word2vec dla Polskiego Internetu

27 listopada 2015

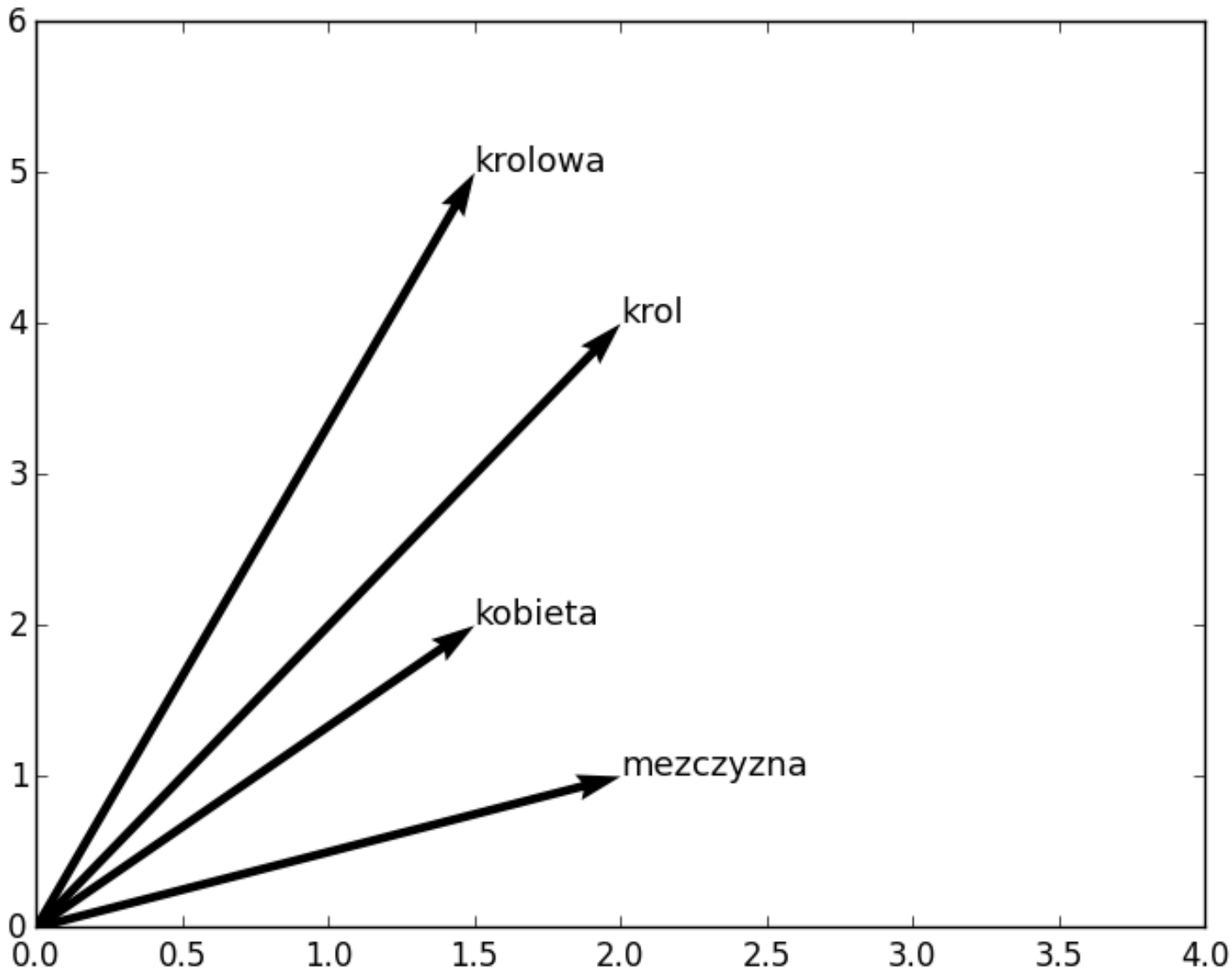
Wojciech Stokowiec
wojciech.stokowiec@opi.org.pl
http://opi-lil.github.io/

Ośrodek Przetwarzania Informacji - Państwowy Instytut Badawczy

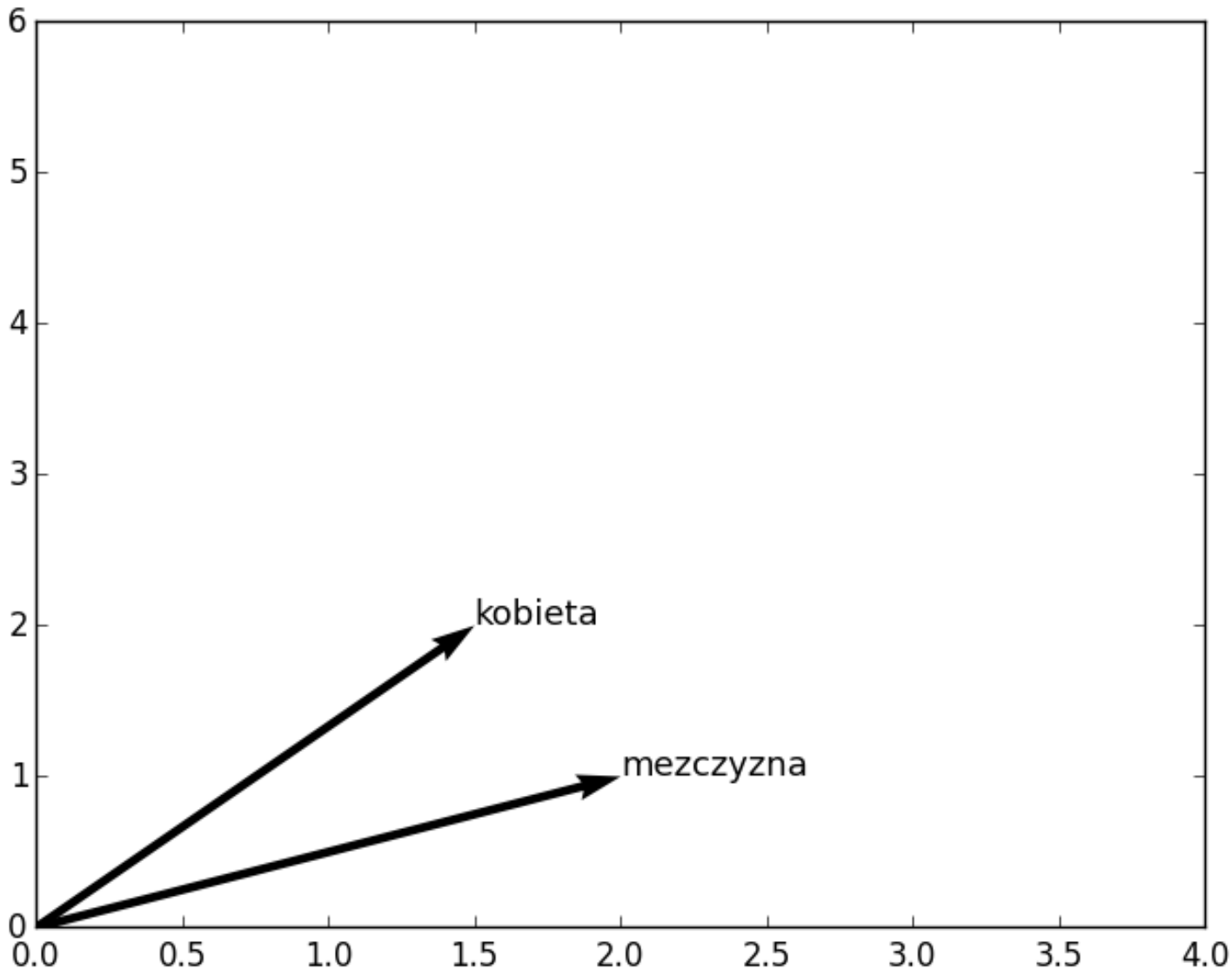
Agenda

- word2vec
  - CBOW
  - Skip-Gram
  - Optymalizacje
    - Hierarchical Softmax
    - Negative Sampling
- Internety
  - Common Crawl
  - Akka
- Przykłady
  - Common Crawl
  - Wiki

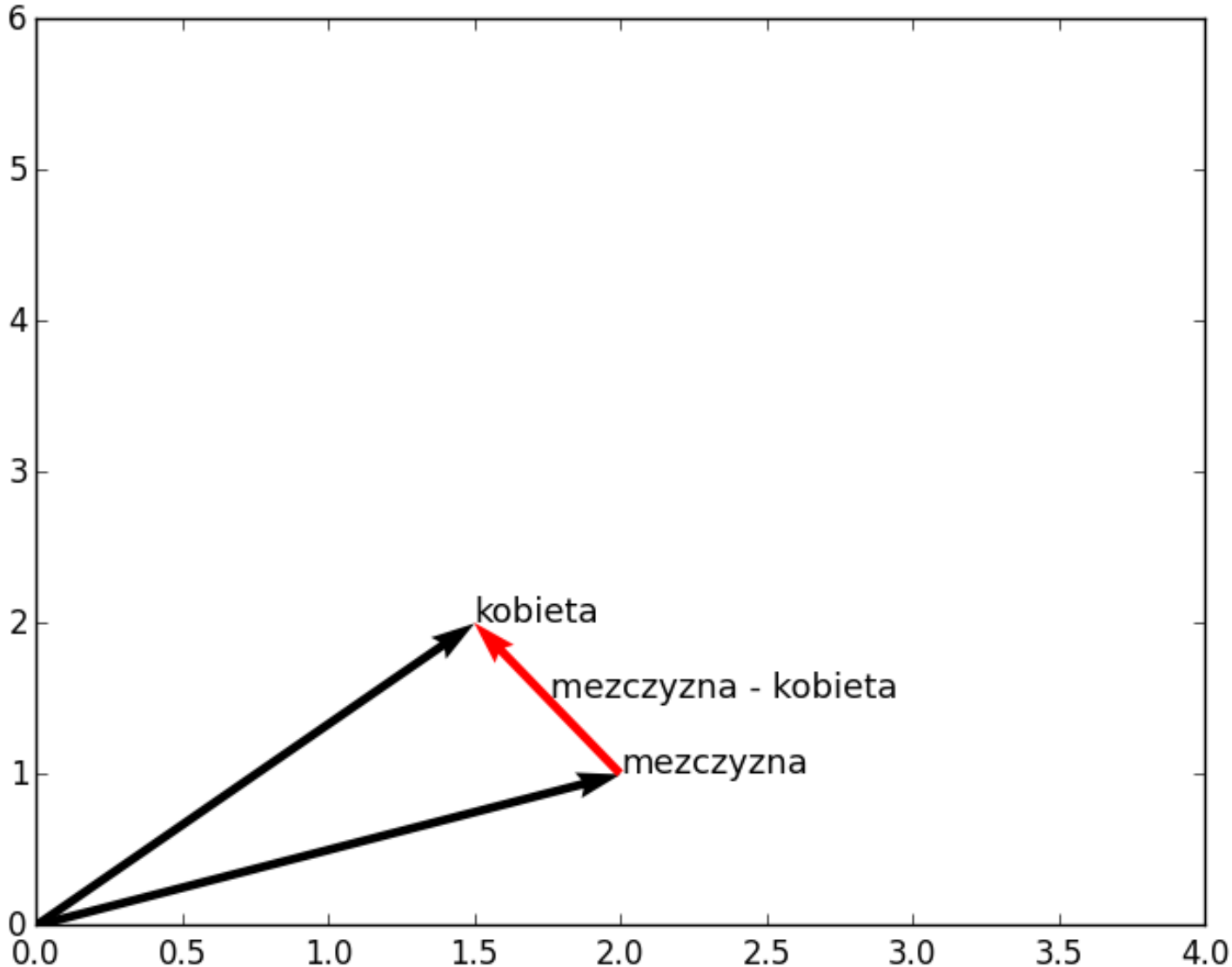
Przykład motywujący



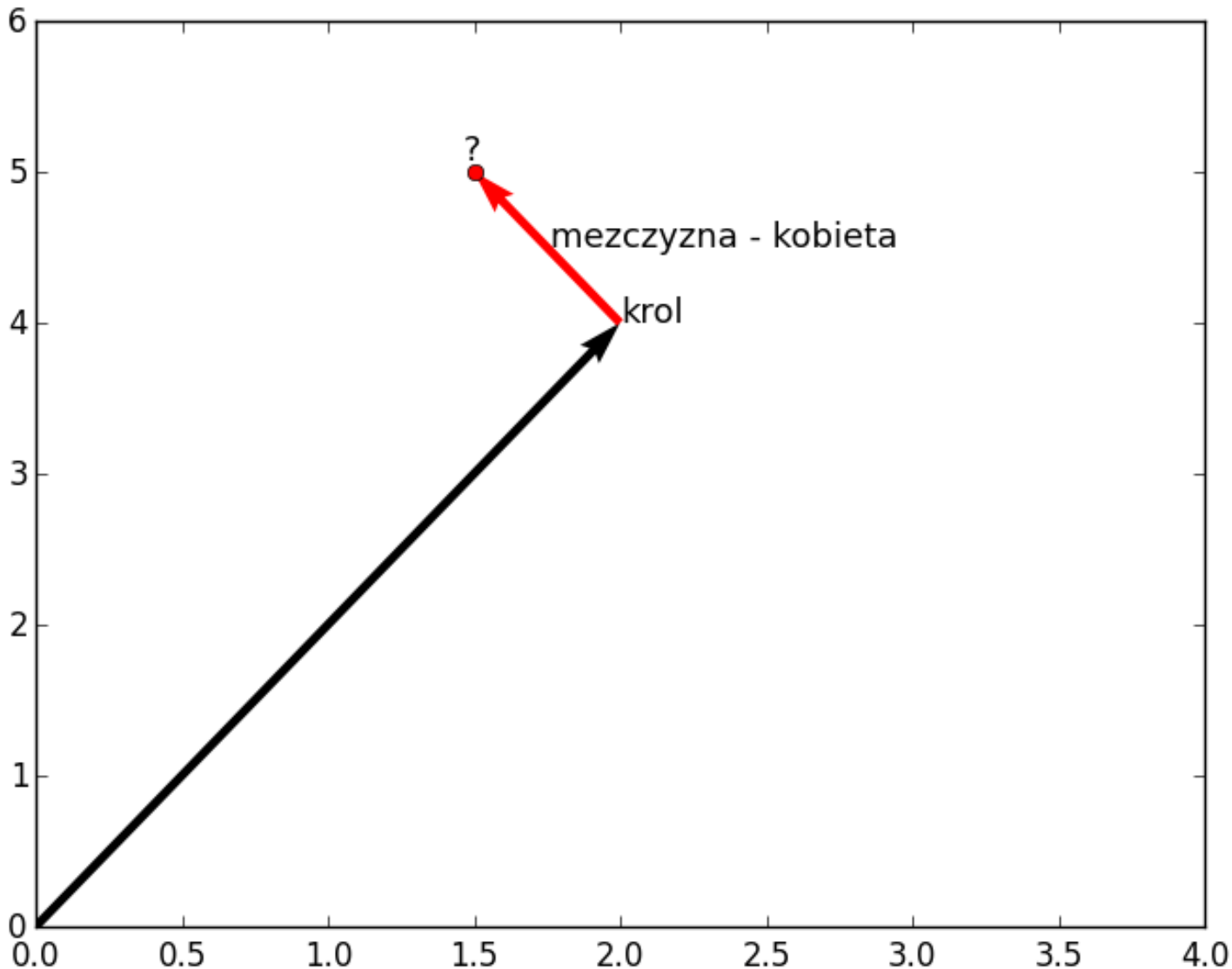
Przykład motywujący



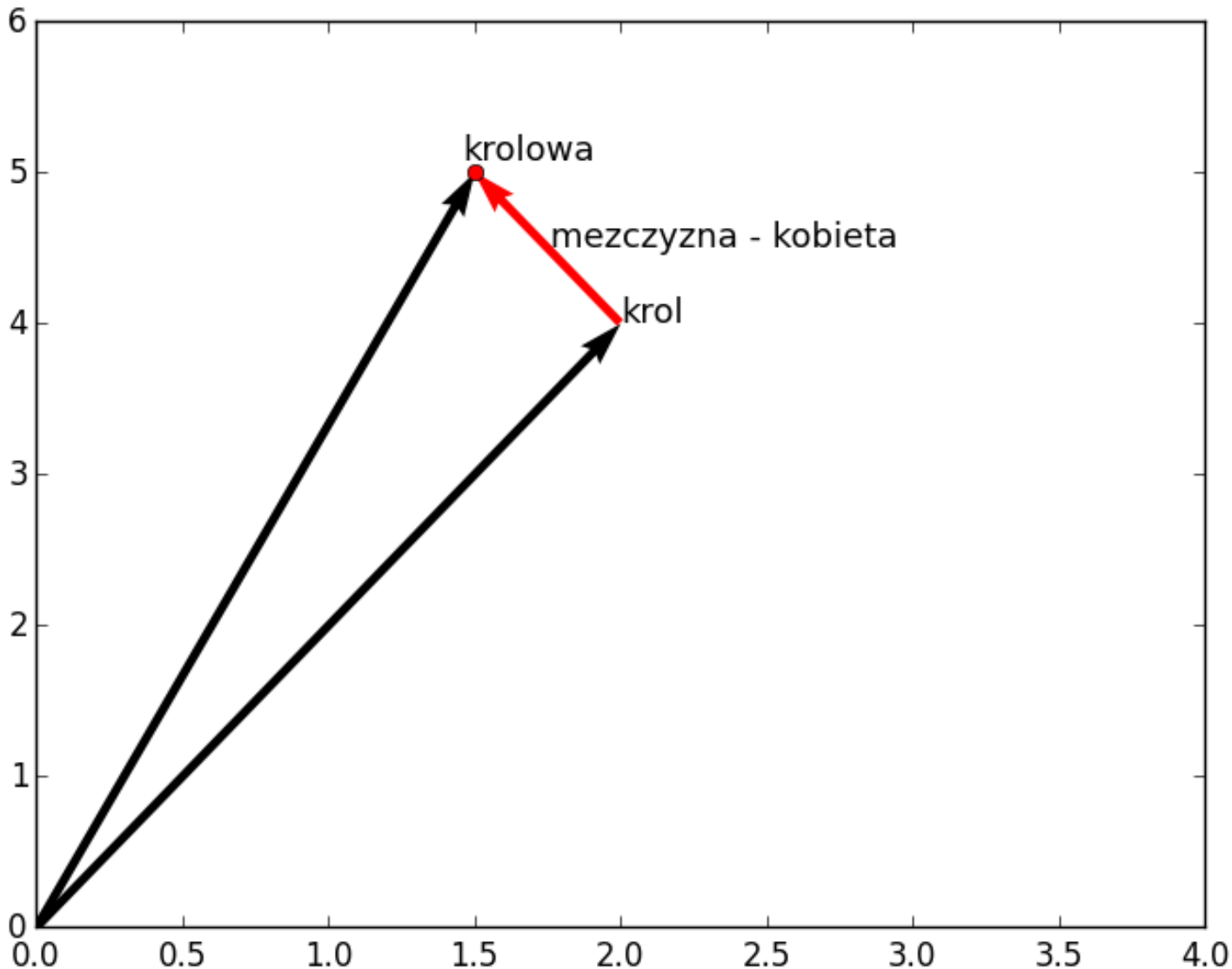
Przykład motywujący



Przykład motywujący



Przykład motywujący



word2vec + Google News

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', 0.590243279933929 4),
 (u'crown_prince', 0.5499461889266968),
 (u'prince', 0.5377323031425476)]
```

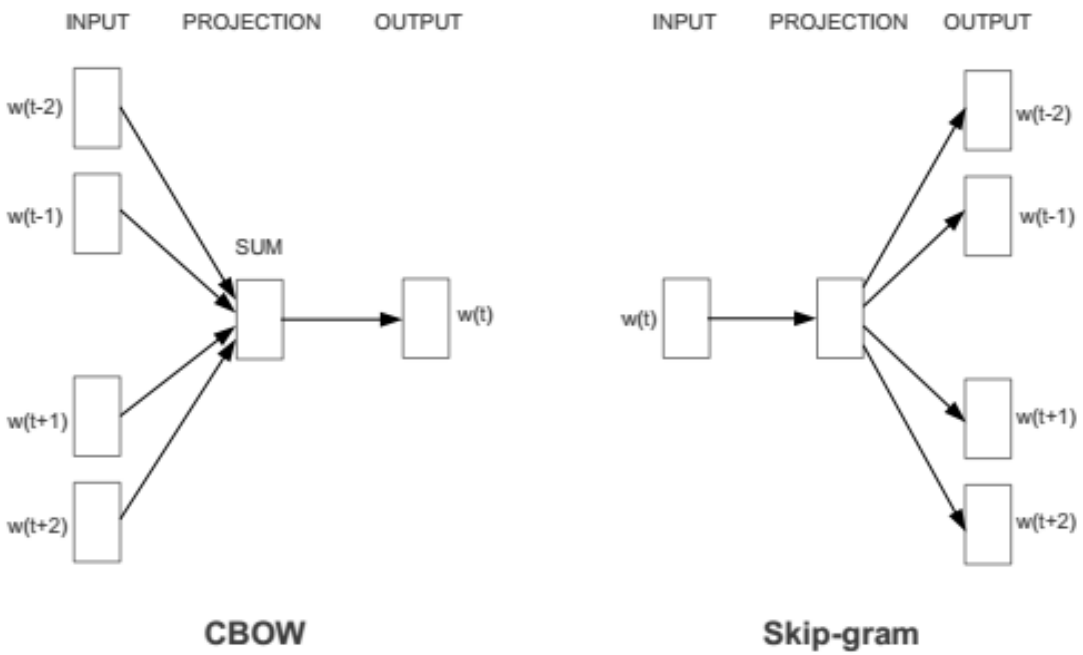
word2vec + Google News

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:

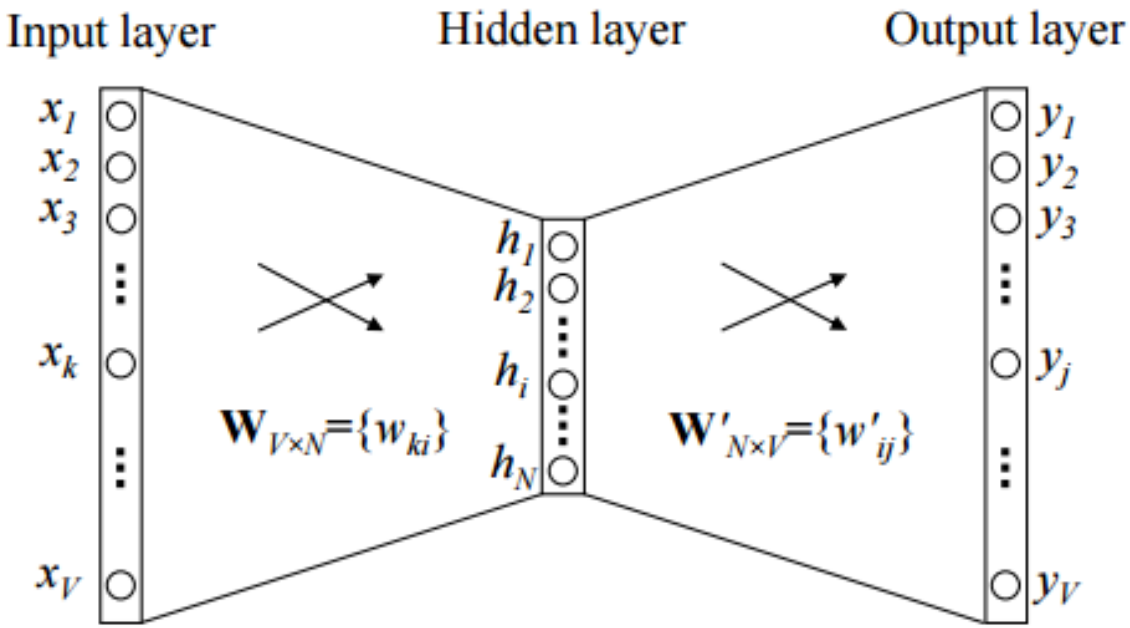
- *word2vec* to nie jeden model!
- *word2vec* to nie *deep learning*!



Rysunek: CBOW oraz Skip-gram

Continuous Bag of Words

Rysunek: CBOW z jednym słowem w kontekście



CBOW

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:

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

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

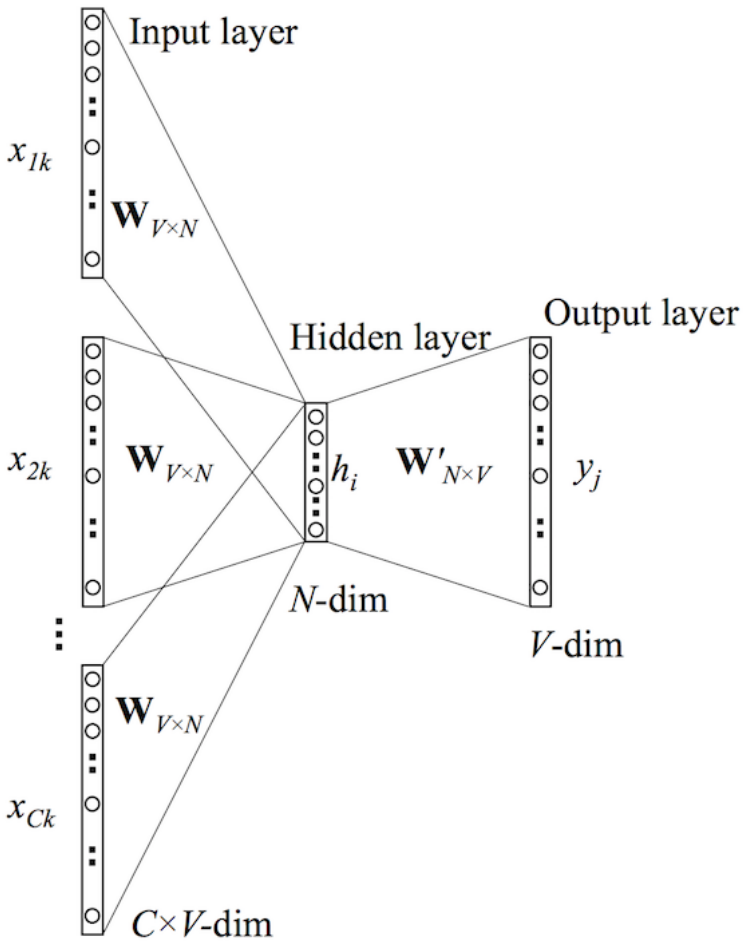
$$u_j = {\mathbf{v}'_{w_j}}^T \cdot \mathbf{h}. \tag{2}$$

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

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

CBOW

Rysunek: CBOW z dłuższym kontekstem



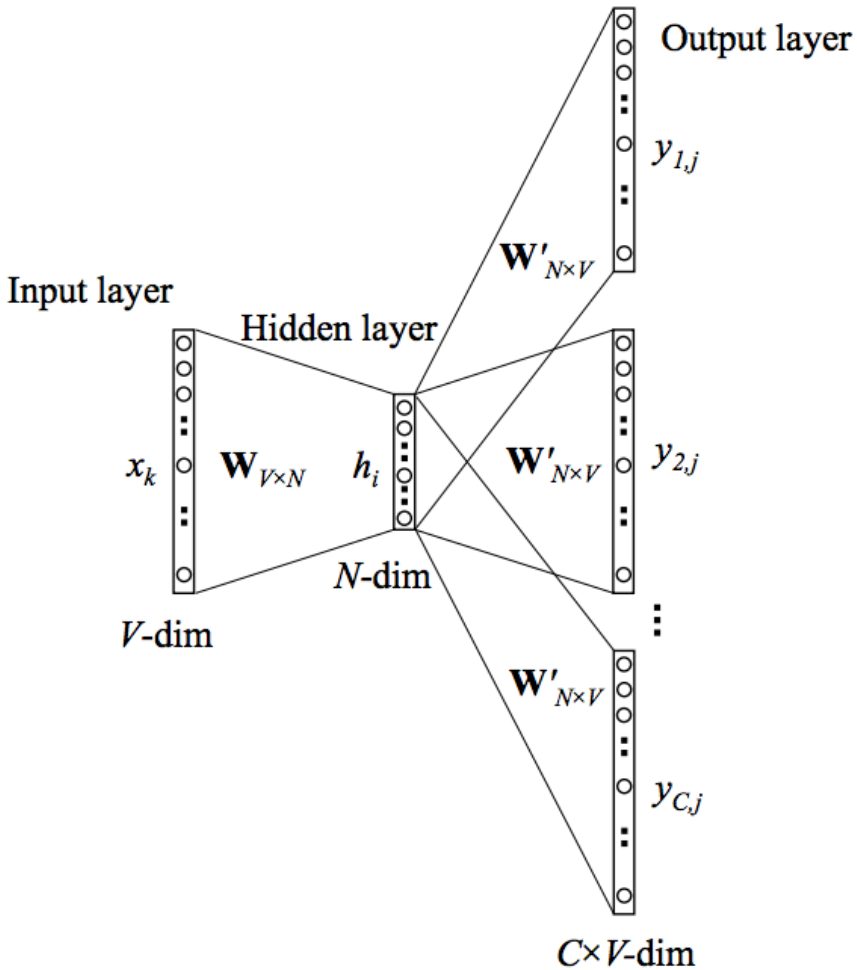
CBOW

Analogicznie, tylko że warstwa ukryta wygląda w sposób następujący:

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

Skip-Gram

Rysunek: Skig-gram



Skip-Gram

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

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

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

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^T v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^T v_{w_I},\right)} \tag{6}$$

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

Optymalizacje

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

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

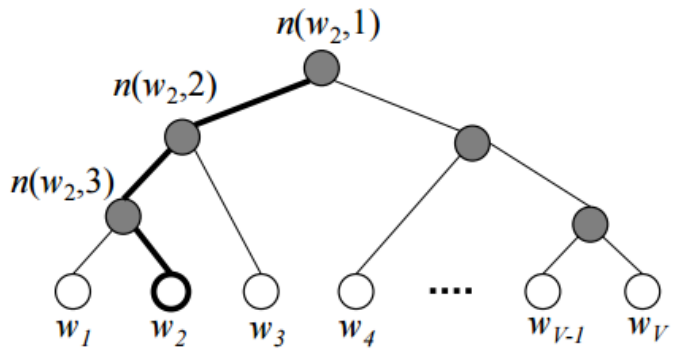
- Przy dużym słowniku (a takie występują w przyrodzie) metody optymalizacyjne oparte na prostym gradiencie są co najmniej nieefektywne.
- Można sobie z tym radzić na parę sposobów!

Hierarchical Softmax

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

Hierarchical Softmax

Rysunek: Przykład drzewa binarnego dla hierarchicznego sofmax'u



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

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

Hierarchical Softmax

Skąd wziąć to drzewo?

- Użyć losowo wygenerowanego
  - Rozwiązanie w najlepszym przypadku nieoptymalne

Hierarchical Softmax

Skąd wziąć to drzewo?

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

Hierarchical Softmax

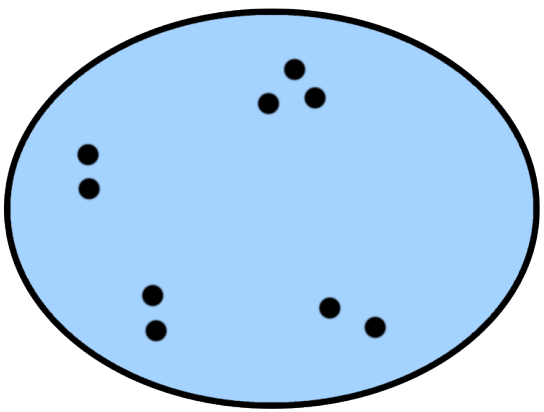
Skąd wziąć to drzewo?

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

Hierarchical Softmax

Skąd wziąć to drzewo?

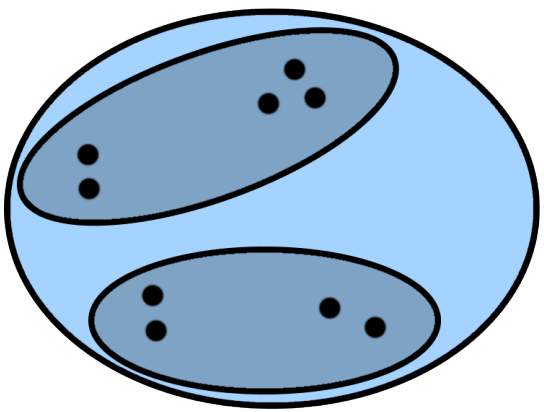
- Użyć losowo wygenerowanego
  - Rozwiązanie w najlepszym przypadku nieoptymalne
- Można użyć zewnętrznych źródeł, np. **WordNet**
- Można użyć metod klastrowania hierarchicznego:



Hierarchical Softmax

Skąd wziąć to drzewo?

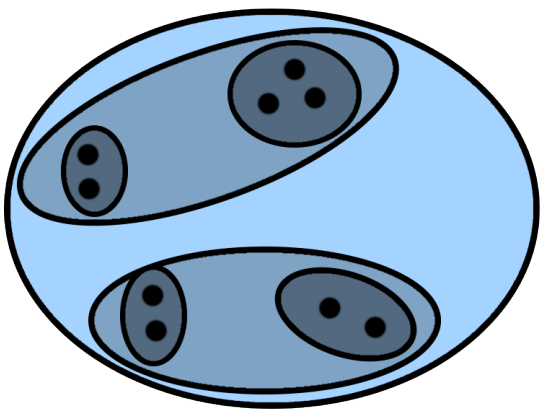
- Użyć losowo wygenerowanego
  - Rozwiązanie w najlepszym przypadku nieoptymalne
- Można użyć zewnętrznych źródeł, np. **WordNet**
- Można użyć metod klastrowania hierarchicznego:



Hierarchical Softmax

Skąd wziąć to drzewo?

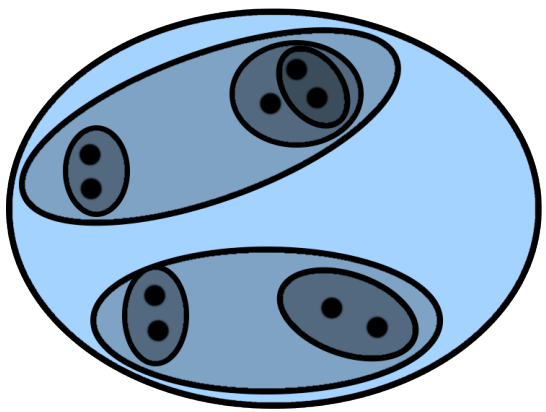
- Użyć losowo wygenerowanego
  - Rozwiązanie w najlepszym przypadku nieoptymalne
- Można użyć zewnętrznych źródeł, np. **WordNet**
- Można użyć metod klastrowania hierarchicznego:



Hierarchical Softmax

Skąd wziąć to drzewo?

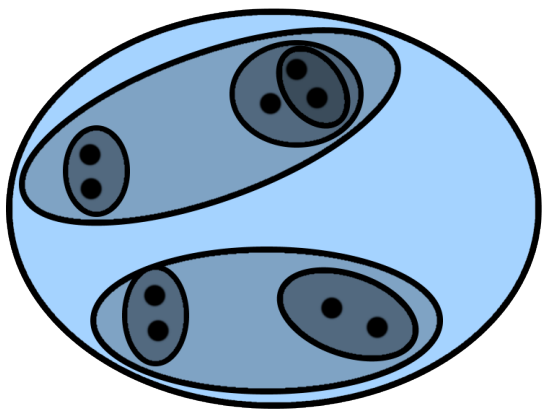
- Użyć losowo wygenerowanego
  - Rozwiązanie w najlepszym przypadku nieoptymalne
- Można użyć zewnętrznych źródeł, np. **WordNet**
- Można użyć metod klastrowania hierarchicznego:



Hierarchical Softmax

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

Negative Sampling

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

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

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

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

Negative Sampling

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

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

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

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

- maksymalizujemy prawdopodobieństwo wystąpienia rzeczywistego kontekstu

Negative Sampling

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

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

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

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

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

Negative Sampling

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

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

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

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

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

Common Crawl

"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

Common Crawl

- Mnóstwo danych!
- Jeden dump "ważyókoło 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 ...

Common Crawl

- Mnóstwo danych!
- Jeden dump "ważyókoło 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.

WET

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:JROHLCS5SKMBR6XY46WXREW7RXM64EJC
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

Narzędzia

- 

- 

- 

Akka

Obowiązkowe Hello World (cz. 1):

```
// definiujemy protokol 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!")
  }
}
```

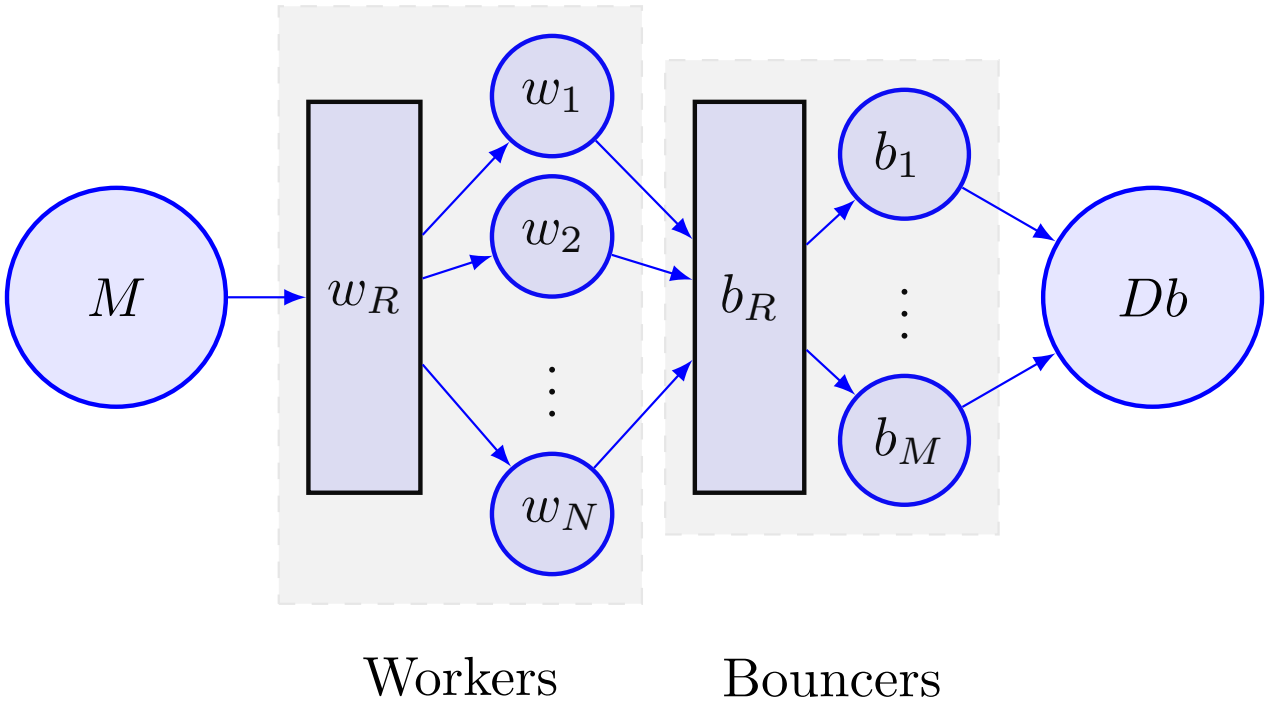
Obowiązkowe Hello World (cz. 2):

```
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!"

}
```

Nasi aktorzy

- 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 teksy z języka polskiego przy pomocy biblioteki CLD2
  - zapis do Cassandry

Nasi aktorzy



Rysunek: Architektura crawlera

word2vec na zbiorze Common Crawl

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

word2vec na zbiorze Common Crawl

Tablica: Algebra wektorów

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

word2vec na polskiej Wikipedii

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 | Kwaśnieswki | Komorowski	0.83625

Tablica: Algebra wektorów

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

Podsumowanie

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

Podsumowanie

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

Podsumowanie

- Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Commmon Crawl, ale ...
- ... to straszny śmietnik.
- Pracujemy nad deduplikacją, korekcją błędów i odśmiecaniem.

Podsumowanie

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

Podsumowanie

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

Podsumowanie

- Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Commmon Crawl, ale ...
- ... to straszny śmietnik.
- Pracujemy nad deduplikacją, korekcją błędów i odśmiecaniem.
- 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

Podsumowanie

- Udało się stworzyć największy korpus języka polskiego filtrując zbiory fundacji Commmon Crawl, ale ...
- ... to straszny śmietnik.
- Pracujemy nad deduplikacją, korekcją błędów i odśmiecaniem.
- 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.

Uwagi
Informacje kontaktowe

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

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

Dziękuje za uwagę!

OPI
OŚRODEK PRZETWARZANIA INFORMACJI
PAŃSTWOWY INSTYTUT BADAWCZY