Przetak:
mniej kąkolu w Sieci

Marcin Ciura

31 maja 2019 r.

Przedmowa

Ze *Słownika języka polskiego* Samuela Bogumiła Lindego

KĄKOL rośnie między wszelkim zbożem, ziarna jego ludziom i zwierzętom szkodliwe

PRZETAK gatunek rzeszot czyli durszlaków, z większemi dziurami czworograniastemi z łyka, z drótów etc.

Przez przetak puścić = cenzorować, krytykować, kritisiren

Przetak i kąkol

Przetak by Jckowal - Praca własna, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=20006225

Kąkol polny Agatka Anonim [CC BY-SA 4.0
(https://creativecommons.org/licenses/by-sa/4.0)]

Motyla noga Tomka Mazura (1)

Tomek: Ale nasypało, motyla noga! Już od kwadransa nie ma autobusu! Kurcze pióro!

Dziewczynka: Nie do wiary! Jak można się tak brzydko wyrażać?!

Tomek: A co mam mówić?! Jak mi nogi zmarzły... Motyla noga...

Wujek: No cóż! Widzicie, klimat był raczej zawsze przeciwko nam, no ale to jeszcze nie powód, aby mówić brzydkie wyrazy, prawda?

Chłopiec: Właśnie! My to też mu tłumaczymy.

Motyla noga Tomka Mazura (2)

Wujek:	Dam wam dobrą radę: kiedy następnym razem znów wyłączą wam ciepłą wodę, przestaną grzać kaloryfery albo stanie komunikacja i wasz kolega znów zacznie mówić brzydkie wyrazy, wiecie co zróbcie wtedy?
Dzieci:	Cooo?
Wujek:	Udawajcie, że nie słyszycie, co do was mówi. Że nic nie słyszycie.
Dzieci:	Huraaaa!
Dziewczynka:	To jest wspaniała rada!
Chłopiec:	Pycha!
Chłopiec II:	Wyobrażam sobie jego minę!
Dziewczynka:	Świetna rada, Wujku. Bardzo dobra rada!

W sieciowej grze komputerowej *League of Legends* (1)

In our observations and research on player behavior, we find that a single source of negative behavior can ripple through hundreds or thousands of games. For example, let's say we have a game with 10 players — 9 are positive, and 1 is negative. The 1 negative player is racist, rages all game, and intentionally leaves the game. This experience can negatively influence some of the other 9 positive players in the game. Some of these 9 players will play another game of League of Legends and instead of being positive, they might start the game neutral or negative. Their actions can then influence 9 other players in the game, and toxicity spreads.

W sieciowej grze komputerowej *League of Legends* (2)

Nieoficjalnie wiadomo, że najwięcej toksycznych graczy pochodzi z Polski i Rosji

Przepisy

W Kodeksie wykroczeń

Art. 141. Kto w miejscu publicznym umieszcza nieprzyzwoite ogłoszenie, napis lub rysunek albo używa słów nieprzyzwoitych, podlega karze ograniczenia wolności, grzywny do 1500 złotych albo karze nagany.

W Kodeksie karnym (1)

Art. 257. Kto publicznie znieważa grupę ludności albo poszczególną osobę z powodu jej przynależności narodowej, etnicznej, rasowej, wyznaniowej albo z powodu jej bezwyznaniowości lub z takich powodów narusza nietykalność cielesną innej osoby, podlega karze pozbawienia wolności do lat 3.

W Kodeksie karnym (2)

Art. 133. Kto publicznie znieważa Naród lub Rzeczpospolitą Polską, podlega karze pozbawienia wolności do lat 3.

W Kodeksie karnym (3)

Art. 196. Kto obraża uczucia religijne innych osób, znieważając publicznie przedmiot czci religijnej lub miejsce przeznaczone do publicznego wykonywania obrzędów religijnych, podlega grzywnie, karze pozbawienia wolności albo pozbawienia wolności do lat 2.

W Kodeksie karnym (4)

Art. 212. § 1. Kto pomawia inną osobę, grupę osób, instytucję, osobę prawną lub jednostkę organizacyjną niemającą osobowości prawnej o takie postępowanie lub właściwości, które mogą poniżyć ją w opinii publicznej lub narazić na utratę zaufania potrzebnego dla danego stanowiska, zawodu lub rodzaju działalności, podlega grzywnie albo karze pozbawienia wolności.

W Kodeksie karnym (5)

Art. 216. § 1. Kto znieważa inną osobę w jej obecności lub choćby pod jej nieobecność, lecz publicznie lub w zamiarze, aby zniewaga do osoby tej dotarła, podlega grzywnie albo karze pozbawienia wolności.

Powinność

Zmniejszyć liczbę obelżywych i wulgarnych komentarzy

Podejście

Trzy niezależne modele statystyczne

Trzy rodzaje toksycznych wyrazów:

1. Wyrazy obelżywe, np. *oszołom*
2. Wyrazy wulgarne ubliżające, np. *ku**a*
3. Wyrazy wulgarne pochlebne, np. *za**biście*

Dane uczące

4 668 625 form wyrazów	pochodzących z Polimorfologika 2.1
876 021 form wyrazów	pochodzących z 2 586 303 zdań
10 539 form wyrazów	obelżywych
19 880 form wyrazów	wulgarnych ubliżających
230 form wyrazów	wulgarnych pochlebnych

Kroki budowania modeli (1)

```
def dereplicate(s):
  """Removes repeated characters from s.

  >>> dereplicate('córrreczkęę')
  'córeczkę'
  >>> dereplicate('inną')
  'iną'
  """
  return ''.join(ch for ch, _ in itertools.groupby(s))
```

Około 1,3% wyrazów zawiera podwójne litery (*nn*, *dd*, *ii* itp.)

Kroki budowania modeli (2)

```
D0 = dict(zip('ąćęłńóśźż', 'acelnoszz'))
D1 = dict(zip('ąćęłńśźżjk', 'acelnszzyq'))
D1.update({'ó': 'ou', 'u': 'oó', 'w': 'fv'})

def dediacritize(s, d):
  """Yields copies of s preserving/removing diacritics.

  >>> list(dediacritize('córeczkę', D0))
  ['córeczkę', 'coreczkę', 'córeczke', 'coreczke']
  """
  if not s: yield ''; return
  for tail in dediacritize(s[1:], d):
    yield s[0] + tail
    for head in d.get(s[0], ''):
      yield head + tail
```

Kroki budowania modeli (3)

Po `dediacritize()` około 0,006% wyrazów zawiera podwójne litery (*puśćcie*, *ideę*, *weźże* itp.)

```
def get_ngrams(s, n):
  """Yields the n-grams of #s# w/o repeated characters.

  >>> list(get_ngrams('córeczkę', 5))
  ['#córe', 'córec', 'órecz', 'reczk', 'eczkę', 'czkę#']
  >>> list(get_ngrams('zzeram', 5))
  ['#zera', 'zeram', 'eram#']
  """
  s = '#' + dereplicate(s) + '#'
  for i in range(max(1, len(s) - n + 1)):
    yield s[i:i+n]
```

Kroki budowania modeli (4)

```
def add_5grams(s, v, X, y):
  """Appends the 5-grams of #s# to X and v to y.

  Handles frequent/creative misspellings.
  """
  X.append(' '.join(ng for ng in get_ngrams(s, 5)))
  y.append(v)
  if 'qu' in s:
    add_5grams(s.replace('qu', 'q'), v, X, y)
  if re.search('ch[uóo][jy]', s):
    add_5grams(re.sub('ch([uóo][jy])', r'h\1', s), v, X, y)
```

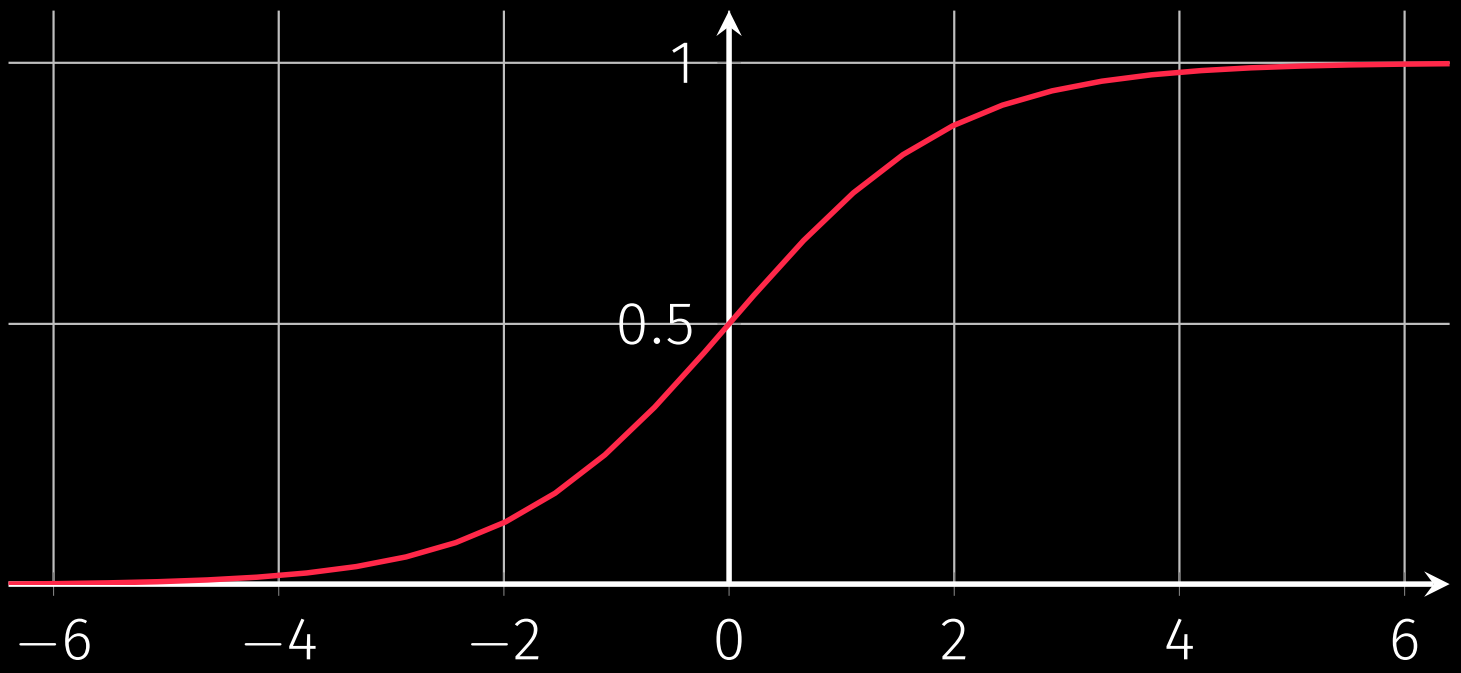
Kroki budowania modeli (5)

```
X, y = [], []
for word, toxicity in data:
  lw = word.lower()
  rw = dereplicate(lw)
  for dw in dediacritize(rw, (D0, D1)[toxicity]):
    add_5grams(dw, toxicity, X, y)
```

Kroki budowania modeli (6)

```
model = sklearn.pipeline.Pipeline([
    ('count vectorizer',
     sklearn.feature_extraction.text.CountVectorizer()),
    ('logistic regression',
     sklearn.linear_model.LogisticRegression(
         penalty='l1', C=200, tol=1e-7)),
])
model.fit(X, y)
```

Regresja logistyczna



Przykład

Problem Scunthorpe (1)

#zako	0,00	0,00	0,00
zakoc	0,00	0,00	0,00
akoch	0,00	0,00	0,00
kochu	0,00	-10,82	0,00
och*j	0,00	+13,53	0,00
ch*je	0,00	+2,59	0,00
h*je#	0,00	-0,43	0,00
przesunięcie	-20,03	-14,46	-18,76
#zakochuje#	-20,03	-9,16	-18,76
	< 0	< 0	< 0

Problem Scunthorpe (2)

#ch*j	0,00	+19,32	0,00
ch*je	0,00	+2,59	0,00
h*je#	0,00	-0,43	0,00
przesunięcie	-20,03	-14,46	-18,76
#ch*je#	-20,03	+7,02	-18,76
	<0	>0	<0

PolEval

Precyzja, czułość i miara F_1

- Precyzja (*precision*) — jaką część zniszczonych ziaren stanowi kąkol
- Czułość (*recall*) — jaką część kąkolu zniszczono
- F_1 = (precyzja × czułość)/(precyzja + czułość)

Wyniki

Program	F_1	Precyzja	Czułość
n-waves ULMFiT	58,58%	66,67%	52,24%
Przetak	57,98%	65,09%	51,49%
ULMFiT + SentencePiece + ...	53,68%	52,90%	54,48%
ensemble spacy + tpot + BERT	51,71%	52,71%	50,75%
Rafal-1	47,65%	41,08%	56,72%
model1-svm	45,58%	60,49%	36,57%
fasttext	41,35%	58,11%	32,09%
SCWAD-CB	38,50%	51,90%	30,60%
J.K.	22,57%	17,41%	32,09%

Przetak

https://github.com/MarcinCiura/przetak

Gotowy do użycia moduł wykrywający
obelżywe i wulgarne komentarze

https://github.com/MarcinCiura/przetak

Niestraszne mu:

- powielanie liiterrr
- r o z s t r z e l i w a n i e tekstu
- wstawianie nieliter po.mię.dzy litery
- zamienianie znaków na p0d0bnie wyglądające
- czętso popełniane literówki
- neologizmy o treści obelżywej lub wulgarnej

https://github.com/MarcinCiura/przetak

Sprawdza około 20 MB tekstu na sekundę

https://github.com/MarcinCiura/przetak

Można go używać na licencji Apache 2.0

https://github.com/MarcinCiura/przetak

Napisany w języku Go ze względu na:

- wygodę przetwarzania napisów w Unicode
- łatwość konsolidacji z programami napisanymi w innych językach
- obsługę systemów operacyjnych Windows, macOS i Linux

https://github.com/MarcinCiura/przetak

250 wierszy	kodu
1 600 wierszy	tablic zamiany podobnych znaków Unicode
12 800 wierszy	współczynników regresji logistycznej
250 wierszy	testów

https://github.com/MarcinCiura/przetak

Gotowe przykłady użycia:

- C
- C++
- Java
- Lua
- Node.js
- Perl 5
- Python 2 i 3
- R
- Ruby

Przyszłość

- Wykrywać wykropkowania wewnątrz wulgaryzmów
- Używać informacji zwrotnych z forów dyskusyjnych

Podsumowanie

15:1 Odpowiedź łagodna uśmierza gniew:
ale mowa przykra pobudza.

Przypisy

Przedmowa: *Miś* w reżyserii Stanisława Barei, scenariusz: Stanisław Tym; *Who invented the word "Toxic"?* na forum *League of Legends*

Przepisy: *Mowa nienawiści w internecie: jak z nią walczyć?* Materiały z konferencji pod redakcją Dominiki Bychawskiej-Siniarskiej i Doroty Głowackiej, Helsińska Fundacja Praw Człowieka, Warszawa 2013

Podsumowanie: *Biblia* w przekładzie ks. Jakuba Wujka