# Normalizacja i klasyfikacja tekstu

Agnieszka Ławrynowicz

Wydział Informatyki Politechniki Poznańskiej

9 marca 2021

# Model języka: bag-of-words

# Model bag-of-words

Kocham ten film! Jest słodki,
ale z satyrycznym zacięciem.
Dialogi są wspaniałe a
przygodowe sceny zabawne
… Film jest zarazem
cudaczny
i romantyczny, gdy bawi się
konwencją bajki. Widziałam
go już kilka razy i jestem
szczęśliwa
gdy mogę obejrzeć go
ponownie z przyjaciółmi,
którzy go jeszcze nie widzieli.


3 go
2 z
2 jest
2 i
2 gdy
2 film
1 zarazem
1 zacięciem
1 zabawne
1 wspaniałe
1 widzieli
1 widziałam
1 ten
1 szczęśliwa
1 słodki
…

Zaadaptowane z "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

# Model bag-of-words

## Bag-of-words (BOW)

Tekst jest reprezentowany jakby to był worek wyrazów, tzn. nieuporządkowany zbiór wyrazów, z pominięciem gramatyki, a nawet szyku wyrazów, ignorując ich pozycję, natomast zachowując częstość wyrazów w tekście.

- najprostszy model wektorowy tekstu
- wartości składowych wektora są równe częstości występowania danego wyrazu w tekście
- słowa najczęstsze mają największą wagę, ale najmniejszą informatywność

Przykład:
Piotr jest szybszy niż Paweł i Paweł jest szybszy niż Piotr → te same wektory

# Schemat wag tf-idf

# Macierz term-dokument

- Każda komórka: liczba wystąpień termu $t$ w dokumencie $d$
  - Każdy dokument jest wektorem wystąpień w przestrzeni $\mathbb{N}_t$

Przykład:

| | Doc1 | Doc2 | Doc3 |
|---|---|---|---|
| samochód | 320 | 15 | 14 |
| auto | 5 | 1 | 2 |
| ubezpieczenie | 3 | 0 | 0 |
| wypadek | 18 | 50 | 49 |

# Macierz term-dokument

- Każda komórka: liczba wystąpień termu $t$ w dokumencie $d$
  - Każdy wyraz jest wektorem wystąpień w przestrzeni $\mathbb{N}_D$

Przykład:

| | Doc1 | Doc2 | Doc3 |
|---|---|---|---|
| samochód | 320 | 15 | 14 |
| auto | 5 | 1 | 2 |
| ubezpieczenie | 3 | 0 | 0 |
| wypadek | 18 | 50 | 49 |

# Macierz term-dokument

- Każda komórka: liczba wystąpień termu $t$ w dokumencie $d$
  - Dwa dokumenty możemy uznać jako podobne jeżeli ich wektory są podobne (analogicznie dla termów)

Przykład:

| | Doc1 | Doc2 | Doc3 |
|---|---|---|---|
| samochód | 320 | 15 | 14 |
| auto | 5 | 1 | 2 |
| ubezpieczenie | 3 | 0 | 0 |
| wypadek | 18 | 50 | 49 |

# Macierz term-dokument

- $t$ – term (wyraz)
- $d$ – dokument
- $N$ – liczba wszystkich dokumentów

# Schemat wag tf-idf

tf-idf: Term frequency - inversed document frequency

- $w_{t,d}$ – waga (składowa wektora)
- $tf_{t,d}$: *term frequency*, liczba wystąpień termu w tekście
- $df_t$: *document frequency*, liczba dokumentów, w których występuje term
- $idf_t$: *inverse document frequency*: $log(\frac{N}{df_t})$

$$w_{t,d} = tf_{t,d} \cdot idf_t$$

najpopularniejszy schemat w information retrieval

# Komentarz

- częstość $tf_{t,d}$ termu $t$ w dokumencie $d$ jest zdefiniowana jako liczba wystąpień tego termu w $d$
- załóżmy, że chcielibyśmy użyć $tf$ obliczając dopasowanie dwóch tekstów, ale jak?
- częstość liczona wprost nie jest tym co nas interesuje
  - dokument z 10 wystąpieniami termu jest bardziej adekwatny niż dokument z 1 wystąpieniem
  - ale nie 10 razy bardziej

# Normalizacja tekstu

# Normalizacja tekstu

Potrzeba normalizacji tekstu (przykład):

- dopasowanie U.S.A i USA w zastosowaniach wyszukiwania informacji (*information retrieval (IR)*)

# Normalizacja tekstu

W przetwarzaniu języka naturalnego często wykonuje się normalizację tekstu polegającą na:

- Filtrowaniu wyrazów funkcyjnych
- Tokenizacji słów
- Normalizacji formatów słów
- Segmentacji zdań

# Filtrowanie wyrazów funkcyjnych

## Stop lista

Lista wyrazów odrzucanych (nie uwzględnianych w analizie) przed lub po przetwarzaniu języka naturalnego.

- słowa o małym znaczeniu (np. spójniki: i, lub)
- słowa popularne

# Tokenizacja (omówiona na poprzednim wykładzie)

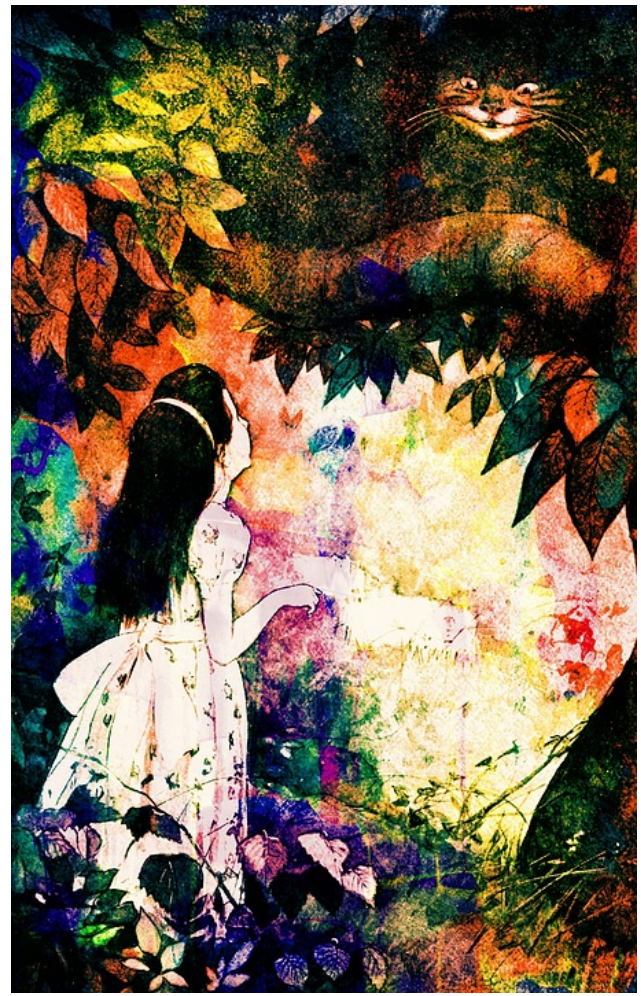
## Tokenizacja

Jeden z początkowych etapów w procesie przetwarzania języka naturalnego, polegający na podziale tekstu na tokeny (ciągi znaków oddzielone znakami zdefiniowanymi jako separatory).

Tokeny są w jakimś sensie elementarnymi "wyrazami".

# Ile wyrazów?

*Kot z Cheshire różni się od innych kotów!*



- Lemat: ten sam rdzeń, ta sama część mowy i zgrubsza sens wyrazu
  - Kot i kotów – taki sam lemat
- Forma powierzchniowa wyrazu: forma wyrazu, występująca w tekście w pełnej odmianie (*surface form*)
  - Kot i kotów – różne formy wyrazu

# Lematyzacja

## Lematyzacja

Zadanie, polegające na ustaleniu, że dane wyrazy mają ten sam lemat, mimo różnicy w formie i pogrupowaniu ich form tak aby były identyfikowane przez lemat lub element słownika.

# Lematyzacja c.d.

jestem, jesteś, są → być
kotów, koty → kot

# Morfologia

## Morfem

Najmniejsza jednostka gramatyczna, część wyrazu (może stanowić samodzielny wyraz).

Temat wyrazu: część wyrazu, która jest nośnikiem znaczenia wyrazu (np. kot-)
Prefiks, interfiks, sufiks: niesamodzielne morfemy (np. -ek w kotek)

# Stemming

## Stemming

Prostsza wersja lematyzacji, gdzie z wyrazu usuwana jest końcówka fleksyjna, pozostawiając tylko temat wyrazu.

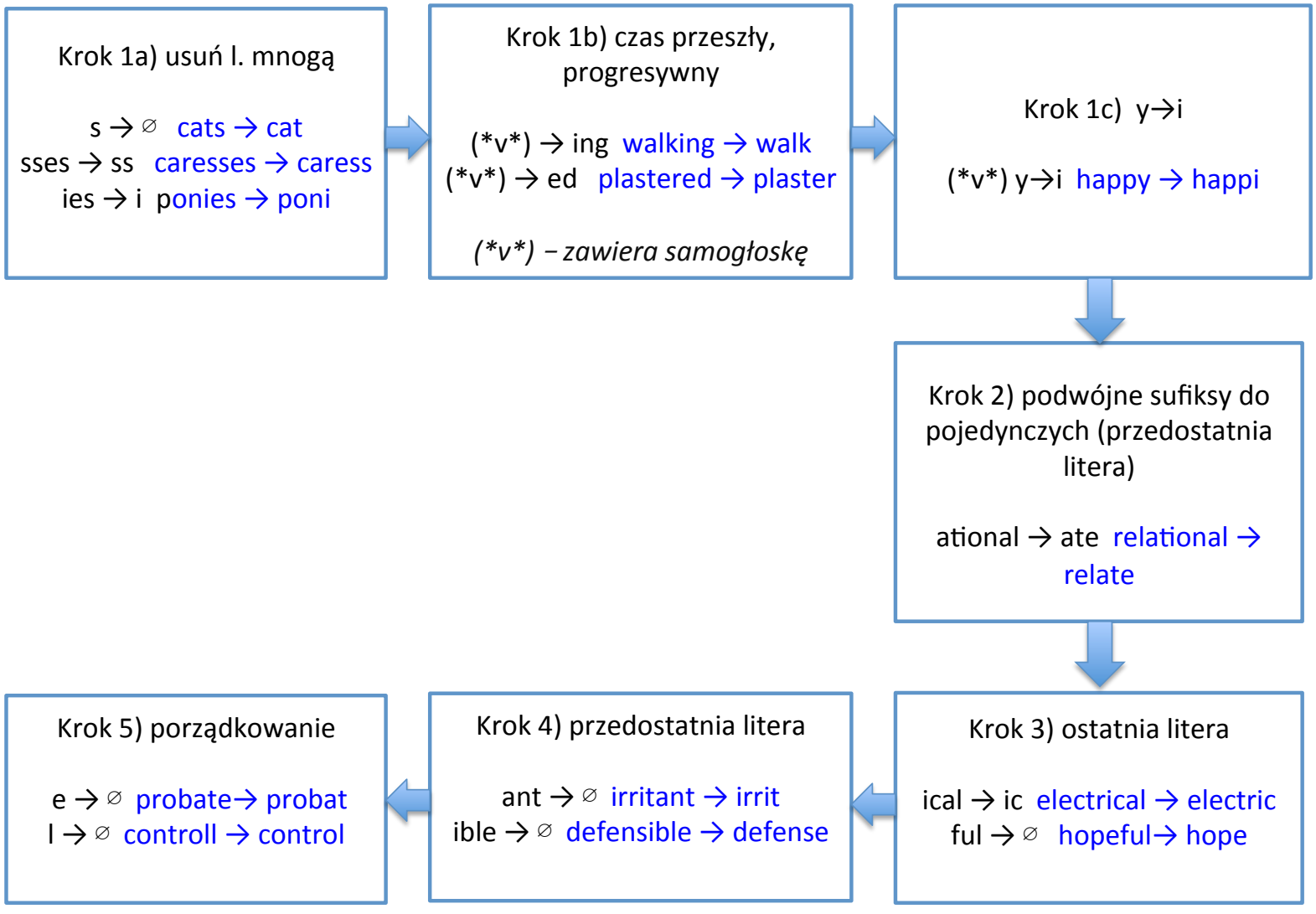
Przykład:
koty, kotów, kotek $\rightarrow$ kot

# Stemming: algorytm Portera

- Standardowy algorytm dla języka angielskiego (Porter, 1980)
- Oparty na serii reguł transformacji uruchamianych w seriach, kaskadowo, gdzie wynik każdego przebiegu jest podawany jako wejście do następnego przebiegu
- forma reguły: S1 $\rightarrow$ S2, gdzie S1 i S2 to sufiksy
- Złożone sufiksy są usuwane po kawałku w różnych krokach algorytmu. Przykład:

  GENERALIZATIONS
  GENERALIZATION (Krok 1)
  GENERALIZE (Krok 2)
  GENERAL (Krok 3)
  GENER (Krok 4)

# Stemming: algorytm Portera



# Segmentacja zdań

- !, ? (raczej jednoznaczne separatory)
- . (dosyć niejednoznaczny separator)
  - występuje np. w skrótach: dr inż.
  - można zbudować klasyfikator binarny decydujący o końcu zdania (tak/nie)
    - ręcznie definiowane reguły, wyrażenia regularne lub uczenie maszynowe

# Klasyfikacja tekstu

# Czy to spam?

**Mrs. Katie Zafer**

Przychodzące - LIBRA Wczoraj 15:40

KZ

[!! SPAM] ***SPAM*** Greetings to you my dear..

Odpowiedź-do: Mrs. Katie Zafer

I hope this message finds you in good health. I am a widow and have been fighting breast cancer for a long time. I am approaching 74 years old and may not have much time left. For this reason I am reaching out to you my dearest. My name is Mrs. Katie Zafer.

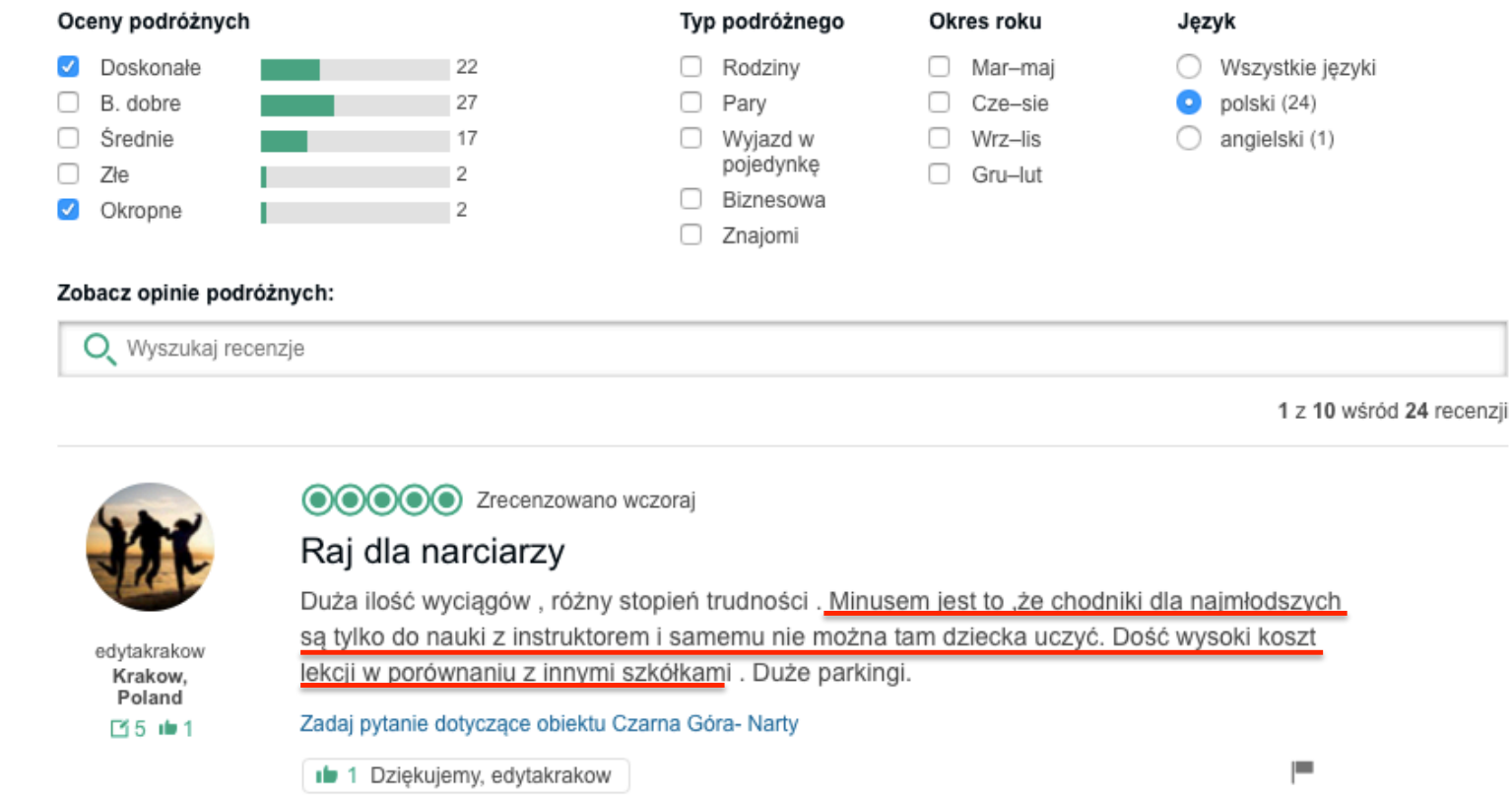
Recently, my doctor told me that i will not survive this sickness because my cancer level has reached a critical stage. I will take surgery operation any day next week. My dearest one, i contact you for a possible help. Please, pray for me always.

I decide to make a donation of my husband hard earn money sum of (US$ 5.500,000.00) Five Million Five Hunderd Thousand Dollars. I wish to know if i can entrust you with this fund to use it build a charity organization on my memorial. Please, i want you to use sum of (US$4.500,000.00) Four Million Five hundred thousand United State Dollar to provide educational opportunities for underprivileged children in rural areas and the area your hand can touch with that. You can support orphanages, widows and poor. It is my husband final wish to build a charitable homage on our monument before died took him away.

I will offer you (US$1,000,000.00) One Million only from the total money as your compensation for the performance of this charity work. If you are willing to carry out this humanitarian work, please kindly reply.

Beloved,
Mrs. Katie Zafer

# Czy to pozytywna czy negatywna recenzja?

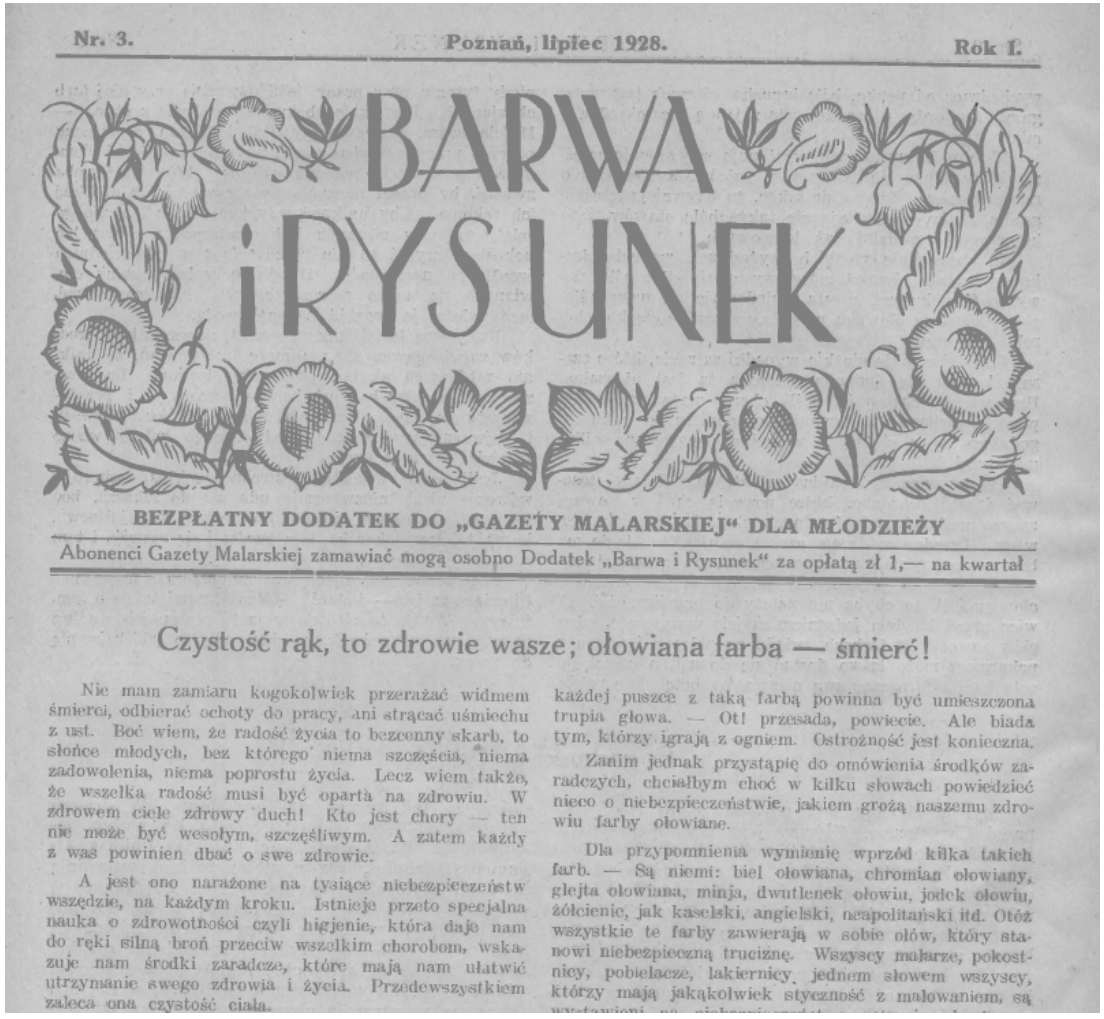



√

# W jakim języku napisano ten tekst?

**Warszawa** (an. *Warsaw*, miem. *Warschau*, fran. *Varsovie*, łac. *Varsovia*) je gardã w westrzédni Pòlsce, òd 1596 rokù ji stolecznym gardã a téż wôżnym ùczebnym, pòliticznym ë gòspòdarzczim môlã. W Warszawie mô swój plac Sejm ë prezydenta Pòlsczi. Warszawa je nôwikszym gardã Pòlsczi Repùbliczi.

# Jaki jest temat tego czasopisma?



Nr. 3. Poznań, lipiec 1928. Rok I.

BARWA i RYSUNEK

BEZPŁATNY DODATEK DO „GAZETY MALARSKIEJ“ DLA MŁODZIEŻY

Abonenci Gazety Malarskiej zamawiać mogą osobno Dodatek „Barwa i Rysunek“ za opłatą zł 1,— na kwartał

## Czystość rąk, to zdrowie wasze; ołowiana farba — śmierć!

Nie mam zamiaru kogokolwiek przerażać widmem śmierci, odbierać ochoty do pracy, ani strącać uśmiechu z ust. Boć wiem, że radość życia to bezcenny skarb, to słońce młodych, bez którego niema szczęścia, niema zadowolenia, niema poprostu życia. Lecz wiem także, że wszelka radość musi być oparta na zdrowiu. W zdrowem ciele zdrowy duch! Kto jest chory — ten nie może być wesołym, szczęśliwym. A zatem każdy z was powinien dbać o swe zdrowie.

A jest ono narażone na tysiące niebezpieczeństw wszędzie, na każdym kroku. Istnieje przeto specjalna nauka o zdrowotności czyli higjenie, która daje nam do ręki silną broń przeciw wszelkim chorobom, wskazuje nam środki zaradcze, które mają nam ułatwić utrzymanie swego zdrowia i życia. Przedewszystkiem zaleca ona czystość ciała.

każdej puszce z taką farbą powinna być umieszczona trupia głowa. — Ot! przesada, powiecie. Ale biada tym, którzy igrają z ogniem. Ostrożność jest konieczna.

Zanim jednak przystąpię do omówienia środków zaradczych, chciałbym choć w kilku słowach powiedzieć nieco o niebezpieczeństwie, jakiem grożą naszemu zdrowiu farby ołowiane.

Dla przypomnienia wymienię wprzód kilka takich farb. — Są niemi: biel ołowiana, chromian ołowiany, glejta ołowiana, minja, dwutlenek ołowiu, jodek ołowiu, żółcienic, jak kasselski, angielski, neapolitański itd. Otóż wszystkie te farby zawierają w sobie ołów, który stanowi niebezpieczną truciznę. Wszyscy malarze, pokostnicy, pobielacze, lakiernicy, jednem słowem wszyscy, którzy mają jakąkolwiek styczność z malowaniem, są

Polska, malarstwo, rzemiosło artystyczne, dekoracje ścienne

http://www.wbc.poznan.pl/dlibra/docmetadata?id=255660&from=publication

# Klasyfikacja tekstu

- Przypisywanie kategorii tematycznych, tematów lub gatunków
- Wykrywanie spamu
- Identyfikacja autora / wieku / płci
- Identyfikacja języka
- Analiza sentymentu

# Klasyfikacja tekstu - definicja

## Klasyfikacja tekstu

Wejście:

- dokument $d$
- ustalony zbiór klas $C = \{c_1, c_2, \ldots, c_j\}$

Wyjście: przewidywana klasa $c \in C$

# Metody klasyfikacji tekstu

- Reguły specyfikowane ręcznie oparte na kombinacjach wyrazów lub innych cechach
  - spam: adres-z-czarnej-listy LUB ("deal" I "free shipping")
  - trafność może być wysoka, gdy reguły są starannie dopracowane przez eksperta
  - definiowanie i utrzymywanie reguł jest kosztowne
- nadzorowane uczenie maszynowe

# Klasyfikacja tekstu - nadzorowane uczenie maszynowe

## Klasyfikacja tekstu - nadzorowane uczenie maszynowe

Wejście:

- dokument $d$
- ustalony zbiór klas $C = \{c_1, c_2, \ldots, c_j\}$
- zbiór trenujący z dokumentami etykietowanymi ręcznie $(d_1, c_1)$, ..., $(d_m, c_m)$

Wyjście: wytrenowany klasyfikator $y : d \to c$

# Klasyfikacja tekstu - nadzorowane uczenie maszynowe

Dowolny metoda klasyfikacji:

- naiwny klasyfikator bayesowski
- regresja logistyczna
- SVM
- (głębokie) sieci neuronowe
- ...

# Klasyfikacja tekstu - Naiwny klasyfikator bayesowski

- prosta metoda klasyfikacji, opierająca się na twierdzeniu Bayesa
- Korzysta z bardzo prostej reprezentacji dokumentu: bag-of-words

# Reguła Bayesa zastosowana do dokumentów i klas

Dla dokumentu $d$ i klasy $c$:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

# Reguła Bayesa zastosowana do dokumentów i klas

Dla dokumentu $d$ i klasy $c$:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

# Naiwny klasyfikator bayesowski

$$
\begin{aligned}
c_{MAP} &= \operatorname*{argmax}_{c \in C} P(c|d) && \text{MAP - maksymalne oszacowanie a posteriori, tzn. najbardziej prawdopodobna klasa} \\
&= \operatorname*{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} && \text{Reguła Bayesa} \\
&= \operatorname*{argmax}_{c \in C} P(d|c)P(c) && \text{opuszczenie mianownika}
\end{aligned}
$$

# Naiwny klasyfikator bayesowski c.d.

$$
\begin{aligned}
c_{MAP} = \quad & \underset{c \in C}{\operatorname{argmax}} P(c|d) \\
& = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, ..., x_n|c)P(c)
\end{aligned}
$$

dokument $d$ reprezentowany jako cechy $x_1, ..., x_n$

# Naiwny klasyfikator bayesowski c.d.

$$
\begin{aligned}
c_{MAP} = \quad & \underset{c \in C}{\operatorname{argmax}} P(c|d) \\
& = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, ..., x_n|c)P(c)
\end{aligned}
$$

dokument $d$ reprezentowany jako cechy $x_1, ..., x_n$

# Naiwny klasyfikator bayesowski c.d.

$$c_{MAP} \operatorname*{argmax}_{c \in C} P(x_1, x_2, ..., x_n|c)P(c)$$

- $O(|X|^n \cdot |C|)$ parametrów ($X$-zbiór cech)
- Jak często występuje taka klasa?
- Można oszacować tylko wtedy, gdy dostępna jest bardzo, bardzo duża liczba przykładów trenujących
- Możemy po prostu liczyć względne częstości występowania w korpusie?

# Naiwny klasyfikator bayesowski c.d.

Dwa założenia:

- założenie *bag-of-words*: kolejność nie ma znaczenia i cechy $x_1, x_2, ...$ jedynie kodują tożsamość wyrazów a nie ich kolejność
- założenie warunkowej niezależności: prawdopodobieństwa $P(x_i|c)$ są niezależne mając daną klasę $c$, a zatem można je "naiwnie" pomnożyć w następujący sposób:

$$P(x_1, x_2, ..., x_n|c) = P(x_1|c) \cdot P(x_2|c) \cdot ... \cdot P(x_n|c)$$

# Naiwny klasyfikator bayesowski c.d.

Końcowe równanie dla klasy wybranej przez naiwny klasyfikator Bayesa:

$$c_{NB} = \operatorname*{argmax}_{c \in C} P(c) \prod_{x \in X} P(x|c)$$

# Naiwny klasyfikator bayesowski c.d.

Aby zastosować naiwny klasyfikator Bayesa do tekstu, musimy rozważyć pozycje wyrazów, uwzględniając każdą pozycję wyrazu w dokumencie (za pomocą indeksu), tj. wszystkie pozycje wyrazów w dokumencie

$$c_{NB} = \operatorname*{argmax}_{c \in C} P(c) \prod_{i \in \text{pozycje}} P(w_i|c)$$

# Trenowanie naiwnego klasyfikatora bayesowskiego

Pierwsze podejście: użycie po prostu częstości występowania

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{doc}}$$

$$\hat{P}(w_j|c_j) = \frac{count(w_j, c_j)}{\sum_{w \in V} count(w, c_j)}$$

$V$ - słownik, który składa się z sumy wszystkich typów wyrazów we wszystkich klasach, nie tylko wyrazów w jednej klasie $c$

# Estymacja parametrów

$$\hat{P}(w_i|c_j) = \frac{count(w_j, c_j)}{\sum_{w \in V} count(w, c_j)}$$

- ułamek razy jaki pojawia się wyraz $w_i$ wśród wszystkich wyrazów w dokumentach w $c_j$
- utwórz zbiorczy dokument dla tematu $j$, łącząc wszystkie dokumenty w tym temacie i używając częstości w zbiorczym dokumencie

# Problem z estymacją

Co jeśli nie ma dokumentów w zbiorze trenującym ze słowem cudowny i sklasyfikowanych jako klasa pozytywna dla danego tematu?

$$\hat{P}(cudowny|poz) = \frac{count(cudowny, poz)}{\sum_{w \in V} count(w, poz)} = 0$$

Prawdopodobieństwo zerowe nie może być uwarunkowane, bez względu na inne dane

# Wygładzanie Laplace'a

Dodajemy 1

$$\hat{P}(w_i|c_j) = \frac{count(w_j, c_j) + 1}{\sum_{w \in V}(count(w, c_j) + 1)} = \frac{count(w_j, c_j) + 1}{(\sum_{w \in V} count(w, c_j)) + |V|}$$

# Trenowanie naiwnego klasyfikatora bayesowskiego c.d.

- wyekstrahuj *słownik* z korpusu trenującego
- Oblicz $P(c_j)$
  - dla każdego $c_j$ w $C$ wykonaj

    $\text{docs}_j \leftarrow$ wszystkie dokumenty z klasą $c_j$

    $P(c_j) \leftarrow \frac{|docs_j|}{\text{całkowita liczba dokumentów}}$
- Oblicz $P(w_k|c_j)$
  - *tekst*$_j \leftarrow$ pojedynczy dokument zawierający wszystkie dokumenty $\text{docs}_j$
  - dla każdego $w_k$ w *słowniku*

    $n_k \leftarrow$ liczba wystąpień $w_k$ w *tekst*$_j$

    $P(w_k|c_j) \leftarrow \frac{n_k+\alpha}{n+\alpha|\text{słownik}|}$

# Przykład

| | Dokument | Wyrazy | Klasa |
|---|---|---|---|
| Trenowanie | 1 | włoska Rzym włoska | w |
| | 2 | włoska włoska Florencja | w |
| | 3 | włoska Mediolan | w |
| | 4 | Poznań Polska włoska | p |
| Testowanie | 5 | włoska włoska włoska Poznań Polska | ? |

Zaadaptowane z "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

# Przykład

| | Dokument | Wyrazy | Klasa |
|---|---|---|---|
| Trenowanie | 1 | włoska Rzym włoska | w |
| | 2 | włoska włoska Florencja | w |
| | 3 | włoska Mediolan | w |
| | 4 | Poznań Polska włoska | p |
| Testowanie | 5 | włoska włoska włoska Poznań Polska | ? |

$$\hat{P}(c_j) = (\frac{N_{c_j}}{N_{doc}})$$

$$P(w) = \frac{3}{4}$$

$$P(p) = \frac{1}{4}$$

Zaadaptowane z "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

# Przykład

| | Dokument | Wyrazy | Klasa |
|---|---|---|---|
| Trenowanie | 1 | włoska Rzym włoska | w |
| | 2 | włoska włoska Florencja | w |
| | 3 | włoska Mediolan | w |
| | 4 | Poznań Polska włoska | p |
| Testowanie | 5 | włoska włoska włoska Poznań Polska | ? |

$$\hat{P}(w|c) = \frac{count(w,c)+1}{count(c)+|V|}$$

$$P(wloska|w) = \frac{5+1}{8+6} = \frac{3}{7}$$

$$P(Poznan|w) = \frac{0+1}{8+6} = \frac{1}{14}$$

$$P(Polska|w) = \frac{0+1}{8+6} = \frac{1}{14}$$

$$P(wloska|p) = \frac{1+1}{3+6} = \frac{2}{9}$$

$$P(Poznan|p) = \frac{1+1}{3+6} = \frac{2}{9}$$

$$P(Polska|p) = \frac{1+1}{3+6} = \frac{2}{9}$$

# Przykład

| | Dokument | Wyrazy | Klasa |
|---|---|---|---|
| Trenowanie | 1 | włoska Rzym włoska | w |
| | 2 | włoska włoska Florencja | w |
| | 3 | włoska Mediolan | w |
| | 4 | Poznań Polska włoska | p |
| Testowanie | 5 | włoska włoska włoska Poznań Polska | ? |

$$P(w|d5) = \frac{3}{4} \cdot (\frac{3}{7})^3 \cdot \frac{1}{14} \cdot \frac{1}{14} \approx 0.0003$$

$$P(p|d5) = \frac{1}{4} \cdot (\frac{2}{9})^3 \cdot \frac{2}{9} \cdot \frac{2}{9} \approx 0.0001$$

Zaadaptowane z "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

# Apache SpamAssasin: lista cech

- Wzmianki o milionach dolarów
- "Online Pharmacy"
- Viagra
- tytuł dużymi literami
- HTML ma niski stosunek tekstu do obszaru obrazu
- "One hundred percent guaranteed"
- groźby o usunięciu z listy
- ...

# Dziękuję za uwagę!