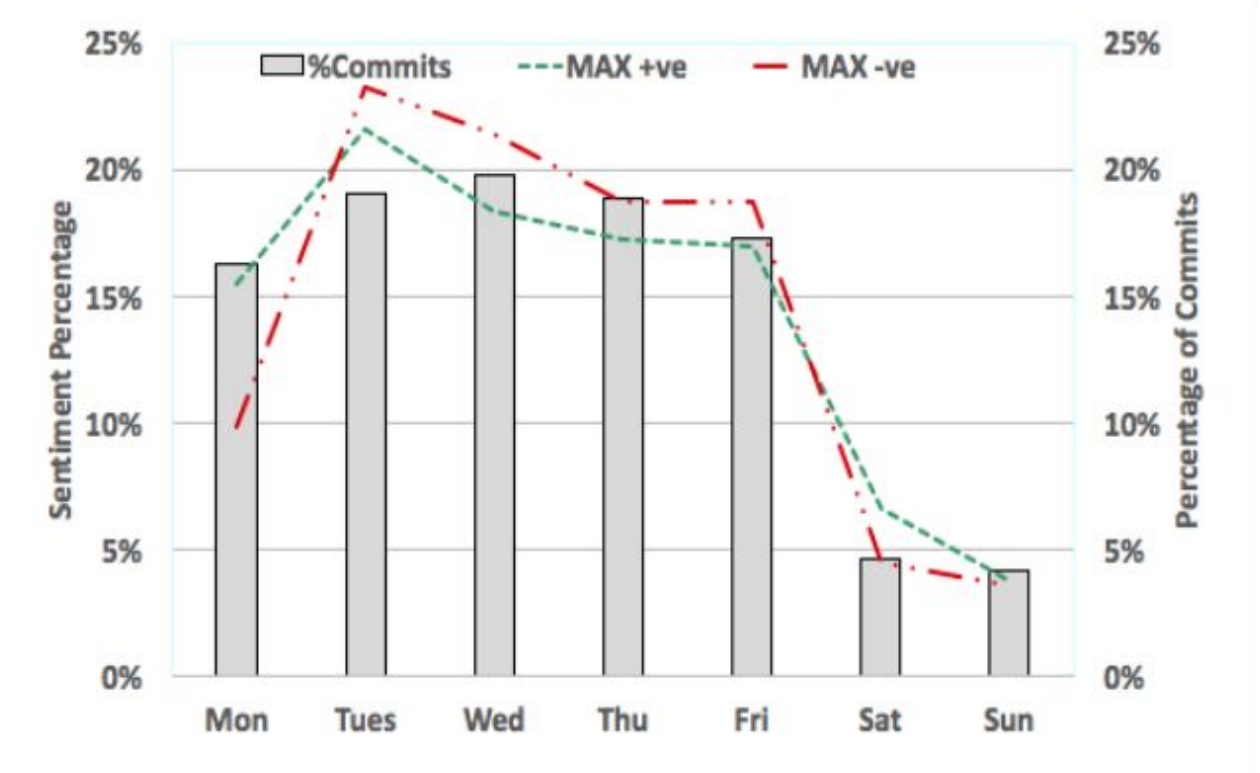
Сентименты в код-артефактах / Примеры

Sentiment	Commit Message	Final Score
Positive	We're not totally terrible.	4
	Build success !!!	3
	pretty pretty code	3
	Added parallelism and seems it works fine :)	3
	A few finishing touches that Anna liked :)	3
	Small tweaks on top of Daniel's excellent refactoring git-svn-id	3
Negative	Terrible, terrible mock folder guid retrieval.	-4
	Trying to complete the qualifier 3. Grounds for suicide :(	-4
	Fix heinous TMemoryBuffer bug and warning in FileTransport Review	-4
	Attempted to fix map camera failed horribly	-4
	ENH: very painfully merge: svn merge --accept	-4
Neutral	initial commit Committer: Jeremy Truelove jtruelove@gmail.com	0

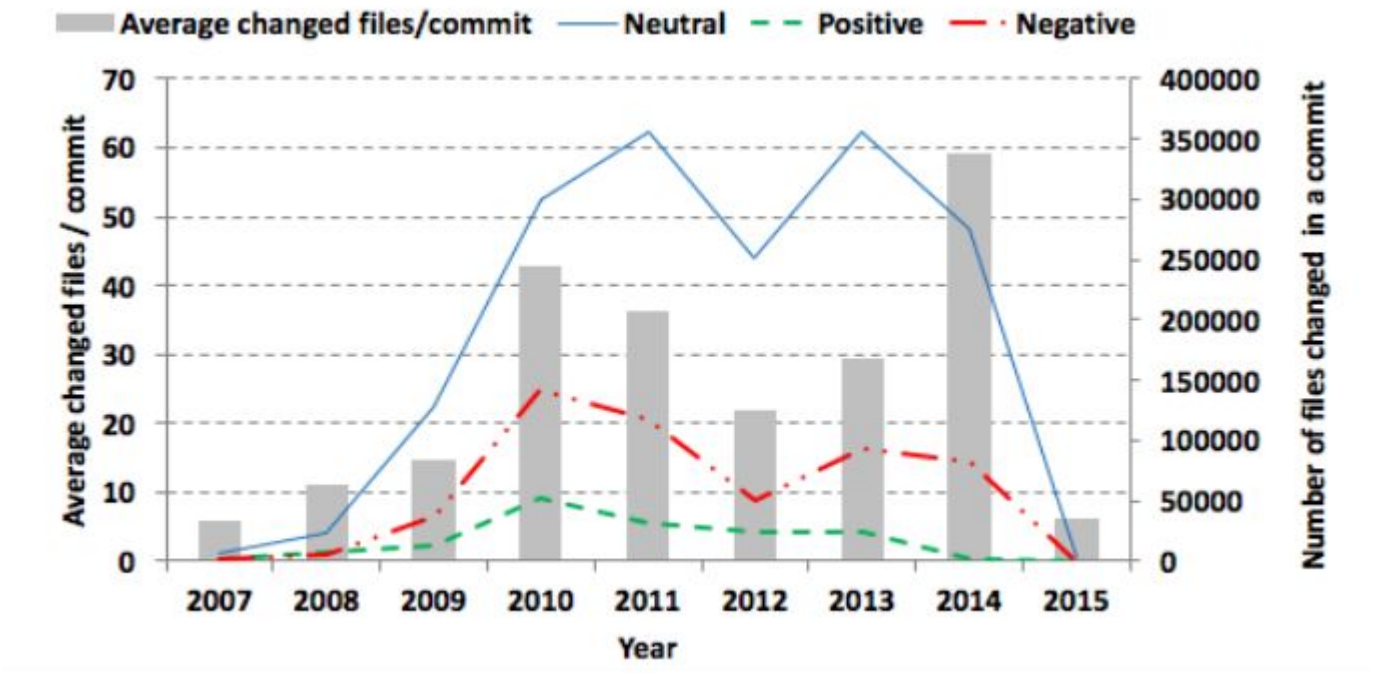
Сентименты в код-артефактах / Расчеты

Sentiment	Final Sentiment Score	Number of Commits	Sentiment Percentage
Negative	-4	66	18.05%
	-3	2793	
	-2	39770	
	-1	363853	
Neutral	0	1683009	74.75%
Positive	1	149931	7.20%
	2	11782	
	3	371	
	4	10	
	Total	**2251585**	

Сентименты в код-артефактах / Дни недели



Сентименты в код-артефактах / Сквозь года



Сентименты в код-артефактах / Что пощупать?

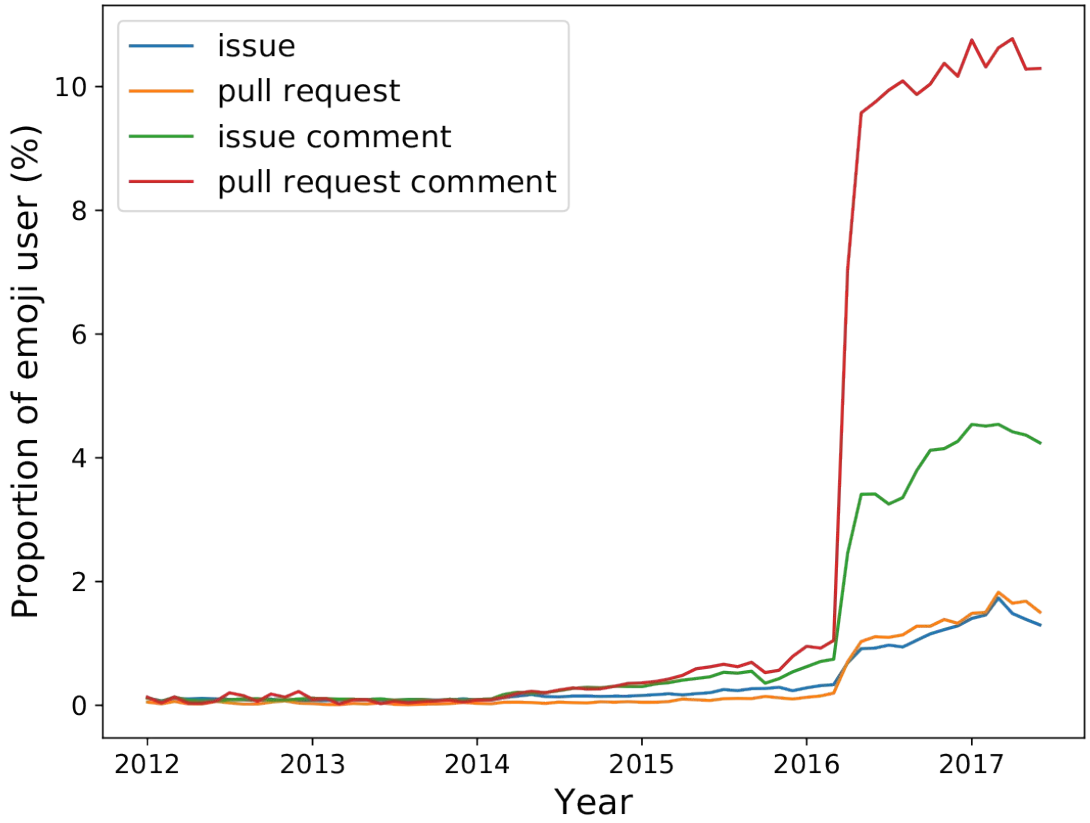
- SentiCR (https://github.com/senticr/SentiCR)
- SentiStrengthSE

  (demo: https://laser.cs.uno.edu/Projects/Projects.html)
- Senti4SD (https://github.com/collab-uniba/Senti4SD)
- EMTk - Emotion Mining Toolkit (https://github.com/collab-uniba/EMTk)

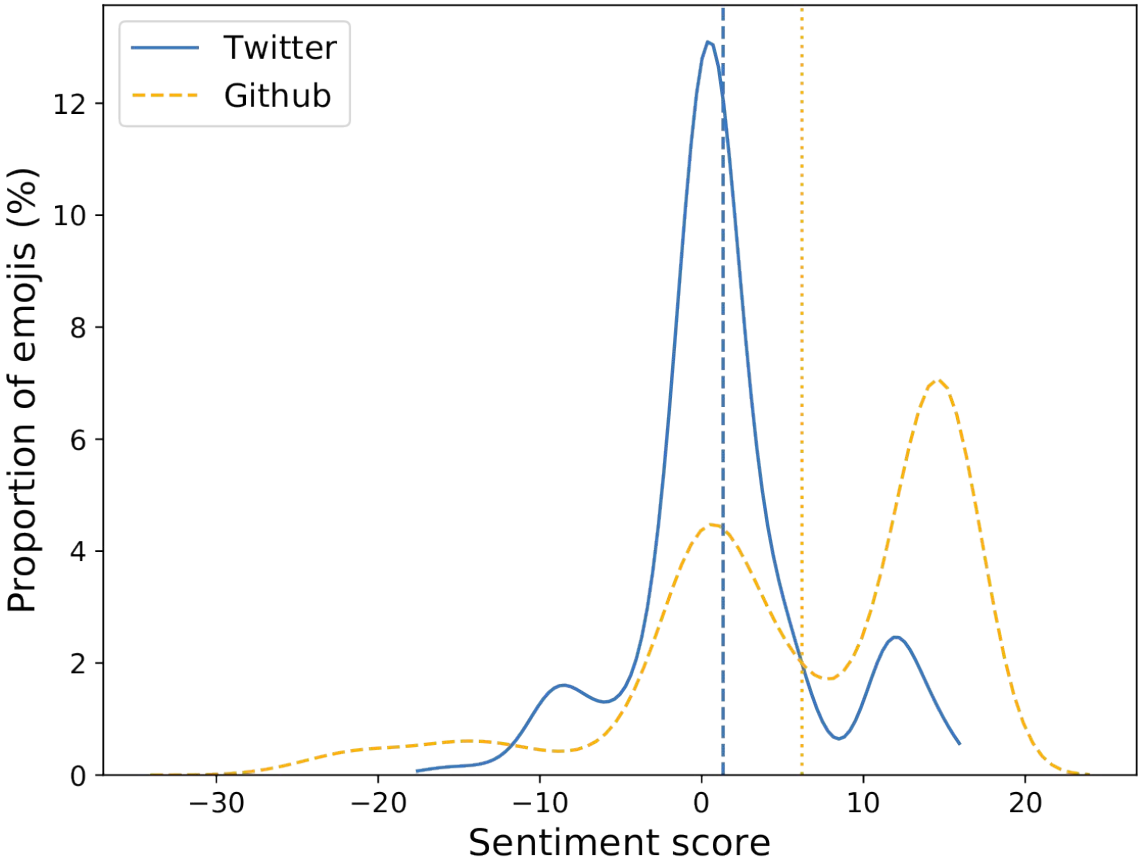
Сентименты в код-артефактах / Emoji - Gitmoji

- 1997 год, появление
- 2010 год, становление в WWW
- 2016 год, становление на Github
  (на Github — 1271 уникальных emoji)
- 2016 год, проект Gitmoji и попытка
  сообщества привести к стандарту:
  https://github.com/carloscuesta/gitmoji
  (сейчас их 64)

Сентименты в код-артефактах / Emoji + Github



Сентименты в код-артефактах / Emoji + Sentiment score



Сентименты в код-артефактах / Применимость

- Понимание «тонуса» разработчика и команды
- Понимание мнения о коллегах и о продукте
- Косвенная оценка технического долга
- Оценка качества тикетов
- Оценка качества работы первой линии
- Оценка эффективности работы в таск-менеджере

Суммаризация и автогенерация документации

Суммаризация и автогенерация документации

Задачи суммаризации

- Генерация названий методов/переменных
- Генерация короткой документации
- Суммаризация в классическом смысле — генерация текста с описанием

Суммаризация и автогенерация документации

Применение

- Генерация документации к коду
- Генерация комментариев к коду
- Изменение названий методов и переменных для улучшения читаемости
- Выделение и подсветка самых важных частей кода
- Автоматический нейминг методов при создании API, SDK
- Каталогизация больших кодовых баз

Простой пример суммаризации

```
def add(a, b):
    return a + b
```

add two numbers
(реальное описание: sum two numbers)

Сложный пример суммаризации

```
def _has_git():
    try:
        subprocess.check_call(['git', '--version'],
            stdout=subprocess.DEVNULL,
            stderr=subprocess.DEVNULL)
    except (OSError,
            subprocess.CalledProcessError):
        return False
    else:
        return True
```

returns true if git is installed

(реальное описание: check if git is installed)

И еще пример

```
def tensor3(name=None, dtype=None):
    if (dtype is None):
        dtype = config.floatX
    type = CudaNdarrayType(dtype=dtype,
                           broadcastable=(False, False, False))
    return type(name)
```

return a symbolic graph

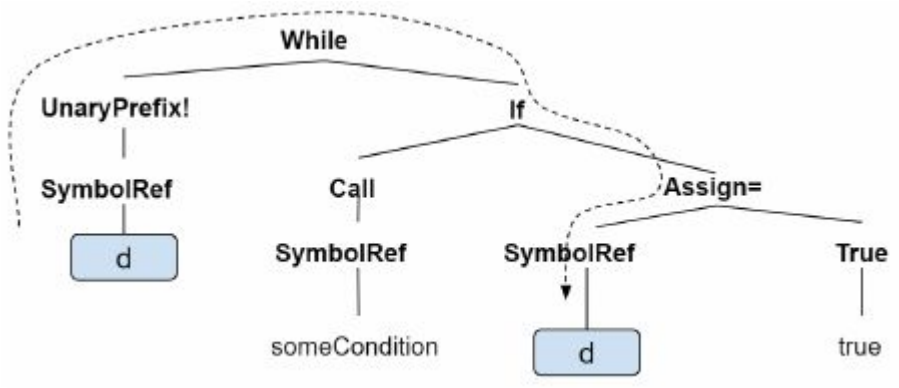
(реальное описание: return a symbolic 3-d variable)

Path-Based представления

Сниппет кода можно представить набором путей в AST дереве:

- Листы дерева - конкретные значения терминалов
- Начало и конец каждого пути - листы дерева
- Путь должен идти слева направо

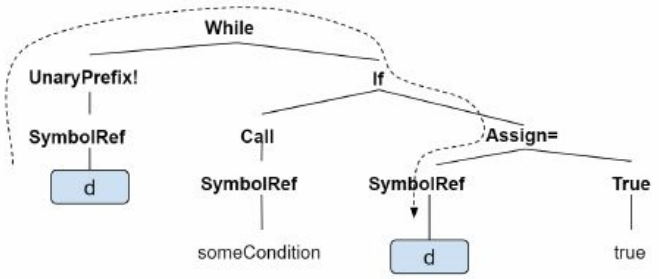
```
while not d:
    if someCondition():
        d = True
```



Path-Based представления

Сниппет кода можно представить набором путей в AST дереве:

- Листы дерева - конкретные значения терминалов
- Начало и конец каждого пути - листы дерева
- Путь должен идти слева направо

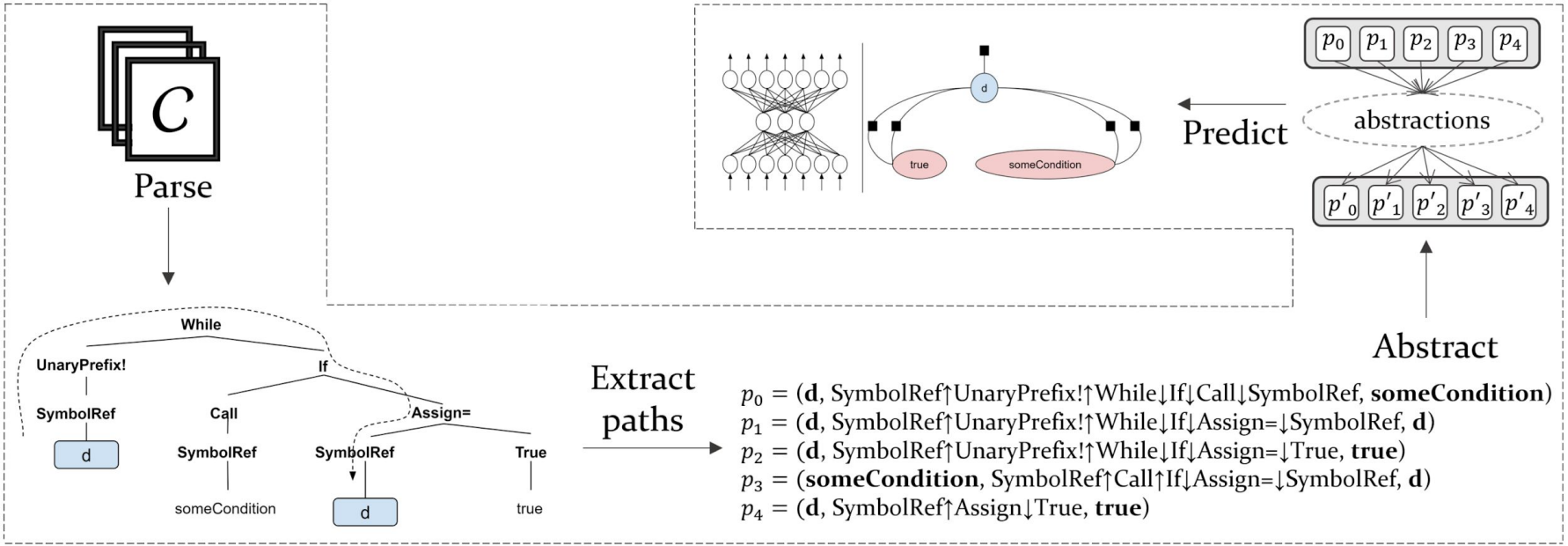


p_0 = (**d**, SymbolRef↑ UnaryPrefix!↑ While↓ If↓ Assign=↓ SymbolRef, **d**)

p_1 = (**d**, SymbolRef↑ Assign=↓ True, **true**)

...

Path-Based представления / Обучение



Path-Based представления / Первые результаты

Точность

- предсказание имен переменных:

  56% (Java, **Python**, C#) - 67%(JavaScript)

- предсказание названий методов:

  50% (JavaScript, Java, **Python**)

Предсказание имен переменных / Пример

Урезанные

```
def sh3(c):
    p = Popen(c, stdout=PIPE,
        stderr=PIPE, shell=True)
    o, e = p.communicate()
    r = p.returncode
    if r:
        raise CalledProcessError(r, c)
    else:
        return o.rstrip(), e.strip()
```

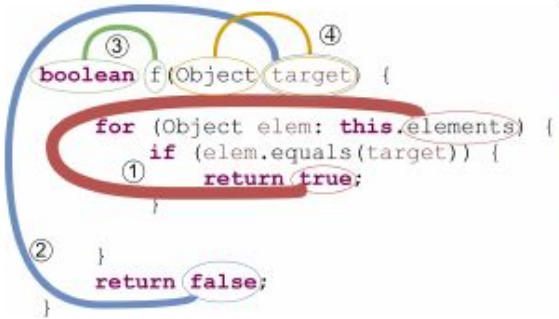
Восстановленные

```
def sh3(cmd):
    process = Popen(cmd, stdout=PIPE,
              stderr=PIPE, shell=True)
    out, err = process.communicate()
    retcode = process.returncode
    if retcode:
        raise CalledProcessError(retcode,
                                 cmd)
    else:
        return out.rstrip(), err.strip()
```

Дальнейшие разработки

- Path-Based и эмбеддинги:
  - Code2Vec (https://code2vec.org/)
  - Code2Seq (https://code2seq.org/)
  - + PathMiner от JB (https://github.com/vovak/astminer)
- Более умные модели “понимания” кода по тексту и AST деревьям
  - Разные рекуррентные нейросети
  - И даже обучение с подкреплением (https://github.com/wanyao1992/code_summarization_public)
- Динамические модели. Анализ трейса выполнения программы
- ...

Code2Vec / Пример предсказания названий методов



(a)

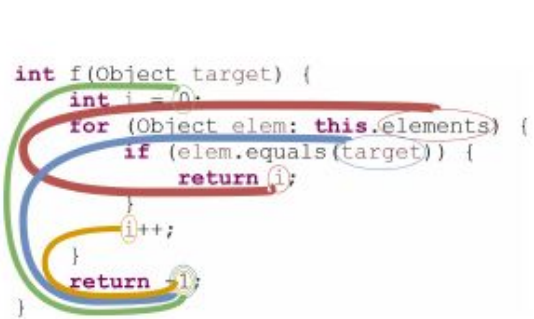
Predictions:

contains	90.93%
matches	3.54%
canHandle	1.15%
equals	0.87%
containsExact	0.77%

(b)

Predictions

get	31.09%
getProperty	20.25%
getValue	14.34%
getElement	14.00%
getObject	6.05%



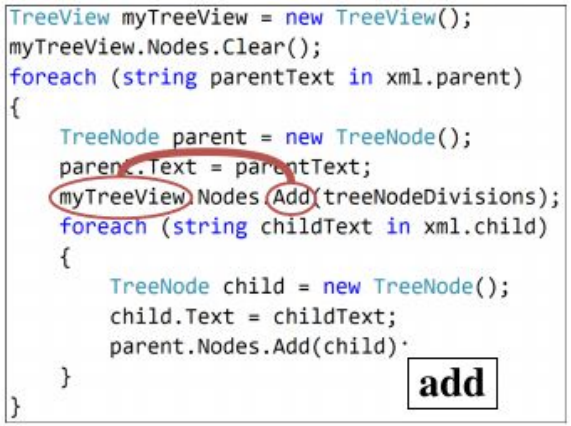
(c)

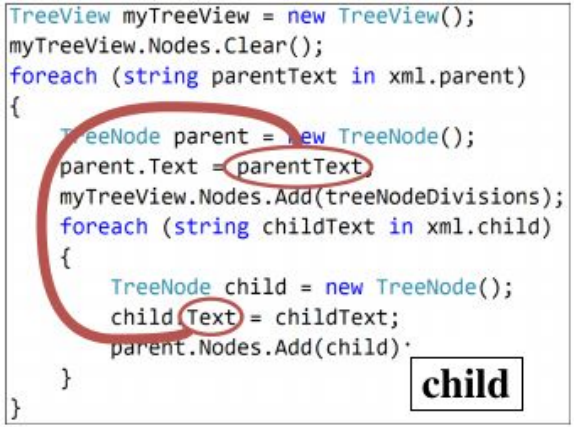
Predictions

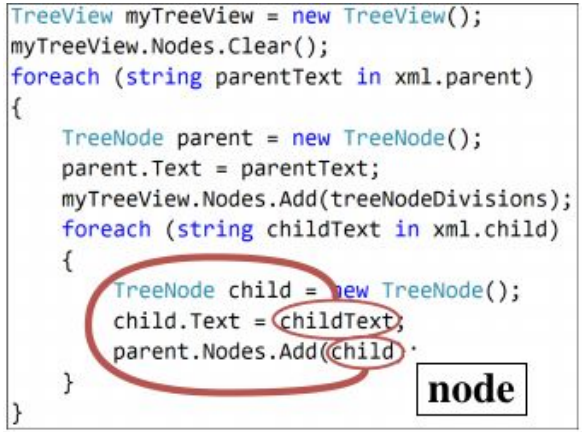
indexOf	96.65%
getIndex	2.24%
findIndex	0.33%
indexOfNull	0.20%
getInstructionIndex	0.13%

Больше примеров на https://code2vec.org/

Code2Seq / Пример суммаризации и главные пути







Сгенерированное описание:

add a child node to a treeview in c#

Больше примеров на https://code2seq.org/

Подводя итоги

Подводя итоги

- Роботизация наступает

Подводя итоги

- Роботизация наступает
- Если последние 20 лет все учились пользоваться IDE...

Подводя итоги

- Роботизация наступает
- Если последние 20 лет все учились пользоваться IDE...

  то можно ещё поучиться ещё лет 10 

Спасибо за внимание!

Будем рады ответить на ваши вопросы.

Telegram / slack: alsmirn, tyapochkinks.

hello@codescoring.com

https://profiscope.io

CODE SCORING
IT Risk Management