

# Автоматическое именованние методов в ООП

Алла Левина

# Как называют методы?

- Начинается с глагола и выполняет то, что говорит его имя ([The Java® Language Specification](#))
- Имя класса следует избегать в имени метода
- Начинаются с маленькой буквы
- CamelCase (в python snake\_case)

*Wrong names-* void getExpensesHistory(), int setMyAge(), line.getLineLength()

*Correct names-* List getExpensesHistory(), void setMyAge(int age), line.getLength()

# Как использовать глаголы?

Общие рекомендации <http://geosoft.no/development/javastyle.html>

- *get/set* при прямом доступе к атрибутам
- *is, has, can, should* булевы методы
- *compute* производятся какие-то вычисления
- *find* что-то ищется

Конкретные рекомендации <http://blog.joda.org/2011/08/common-java-method-names.html>

- *check* проверяет верно ли что-то и если нет, то бросает эксепшн
- *contains* проверяет, содержит ли коллекция объект
- *put* мутатор; добавляет или заменяет пару key-value
- *build* создает объект на основе аргументов или указанного целевого объекта (или оба)
- *reset* возвращает объект к начальному состоянию

<https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CodingGuidelines/Articles/NamingMethods.html>

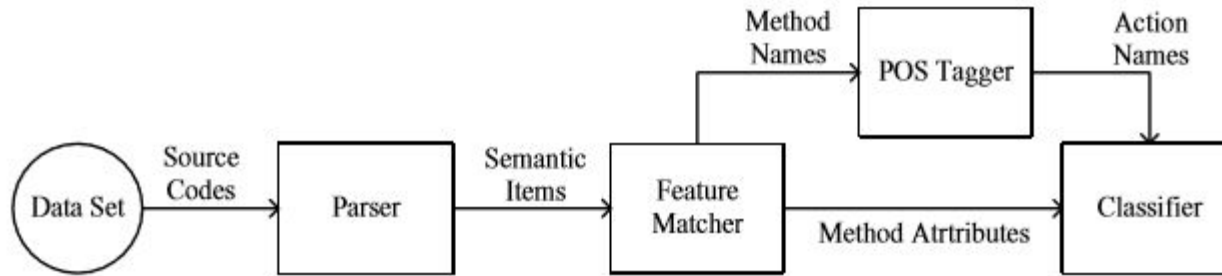
- Не использовать “do”/ “does”

# Properly and Automatically Naming Java Methods: A Machine Learning Based Approach

Shusi Yu, Ruichang Zhang, Jihong Gao

Springer, 2012

# RoadMap



# Инструменты

- AST
  - [http://www.antlr.org/](http://wwwantlr.org/)
- POS Tagger
  - <https://gate.ac.uk/>
  - Рассматривались только глагольные методы
  - Понятие глагола расширялось (*size()*)
- Классификатор
  - SVM (библиотека Libsvm)
  - precision

# Feature Matcher: modeling methods

Signature features	
Returns void	Returns boolean
Returns digit	Returns string
Returns object	Has parameters
Has boolean parameters	Has digit parameters
Has string parameters	Has string parameters
	Returns field value
Content features	
Reads field	Writes field
Accesses argument	Accesses boolean
Accesses digit	Accesses string
Accesses object	Accesses array
Creates string	Creates array
Declares variable	Creates object
Reads constant	Casts variable
For loop	While loop
If branch	Switch branch
Return points	Throws exceptions
Catches exceptions	Exposes checked exceptions
Recursive Method call	Method call on field value
Method call on parameter value	
Sub-verb features	
Methods called with certain action names	

# Feature Matcher: modeling variables

Signature features	
Is parameter	Is attribute
Is return value	Is boolean
Is digit	Is string
Is object	
Content features	
Being assigned	Being accessed
If branch condition	If branch body
Switch branch condition	Switch branch body
For loop condition	For loop body
While loop condition	While loop body
Try branch body	Catch branch body
Sub-method parameter	Sub-method return value
Sub-verb features	
Sub-method verb, as a parameter	
Sub-method verb, as a return value	

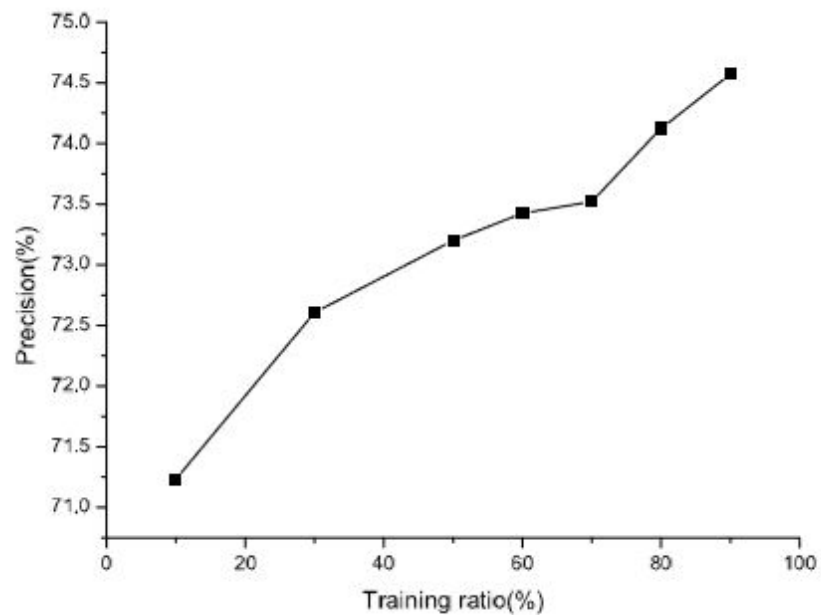
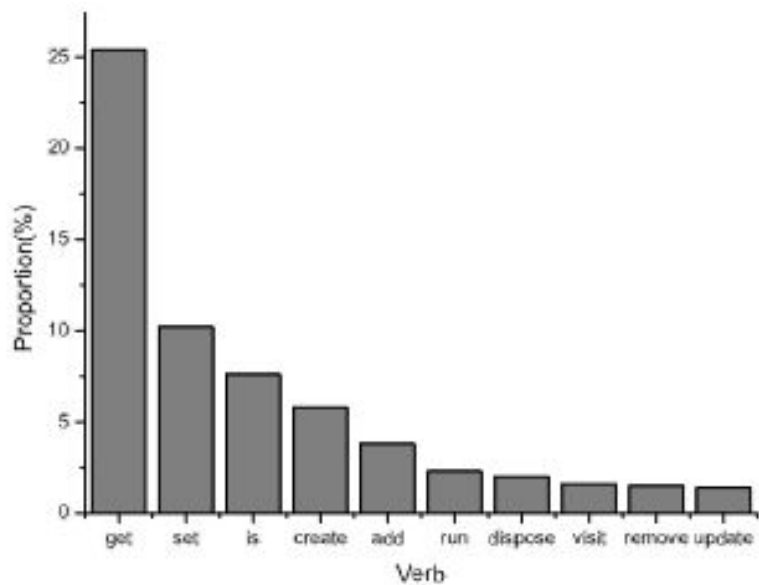


# Данные

- Eclipse
- 101015 МЕТОДОВ

# Verb Namer

- 237 глаголов



# Target Namer

getAllProperties  get (verb) + AllProperties (target)

- Таргет - атрибут класса, параметр/переменная метода
  - *full-target* переменная совпадает с таргетом (AllProperties)
  - *semi-target* переменная совпадает с частью таргета (Properties)
  - *non-target*
- Названия методов
  - 10% только глагол
  - < 10% *none-target*
- Атрибуты/параметры/переменные
  - 12.7% *full-target*
  - 15.1% *semi-target*
  - 70% *non-target*

# Target Namer: эксперименты

- бинарный классификатор про каждую переменную решает, таргет она или нет
- Full-target 93.6%
- Full-Semi 97.1%
- Semi-target 90%

# Результаты

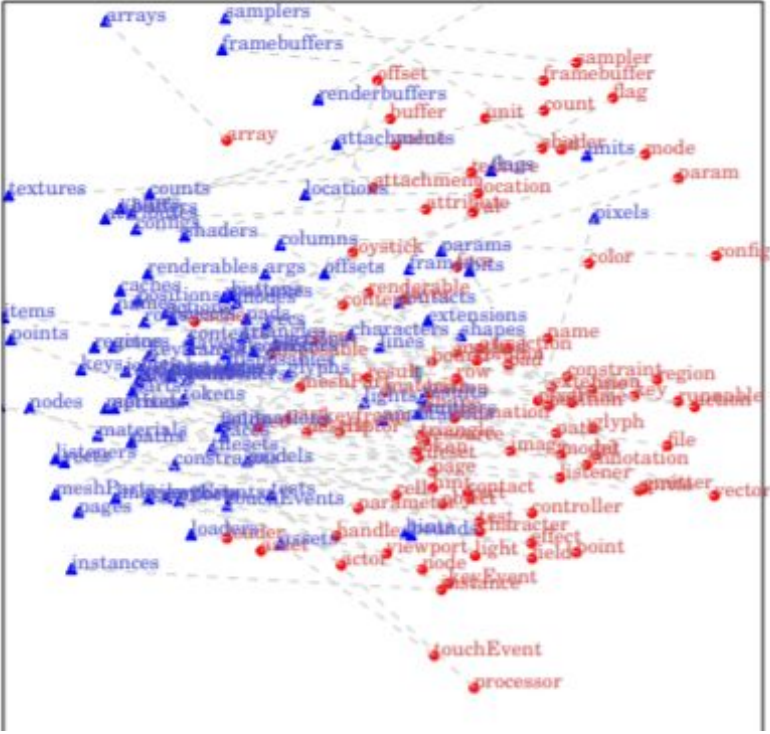
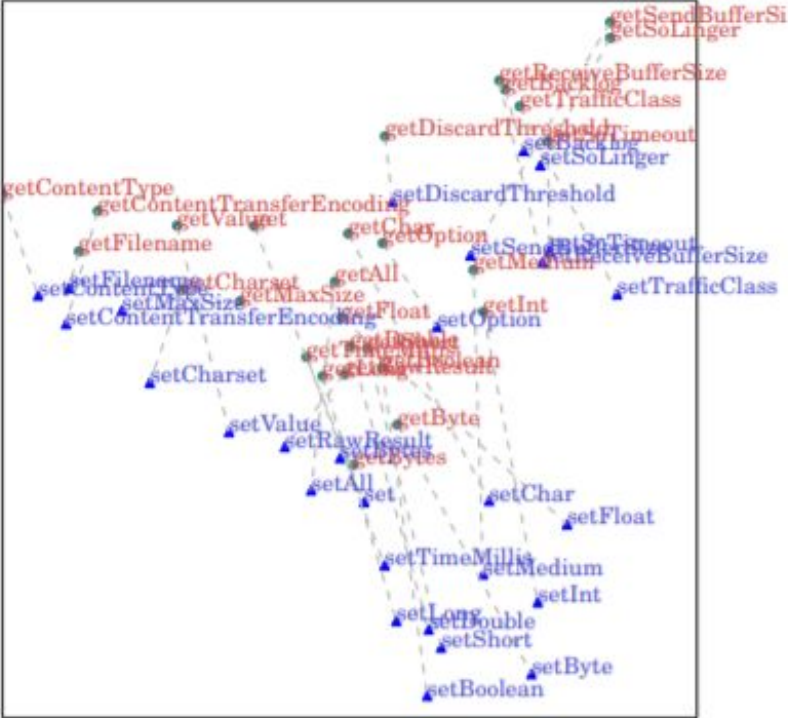
- Векторное пространство, описывающее функциональность и структуру Java методов и пространства переменных.
- Классификатор для автоматического именования методов в Java.
- Для оценки подхода проводилось множество экспериментов на датасете проекта Eclipse. Они показали, что для 70% методов точность предсказания правильного глагола достигает 70%. Точность предсказания таргета достигает 90%.

# Что еще делают люди?

- [An Approach for Evaluating and Suggesting Method Names using N-gram Models](#), 2014
  - 1000 проекта, 2,939,445 метода
  - 3-граммная языковая модель
  - оценка “понятности” названия (75%)
  - саджест следующее слово (92%)
- [Method verb recommendation using association rule mining in a set of existing projects](#), 2015
  - method-name, return-type, argument-type, argument-name, field-name, call-name
  - 445 проектов для экстракции правил
  - 82,101 правил
  - 230 глаголов, 192 глагола из трейна
  - 97 переименованных метода

		correct verbs				sum	
		in top 10		out of top 10	out of the list		
origin verbs	in top 10	higher 3	lower 5	12	5	25	
	out of top 10	9		higher 9	lower 4	8	30
	out of the list	9		20		13	42
	sum	26		45		26	97

# Suggesting Accurate Method and Class Names, 2015



# Инструменты

- Python AST
- Stanford NLP, NLTK
- Scikit-learn, XGBoost



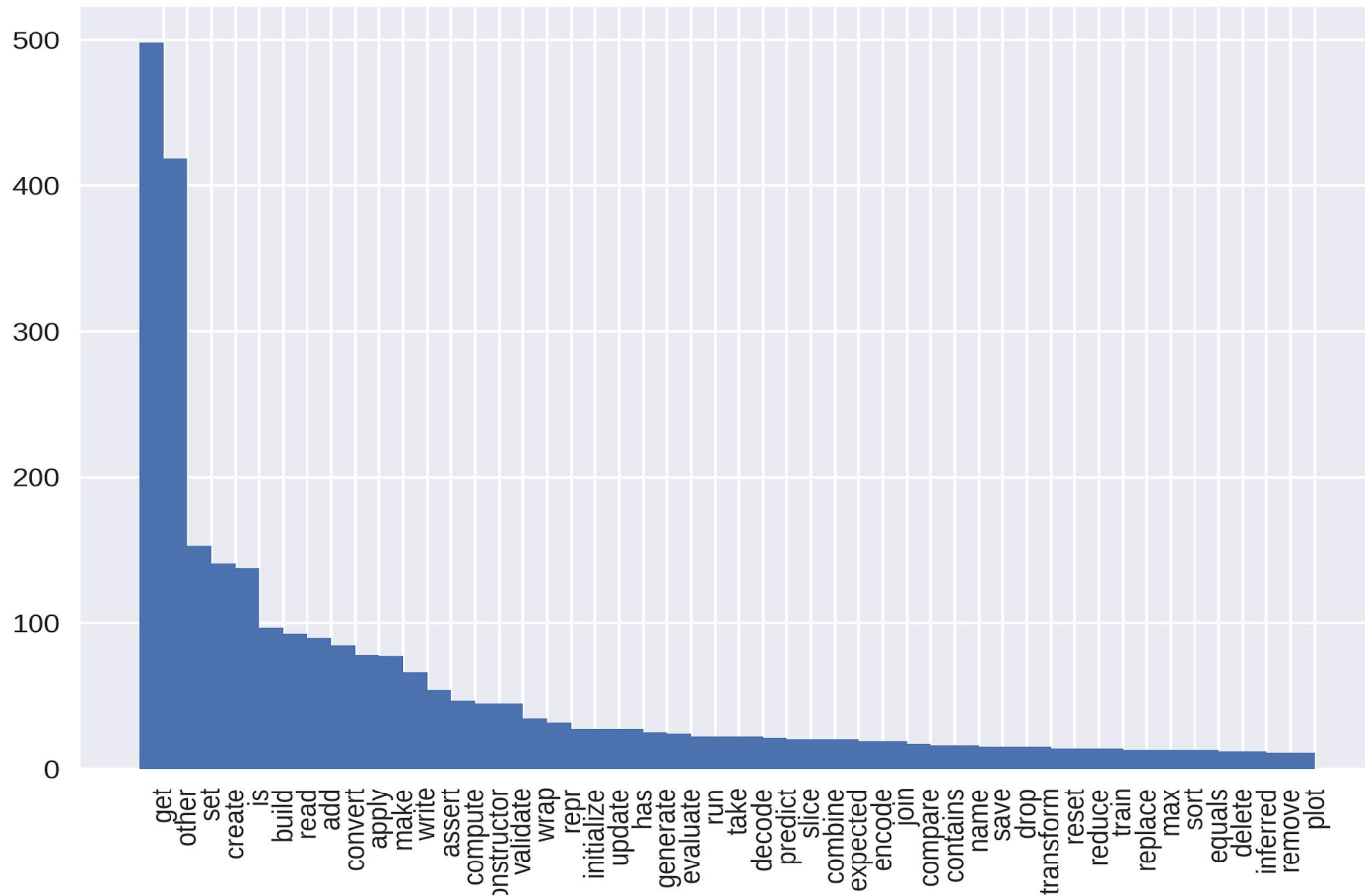
# Данные

Проект	Файлов	Методов	Глагольных методов
TensorFlow	1020	4268	967
Pandas	675	14024	1775

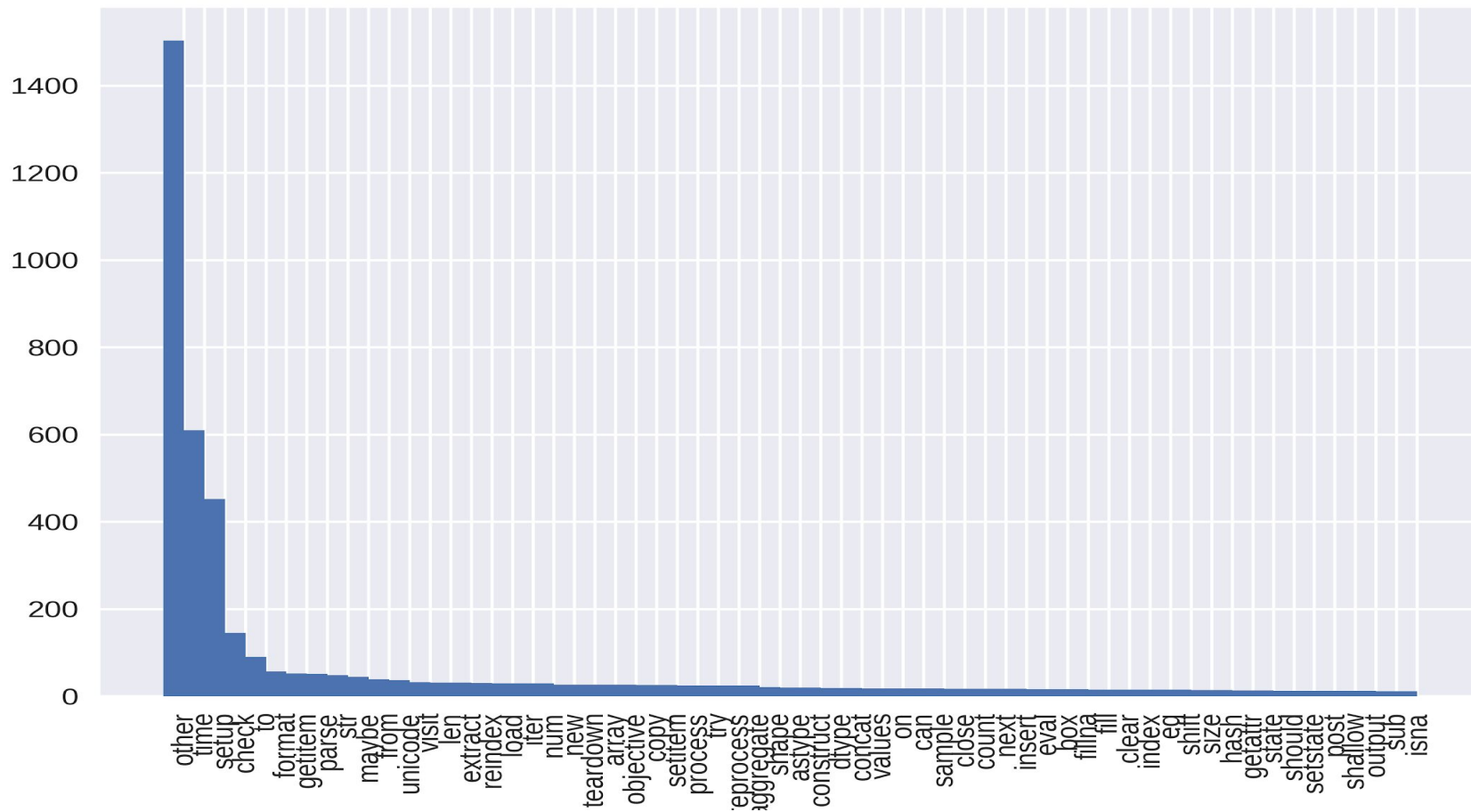
**NB:** test 10515

init 710

# Распределение глаголов



# Распределение не глаголов



# Фичи AST

- Ноды - 86 типов

'\_ast.arg', '\_ast.Name', '\_ast.Attribute', '\_ast.Name' ...

- Тело метода

Node\_i - количество нод в теле метода

- Return

Дамми фичи - какой тип у ноды Return (может быть 'NoneType')

- Общее количество нод

# Фи́чи NLP

$\text{bag}[V] = \{\text{все слова из методов с глаголом } V\}$

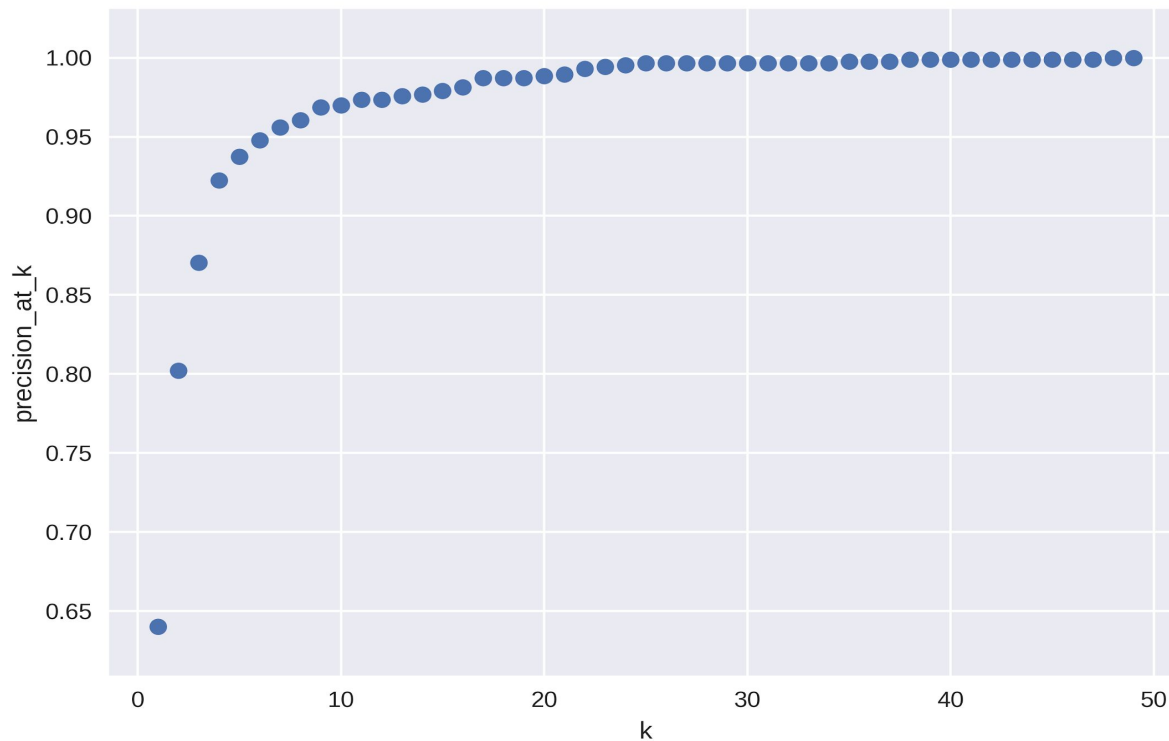
$\text{influence}[V] = \{w: w \in \text{bag}[V] \text{ and } P(w|V)/P(w|\text{not } V) > k\}$

$V\_influence[M] = \#\{w: w \in M \text{ and } w \in \text{influence}[V]\}$

# Результаты

precision\_at\_k: правильный ответ в top-k предсказаниях

49 классов



# Планы

- Улучшить определение частей речи
- Предсказание таргетов
- Посмотреть на нейросетевые подходы
- Улучшить векторизацию кода
- tf-idf от AST
- Типизация