

word2Vec dla Polskiego Internetu

25 listopada 2015

Wojciech Stokowiec

`wojciech.stokowiec@opi.org.pl`

`http://opi-lil.github.io/`

Ośrodek Przetwarzania Informacji - Państwowy Instytut Badawczy



OŚRODEK PRZETWARZANIA INFORMACJI
PAŃSTWOWY INSTYTUT BADAWCZY

Agenda

word2Vec

- CBOW

- Skip-Gram

- Optymalizacje

 - Hierarchical Softmax

 - Negative Sampling

Internety

- Common Crawl

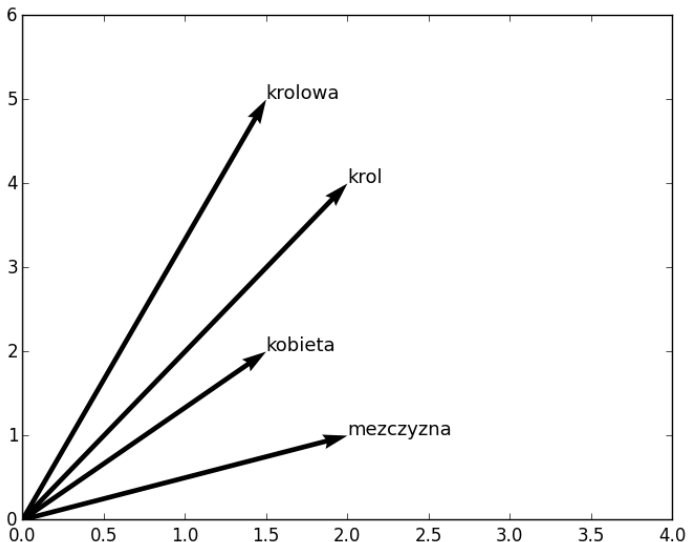
- Akka

Przykłady

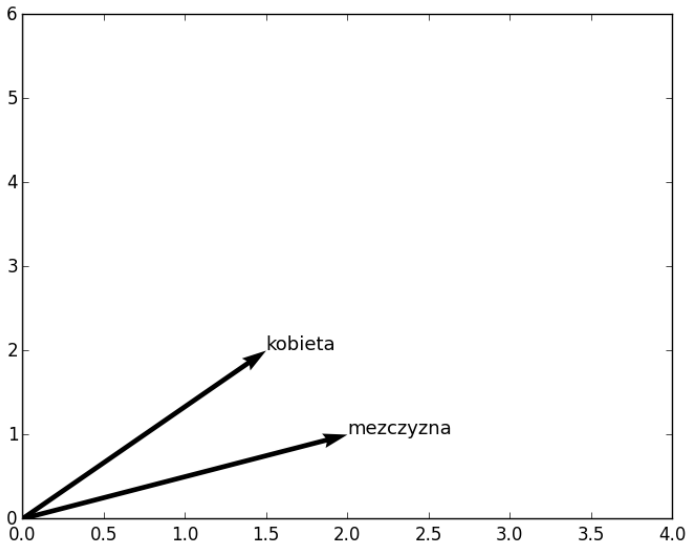
- Common Crawl

- Wiki

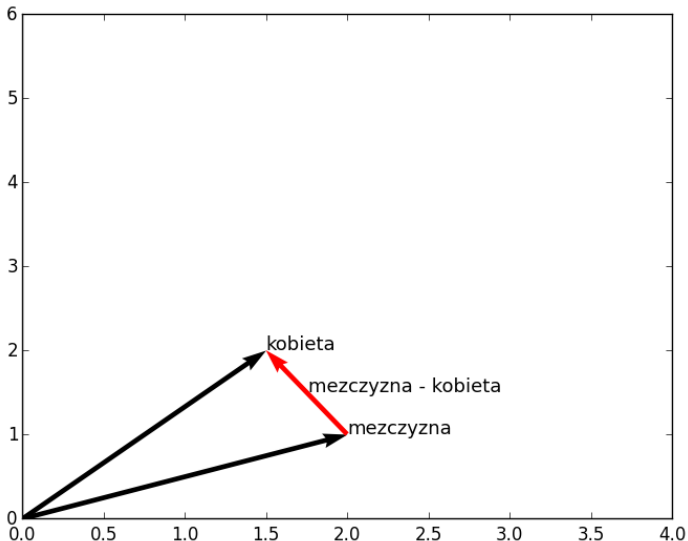
Przykład motywujący



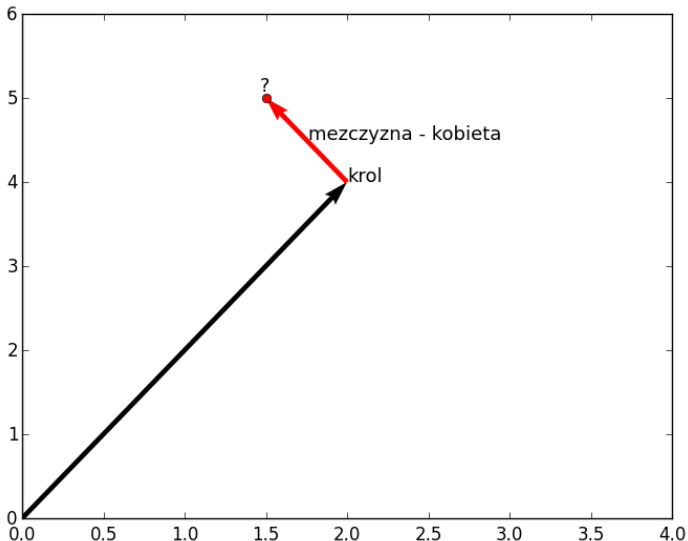
Przykład motywujący



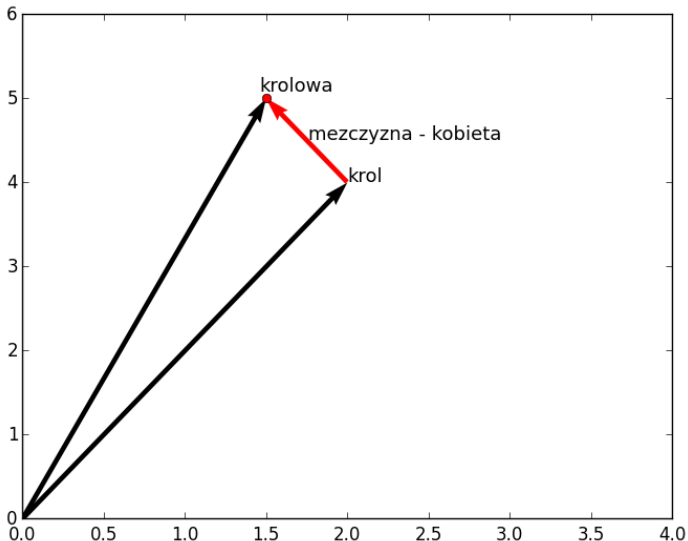
Przykład motywujący



Przykład motywujący



Przykład motywujący



Korzystając z gotowego modelu ze strony
<https://code.google.com/p/word2vec/>:

```
from gensim.models.word2vec import Word2Vec

model = Word2Vec.load_word2vec_format(
    'GoogleNews-vectors-negative300.bin',
    binary=True)

model.most_similar(positive=['woman', 'king'],
    negative=['man'], topn=5)

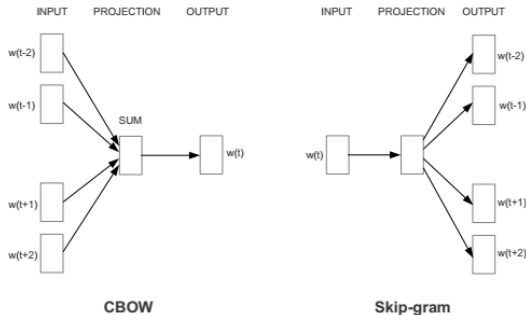
[(u'queen', 0.711819589138031),
 (u'monarch', 0.618967592716217),
 (u'princess', 0.5902432799339294),
 (u'crown_prince', 0.5499461889266968),
 (u'prince', 0.5377323031425476)]
```


Co ciekawe, poza semantycznymi relacjami, *word2vec* jest w stanie "wyłapać" podstawy gramatyki, takie jak stopniowanie przymiotników:

```
model.most_similar(positive=['biggest', 'small'],  
                    negative=['big'], topn=5)  
  
[(u'smallest', 0.6086569428443909),  
 (u'largest', 0.6007465720176697),  
 (u'tiny', 0.5387299656867981),  
 (u'large', 0.456944078207016),  
 (u'minuscule', 0.43401968479156494)]
```

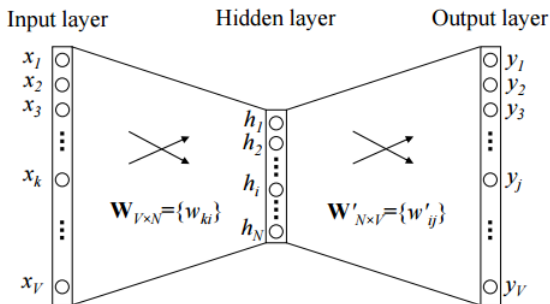
Należy pamiętać, że:

- ▶ *Word2vec* to nie jeden model!
- ▶ *Word2vec* to nie *deep learning*!



Rysunek: CBOW oraz Skip-gram

Rysunek: CBOW z jednym słowem w kontekście



Dla danego kontekstu \mathbf{x} , zakładając kodowanie 1 z N , tj $x_k = 1$ oraz $x_{k'} = 0$ dla $x_k \neq x_{k'}$ możemy obliczyć wartości warstwy ukrytej:

$$\mathbf{h} = \mathbf{x}^T \mathbf{W} = \mathbf{W}_{(k, \cdot)} := \mathbf{v}_{w_I} \quad (1)$$

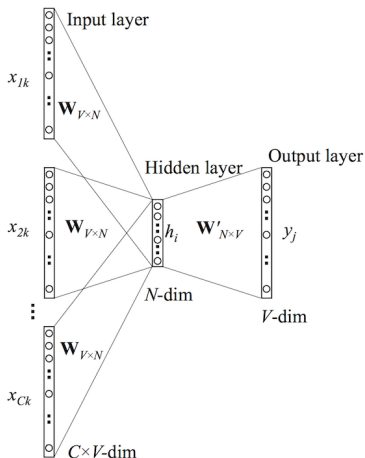
Następnie wyliczamy wartość funkcji oceny u_j dla każdego słowa w_j w słowniku:

$$u_j = \mathbf{v}'_{w_j}{}^T \cdot \mathbf{h}. \quad (2)$$

Aby wyznaczyć prawdopodobieństwo wyemitowania słowa w_j pod warunkiem zaobserwowania danego kontekstu korzystamy z funkcji softmax:

$$p(w_O \mid w_I) = \frac{\exp(u_O)}{\sum_{i=1}^V \exp(u_i)} = \frac{\exp\left(\mathbf{v}'_{w_O}{}^T \cdot \mathbf{v}_{w_I}\right)}{\sum_{i=1}^V \exp\left(\mathbf{v}'_{w_i}{}^T \cdot \mathbf{v}_{w_I}\right)}. \quad (3)$$

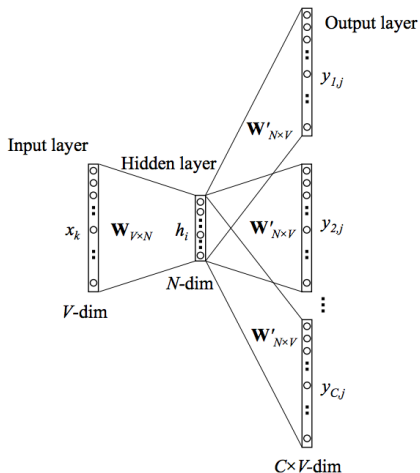
Rysunek: CBOW z dłuższym kontekstem



Analogicznie, tylko że warstwa ukryta wygląda w sposób następujący:

$$\begin{aligned} \mathbf{h} &= \frac{1}{C} \mathbf{W} \cdot (\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_C) \\ &= \frac{1}{C} \cdot (\mathbf{v}_{w_1} + \mathbf{v}_{w_2} + \dots + \mathbf{v}_{w_C}). \end{aligned} \tag{4}$$

Rysunek: Skig-gram



Skip-gram jest lustrzanym odbiciem architektury **CBOW**, tj. na podstawie słowa staramy się przewidzieć jego kontekst. Niech dany będzie ciąg słów: w_1, w_2, \dots, w_T oraz długość kontekstu c , wtedy staramy się maksymalizować następującą funkcję:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t), \quad (5)$$

a prawdopodobieństwa $p(w_{t+j}|w_t)$ zdefiniowane są w sposób następujący:

$$p(w_o|w_I) = \frac{\exp(v'_{w_o}{}^T v_{w_I})}{\sum_{w=1}^W \exp(v'_w{}^T v_{w_I})} \quad (6)$$

gdzie v_w oraz v'_w oznaczają "wejściowe" i "wyjściowe" reprezentacje wektorowe słowa "w", a W jest liczbą słów w słowniku.

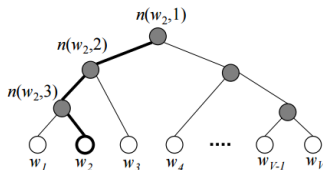
Przypomnijmy, że w wzorze 7 w mianowniku znajduje się czynnik normalizujący:

$$p(w_O|w_I) = \frac{\exp(v'_{w_O}{}^T v_{w_I})}{\sum_{w=1}^W \exp(v'_w{}^T v_{w_I},)} \quad (7)$$

- ▶ Przy dużym słowniku (a takie występują w przyrodzie) metody optymalizacyjne oparte na prostym gradiencie są co najmniej nieefektywne.
- ▶ Można sobie z tym radzić na parę sposobów!

- ▶ Jest to efektywny sposób na znalezienie szukanego prawdopodobieństwa
- ▶ Model używa drzewa binarnego do reprezentacji słów ze słownika

Rysunek: Przykład drzewa binarnego dla hierarchicznego softmax'u



Prawdopodobieństwo, że dane słowo w jest słowem wyjściowym zadane jest następującym wzorem:

$$p(w = w_O) = \prod_{j=1}^{L(w)-1} \sigma\left(\mathbb{I}[n(w, j+1) = \text{ch}(n(w, j))] \cdot \mathbf{v}'_{n(w, j)}{}^T \mathbf{h}\right) \quad (8)$$

Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne

Skąd wziąć to drzewo?

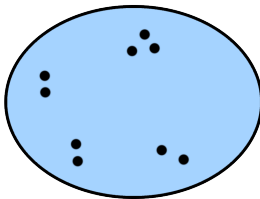
- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**

Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:

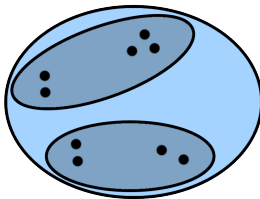
Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:



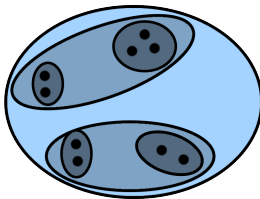
Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:



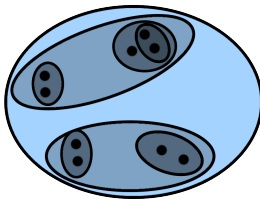
Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:



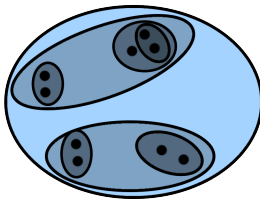
Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:



Skąd wziąć to drzewo?

- ▶ Użyć losowo wygenerowanego
 - ▶ Rozwiązanie w najlepszym przypadku nieoptymalne
- ▶ Można użyć zewnętrznych źródeł, np. **WordNet**
- ▶ Można użyć metod klastrowania hierarchicznego:



- ▶ Mikolov w swojej implementacji *Word2vec*'a używa drzew Huffmana

W swojej pracy z 2013 roku Mikolov używają następującej funkcji celu:

$$\log \sigma(v'_{w_O}{}^T v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(W)} [\log \sigma(-v'_{w_i}{}^T v_{w_I})], \quad (9)$$

Gdzie k , to liczba próbek z rozkładu szumu $P_n(W)$. Równanie 9 można zapisać w trochę czytelniejszy sposób:

$$\underbrace{\log \sigma(v'_{w_O}{}^T v_{w_I})}_{\text{Prawidłowy rozkład}} + \underbrace{\sum_{i \sim P_n(W)} \log \sigma(-v'_{w_i}{}^T v_{w_I})}_{\text{Rozkład szumu}} \quad (10)$$

W swojej pracy z 2013 roku Mikolov używają następującej funkcji celu:

$$\log \sigma(v'_{w_O}{}^T v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(W)} [\log \sigma(-v'_{w_i}{}^T v_{w_I})], \quad (9)$$

Gdzie k , to liczba próbek z rozkładu szumu $P_n(W)$. Równanie 9 można zapisać w trochę czytelniejszy sposób:

$$\overbrace{\log \sigma(v'_{w_O}{}^T v_{w_I})}^{\text{Prawidłowy rozkład}} + \overbrace{\sum_{i \sim P_n(W)} \log \sigma(-v'_{w_i}{}^T v_{w_I})}^{\text{Rozkład szumu}} \quad (10)$$

- maksymalizujemy prawdopodobieństwo wystąpienia rzeczywistego kontekstu

W swojej pracy z 2013 roku Mikolov używają następującej funkcji celu:

$$\log \sigma(v'_{w_O}{}^T v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(W)} [\log \sigma(-v'_{w_i}{}^T v_{w_I})], \quad (9)$$

Gdzie k , to liczba próbek z rozkładu szumu $P_n(W)$. Równanie 9 można zapisać w trochę czytelniejszy sposób:

$$\overbrace{\log \sigma(v'_{w_O}{}^T v_{w_I})}^{\text{Prawidłowy rozkład}} + \overbrace{\sum_{i \sim P_n(W)} \log \sigma(-v'_{w_i}{}^T v_{w_I})}^{\text{Rozkład szumu}} \quad (10)$$

- ▶ maksymalizujemy prawdopodobieństwo wystąpienia rzeczywistego kontekstu
- ▶ minimalizujemy prawdopodobieństwo wystąpienia losowych słów w kontekście

W swojej pracy z 2013 roku Mikolov używają następującej funkcji celu:

$$\log \sigma(v'_{w_O}{}^T v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(W)} [\log \sigma(-v'_{w_i}{}^T v_{w_I})], \quad (9)$$

Gdzie k , to liczba próbek z rozkładu szumu $P_n(W)$. Równanie 9 można zapisać w trochę czytelniejszy sposób:

$$\overbrace{\log \sigma(v'_{w_O}{}^T v_{w_I})}^{\text{Prawidłowy rozkład}} + \overbrace{\sum_{i \sim P_n(W)} \log \sigma(-v'_{w_i}{}^T v_{w_I})}^{\text{Rozkład szumu}} \quad (10)$$

- ▶ maksymalizujemy prawdopodobieństwo wystąpienia rzeczywistego kontekstu
- ▶ minimalizujemy prawdopodobieństwo wystąpienia losowych słów w kontekście
- ▶ $P_n(w) \sim U(w)^{3/4}/Z$

“Fundacja Common Crawl to organizacja non-profit której celem jest demokratyzacja dostępu do informacji zawartej w internecie poprzez stworzenie i utrzymanie otwartego repozytorium tekstowych danych internetowych, które są powszechnie dostępne i łatwe do analizy.”

— strona fundacji Common Crawl

- ▶ Mnóstwo danych!
- ▶ Jeden dump "waży" około 140TB danych i zawiera 1.80 mld stron internetowych
- ▶ Około 17 dumpów, zrobionych na przestrzeni lat 2013-2015
- ▶ Każdy dump składa się z plików:
 - ▶ **WARC** - zawierających nieobrobione dane
 - ▶ **WAT** - zawierających meta-dane opisujące dany rekord
 - ▶ **WET** - zawierających wyłuskany ze strony tekst
- ▶ Jeden dump zawiera około 10TB danych w formacie WET
- ▶ Do tej pory przetworzyliśmy 9 dumpów, co daje około 90TB danych tekstowych, ale ...

- ▶ Mnóstwo danych!
- ▶ Jeden dump "waży" około 140TB danych i zawiera 1.80 mld stron internetowych
- ▶ Około 17 dumpów, zrobionych na przestrzeni lat 2013-2015
- ▶ Każdy dump składa się z plików:
 - ▶ **WARC** - zawierających nieobrobione dane
 - ▶ **WAT** - zawierających meta-dane opisujące dany rekord
 - ▶ **WET** - zawierających wyłuskany ze strony tekst
- ▶ Jeden dump zawiera około 10TB danych w formacie WET
- ▶ Do tej pory przetworzyliśmy 9 dumpów, co daje około 90TB danych tekstowych, ale ...
- ▶ ... około 0.3% jest w języku polskim.

Format WET zawiera minimalną ilość meta-danych, główną jego zawartością jest czysty tekst ze strony.

```
WARC/1.0
WARC-Type: conversion
WARC-Target-URI: http://news.bbc.co.uk/2/hi/africa/3414345.stm
WARC-Date: 2014-08-02T09:52:13Z
WARC-Record-ID:
WARC-Refers-To:
WARC-Block-Digest: sha1:JROHLC55SKMBR6XY46WXREW7RXM64EJC
Content-Type: text/plain
Content-Length: 6724

BBC NEWS | Africa | Namibia braces for Nujoma exit
...
President Sam Nujoma works in very pleasant surroundings in the small but
```

Rysunek: Przykład pliku w formacie WET



Obowiązkowe Hello World (cz. 1):

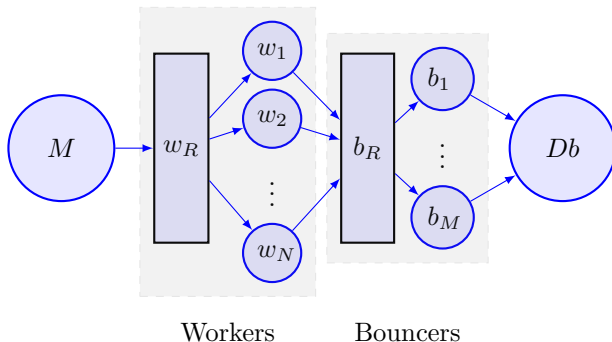
```
// definiujemy protokół rozmowy
case class Hello(who: String)

// minimalny aktor
class Greeter extends Actor {
  def receive = {
    case Hello(who) => println(s"It's a you, a $who!")
    case _          => println("Name, please!")
  }
}
```

Obowiązkowe Hello World (cz. 2):

```
object Main extends App {  
  
  // inicjalizacja systemu aktorow  
  val system = ActorSystem("HelloWorldSystem")  
  
  // stworzenie aktora  
  val greeter = system.actorOf(Props[Greeter],  
                                name = "helloactor")  
  
  // przeslanie wiadomosci  
  greeter ! "Mario!"  
  
}
```

- ▶ System aktorów do zarządzania aplikacją
- ▶ 1 File Master tworzący File Workerów oraz rozdzielający im prace
- ▶ 1 Terminator monitorujący cykl życia aktorów, oraz zamykający system
- ▶ 24 File Workerów do przetwarzania strumienia danych oraz wysyłania kawałków tekstu reprezentujących stronę do identyfikacji języka
- ▶ 36 Bouncerów
 - ▶ filtrują teksty z języka polskiego przy pomocy biblioteki CLD2
 - ▶ zapis do Cassandra



Rysunek: Architektura crawlera

Tablica: Najbliższe słowa

Słowo	Najbliższy Wektor	Dystans
Król	Cesarz	0.7349
Tusk	Donald	0.7382
Kobieta	Dziewczyna	0.7998
Mężczyzna	Chłopak	0.84556
Sushi	Pizza	0.75798
Apple	Tablety	0.78932
Dziewczyna	Rozochocona :-)	0.81743
Kaczyński	Palikot :-)	0.83625

Tablica: Algebra wektorów

Wyrażenie	Najbliższy wektor
Król – Mężczyzna + Kobieta	Edyp :-)
Większy – Duży + Mały	Mniejszy
Włochy – Rzym + Francja	Paryż
Dżungla + Król	tarzantarzan lewkról

Tablica: Najbliższe słowa

Słowo	Najbliższy Wektor	Dystans
Król	Władca	0.61166
Tusk	Ramotar :-)	0.54940
Kobieta	Dziewczyna	0.74277
Mężczyzna	Chłopak	0.70107
Sushi	Chowder	0.52896
Apple	Iphone	0.6675
Dziewczyna	Kobieta	0.7428
Kaczyński	Wałęsa Kwaśniewski Komorowski	0.83625

Tablica: Algebra wektorów

Wyrażenie	Najbliższy wektor
Król – Mężczyzna + Kobieta	Królowa
Większy – Duży + Mały	Mniejszy
Włochy – Rzym + Francja	Szwajcaria :-)

- Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.
- ▶ Pracujemy nad deduplikacją, korektą błędów i odświeżaniem.

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.
- ▶ Pracujemy nad deduplikacją, korekcją błędów i odświeżaniem.
- ▶ Wektorowe reprezentacje słów uzyskane poprzez uczenie word2veca na naszym korpusie są przesycone seksem :-)

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.
- ▶ Pracujemy nad deduplikacją, korekcją błędów i odświeżaniem.
- ▶ Wektorowe reprezentacje słów uzyskane poprzez uczenie word2veca na naszym korpusie są przesycone seksem :-)
- ▶ Wektorowe reprezentacje słów na polskiej wiki są nieznacznie lepsze.

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.
- ▶ Pracujemy nad deduplikacją, korekcją błędów i odświeżaniem.
- ▶ Wektorowe reprezentacje słów uzyskane poprzez uczenie word2veca na naszym korpusie są przesycone seksem :-)
- ▶ Wektorowe reprezentacje słów na polskiej wiki są nieznacznie lepsze.
- ▶ Zastosowanie word2veca do uzupełniania leksykonów sentymentu w zagadnieniu analizy wydźwięku

- ▶ Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Common Crawl, ale ...
- ▶ ... to straszny śmietnik.
- ▶ Pracujemy nad deduplikacją, korektą błędów i odświeżaniem.
- ▶ Wektorowe reprezentacje słów uzyskane poprzez uczenie word2veca na naszym korpusie są przesycone seksem :-)
- ▶ Wektorowe reprezentacje słów na polskiej wiki są nieznacznie lepsze.
- ▶ Zastosowanie word2veca do uzupełniania leksykonów sentymentu w zagadnieniu analizy wydźwięku
- ▶ Polska język, trudna język.

W razie jakichkolwiek uwag, komentarzy lub wątpliwości proszę o kontakt:

Wojciech Stokowiec
wojciech.stokowiec@opi.org.pl
<http://opi-lil.github.io/>

Dziękuję za uwagę!



OŚRODEK PRZETWARZANIA INFORMACJI
PAŃSTWOWY INSTYTUT BADAWCZY