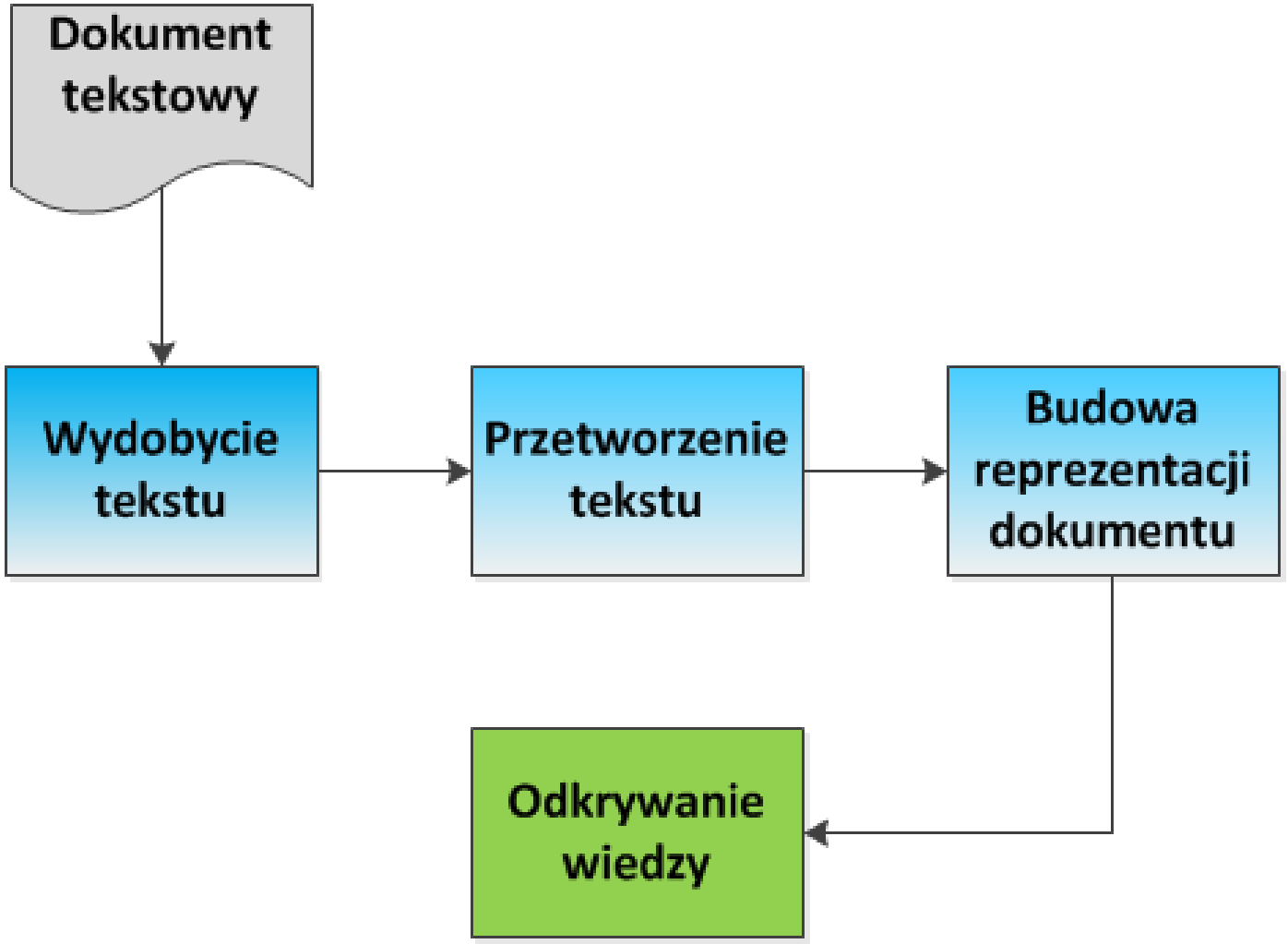
Text mining - odkrywanie wiedzy z tekstowych zbiorów danych

Wykład 2

Przetwarzanie tekstu

Reprezentacja dokumentów

Etapy analizy dokumentów tekstowych



Wyodrębnianie rdzenia (ang. stemming)

stemming – automatyczne odnajdywanie rdzeni wyrazów.

Większość stemerów nie zapewnia tego, że utworzone przez nie ciągi liter to rzeczywiście rdzenie – nie jest to jednak istotne, tak długo jak dla wszystkich wyrazów należących do danego leksemu otrzymujemy taki sam rdzeń.

Stemmery (angielskie)

Ogólny podział

- Stemery specjalizowane do zastosowań lingwistycznych (generowane rdzenie powinny rzeczywiście odpowiadać rdzeniom w rozumieniu lingwistyki, szybkość działania nie jest bardzo istotna)
- Stemery specjalizowane do zastosowań TM oraz IR (information retrieval)

- Pierwszy skuteczny algorytm dla angielskiego - **Lovin's stemmer** (1968) – stemmer jednoprzebiegowy, wykorzystujący tablicę 250 możliwych podstawień końcówek oraz dodatkowy etap postprocessingu – był projektowany jako uniwersalny
- **Obecnie najpopularniejszy stemmer – Porter's stemmer**, specjalizowany dla TM oraz IR, wieloprzebiegowy, nie generuje poprawnych językowo rdzeni
- SNOWBALL – język (+kompilator do ANSI C) do tworzenia stemmerów

  http://snowballstem.org/

Stemmery dla języka polskiego

- **Stempel - Algorithmic Stemmer for Polish Language (http://getopt.org/stempel/index.html)**
  - „a software package consisting of high-quality stemming tables for Polish, and a universal algorithmic stemmer, which operates using these tables”
  - The stemmer code is taken virtually unchanged from the Egothor project (http://egothor.sourceforge.net/).
- **http://www.cs.put.poznan.pl/dweiss/xml/projects/lametyzator/index.xml?lang=pl**
  - Oparty o słownik "ispell" oraz pakiek FSA Jana Daciuka
  - Przeniesiony do http://morfologik.blogspot.com/

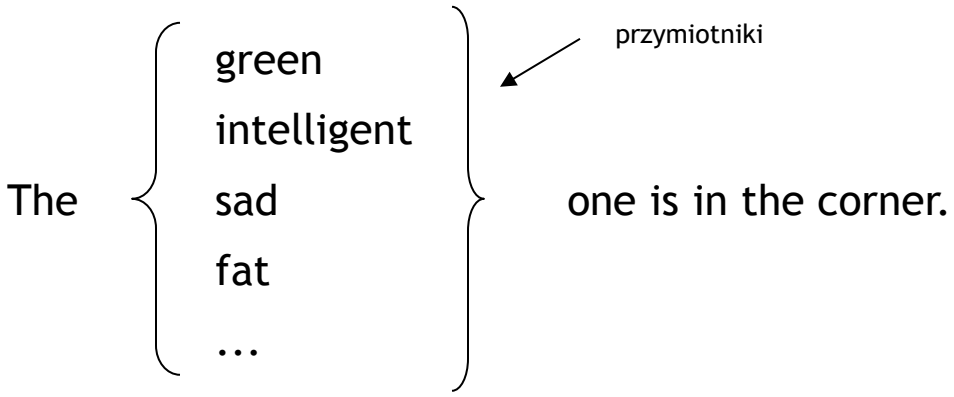
Lematyzacja

- Lematyzacja (ang. lemmatisation) – sprowadzenie słowa do formy podstawowej.

- Dostępna w różnych pakietach do NLP: https://opennlp.apache.org, https://nlp.stanford.edu/software/
- Język polski: http://sgjp.pl/morfeusz/ https://github.com/morfologik

Części mowy (1)

- **Nazwy** – części mowy (ang. *parts of speech – POS*), kategorie syntaktyczne itp.
- **Najważniejsze klasy**
  - **rzeczownik** – opis rzeczy (przedmiotów, pojęć itp.)
  - **czasownik** – opis działania, akcji
  - **przymiotnik** – opis cech rzeczowników
  - przysłówek, liczebnik, przyimek, spójnik, zaimek, partykuła, wykrzyknik
- **Test substytucji**



Części mowy (2)

Słowa mogą należeć do więcej niż jednej klasy, np.
sweet – słodki (przymiotnik), sweet – cukierek (rzeczownik)

Zamknięte i otwarte klasy POS

- otwarte – duża liczba słów, zmienna zawartość, np.
  - przymiotniki
  - rzeczowniki
  - czasowniki

- zamknięte – mała liczba słów, ściśle określona funkcja, np.
  - przyimki
  - zaimki
  - rodzajniki
  - spójniki

Zwykle oznaczane za pomocą znaczników (POS tags), szczególnie popularne znaczniki użyte przy tworzeniu *Brown corpus*

Części mowy - tagi

Przykłady oznaczeń wg. *Brown corpus* – oczywiście specyficzne dla języka angielskiego

- **rzeczowniki (NN)**
  - nazwy własne (NNP) – *United States*
  - adverbial nouns (NR) – *home, west, tomorrow*
  - liczba mnoga – NNS, NNPS, NRS - *flowers*
  - *possesive – NN$, NNS$, NNP$, itd. – Peter's*

- **przymiotniki (JJ)**
  - stopień wyższy (JJR) – *richer*
  - najwyższy (JJT + JJS) *(np. chief, main, top)*
  - liczby ! (CD) – *one, two, 60000*

- **czasowniki (VB)**
  - trzecia osoba lp. (VBZ) – *takes*
  - czas przeszły (VBD) – *took*
  - present participle (VBG) – *taking*
  - past participle (VBN) – *taken*
  - modal auxiliaries (MD) – *can, may, must, could* itd.
  - specjalne oznaczenia dla form *be*, *have* i *do (np. past participle have -> had HVN)*

POS tagi dla języka polskiego

Tabela 1. Klasy leksemów i ich rozbicie na fleksemy. W wypadku leksemów złożonych z tylko jednego fleksemu wspólna nazwa leksemu i fleksemu zajmuje dwie kolumny tabeli.

leksem	fleksem	ozn.
rzeczownik	rzeczownik	subst
	forma deprecjatywna	depr
przymiotnik	przymiotnik	adj
	przymiotnik przyprzymiotnikowy	adja
	przymiotnik poprzyimkowy	adjp
przysłówek odprzymiotnikowy i/lub stopniowalny		adv
liczebnik		num
zaimek nietrzecioosobowy		ppron12
zaimek trzecioosobowy		ppron3
zaimek SIEBIE		siebie
czasownik	forma nieprzeszła	fin
	forma przyszła czasownika BYĆ	bedzie
	aglutynant czasownika BYĆ	aglt
	pseudoimiesłów	praet
	rozkaźnik	impt
	bezosobnik	imps
	bezokolicznik	inf
	imiesłów przys. współczesny	pcon
	imiesłów przys. uprzedni	pant
	odsłownik	ger
	imiesłów przym. czynny	pact
	imiesłów przym. bierny	ppas
czasownik typu WINIEN (forma teraźniejsza)		winien
predykatyw		pred
przyimek		prep
spójnik		conj
kublik (partykuło-przysłówek)		qub
ciało obce nominalne		xxs
ciało obce luźne		xxx

Źródło: M.Wloliński System znaczników morfosyntaktycznych w korpusie IPI PAN

Składnia

- **Kolejność słów w zdaniach nie jest bez znaczenia – choć w niektórych językach (angielski) jest istotniejsza niż w innych (polski)**
- **Języki pozycyjne <-> języki fleksyjne**
  - **informacja która w językach fleksyjnych zawarta jest w odmianie słów, w językach pozycyjnych przekazywana jest w strukturze zdania i kontekście**

- **Podział wypowiedzi na zdania, zdań na części zdania (*constituents*):**
  - **I put *the bagels* in the freezer**
  - **I put in the fridge *the bagels***

she			him
the woman			the man
the tall woman	saw		the short man
the very tall woman			the very short man

Rozbiór zdania (1)

Nieco inny w języku polskim (podmiot, orzeczenie, dopełnienie, zdania proste i złożone – równorzędnie i podrzędnie) i angielskim

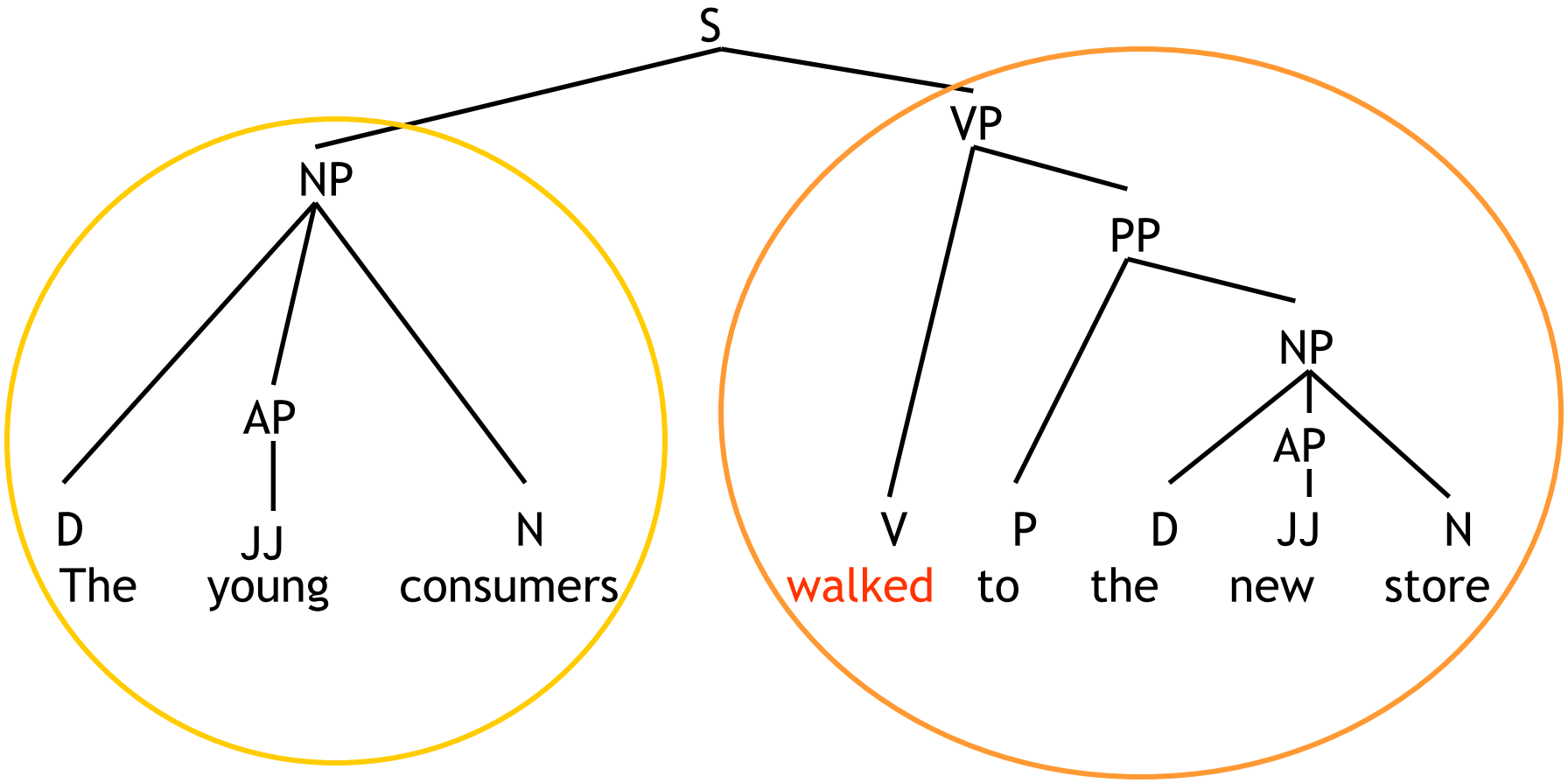
- **Noun phrases (NP)** np. *The homeless man in the park that I tried to help yesterday*
- **Verb phrases (VP)** np. He *was trying to keep his temper*
- **Prepositional phrases (PP)** np. *with a net*
- **Adjective phrases (AP)** np. she is *very sure of herself*

Rodzaje zdań

- oznajmujące
- pytające
- rozkazujące

Rozbiór zdania (2)

Zwykle zdanie w języku angielskim ma taką postać:



Rozbiór zdania (3)

Struktura zdania jest rekursywna, tego rodzaju drzewa mogą być generowane przez reguły podstawień (*rewrite rules*) np:

S -> NP VP
NP -> AT NNS | AT NN | NP PP
VP -> VP PP | VBD | VBD NP
P -> IN NP
AT -> the
NNS -> children | students | mountains
VBD -> slept | ate | saw
IN -> in | of
NN -> cake

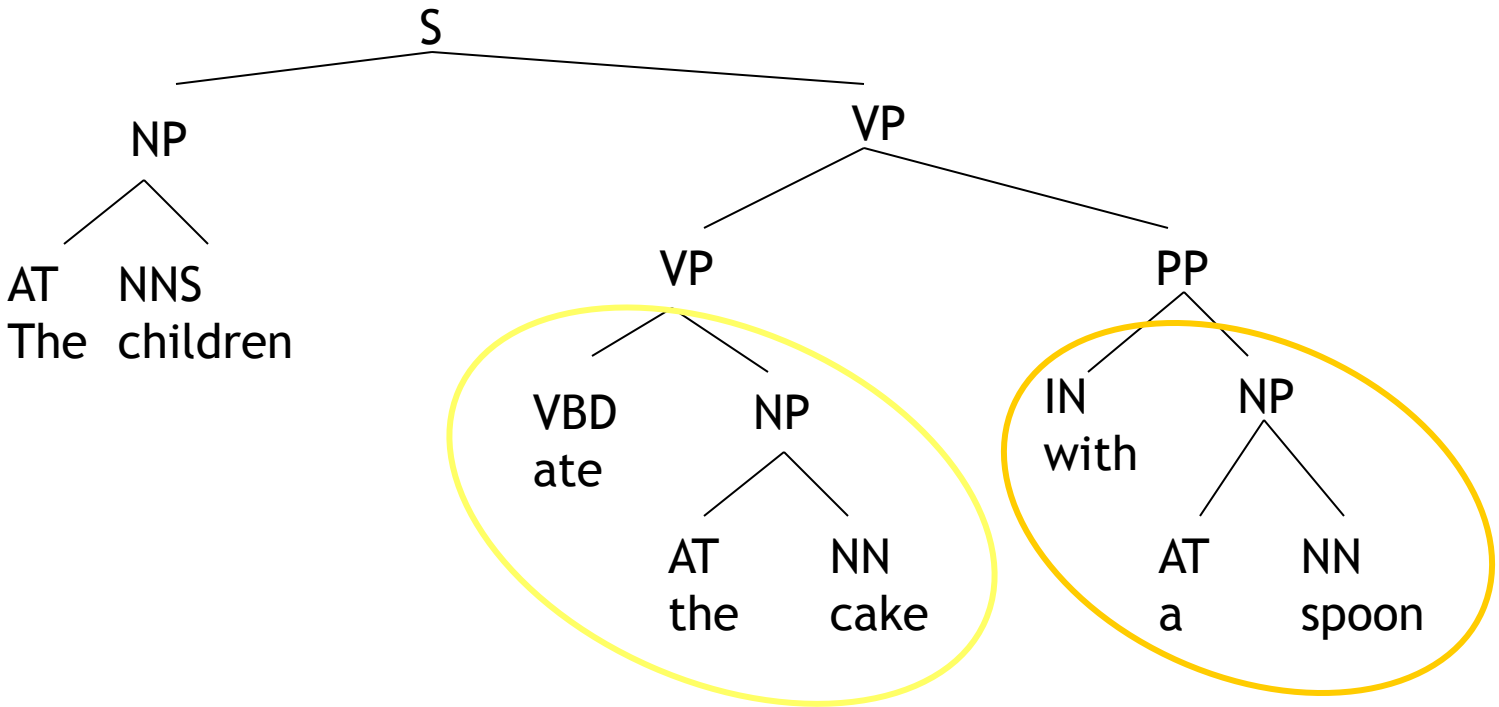
} lexicon

S -> NP VP -> AT NNS VBD -> The children slept

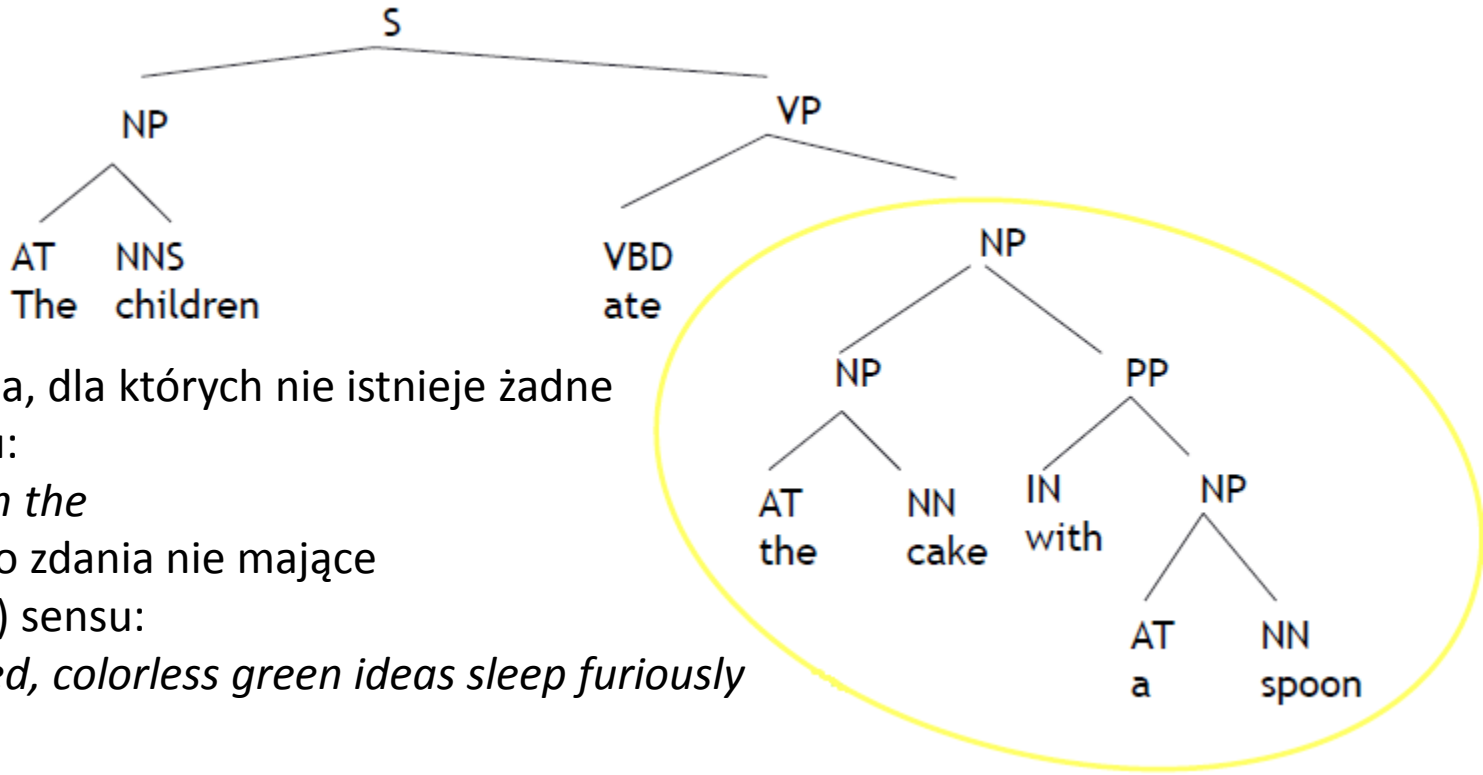
S -> NP VP -> AT NNS VBD NP -> AT NNS VBD AT NN -> The children ate the cake

– **Dokonując przekształceń korzystamy tylko z pojedynczych reguł, nie interesuje nas kontekst całego zdania – gramatyka bezkontekstowa (context free grammar, *CFG*)**

Rozbiór zdania(4)



Rozbiór zdania(5)



Istnieją też zdania, dla których nie istnieje żadne drzewo rozbioru:

- *Slept children the*

To nie to samo co zdania nie mające (semantycznego) sensu:

- *the cat barked, colorless green ideas sleep furiously*

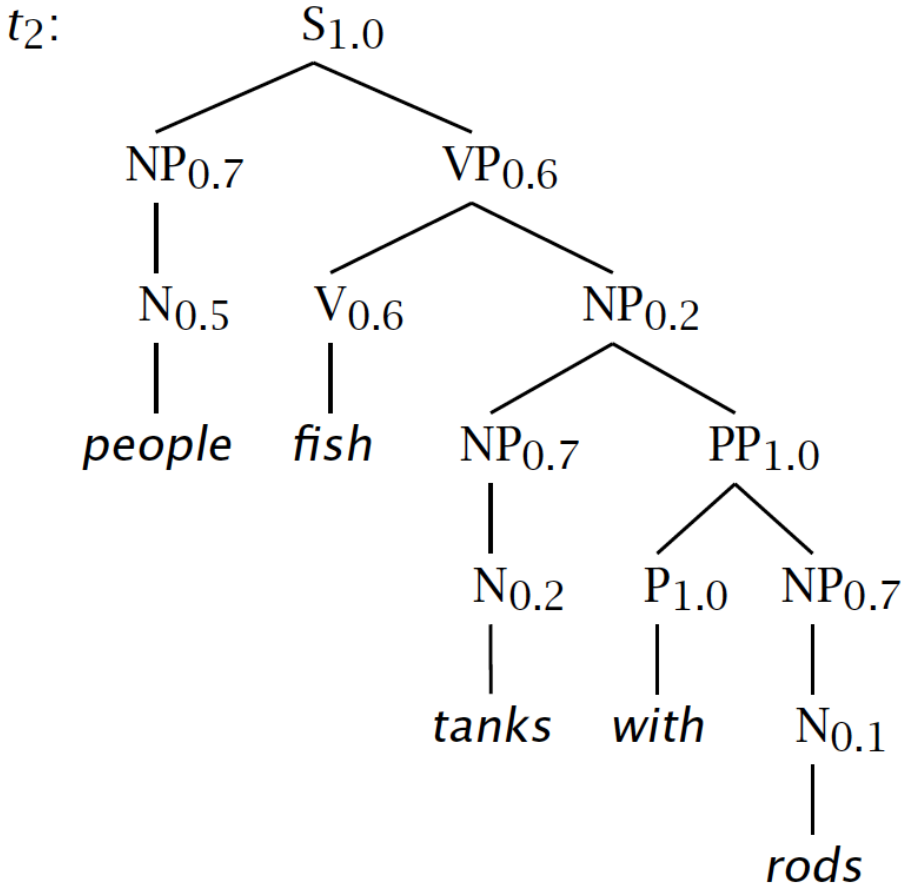
Główne problemy:

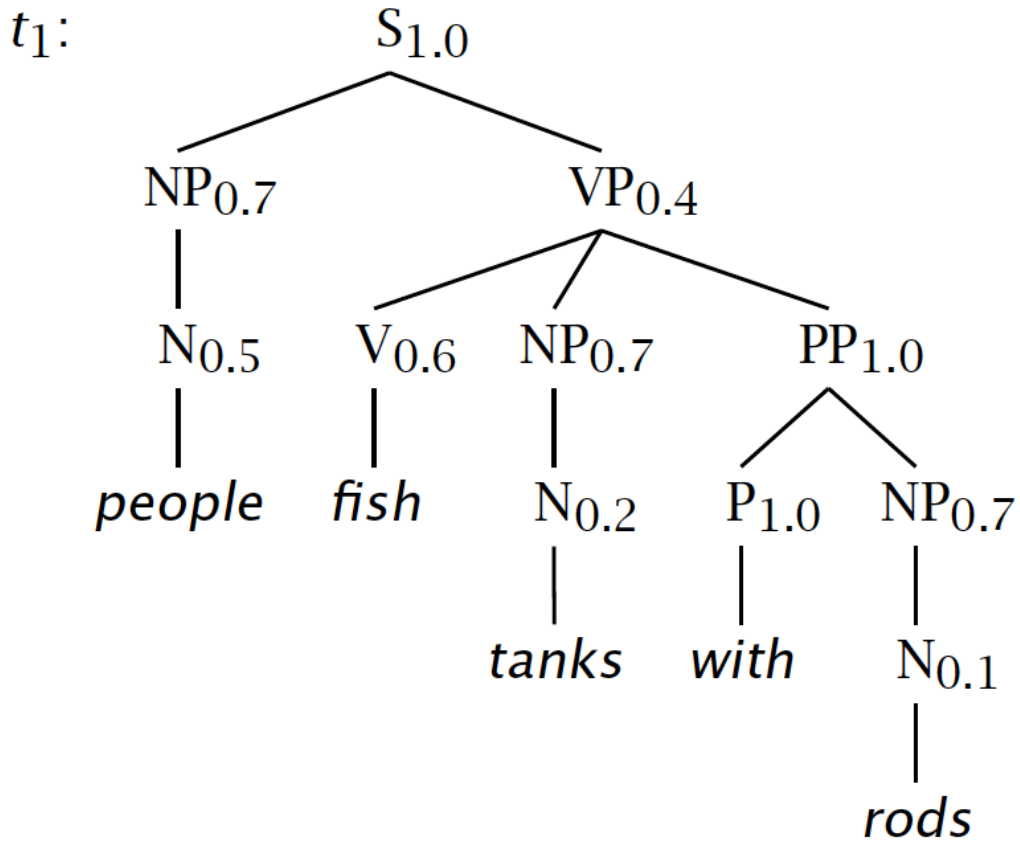
- generowanie drzew rozbioru nie jest zadaniem prostym – programowanie dynamiczne
- z wielu możliwych drzew rozbioru trzeba wybrać jedno właściwe, najbardziej prawdopodobne –probabilityczne gramatyki bezkontekstowe (*probabilistic context free grammars, PCFG*)

Prawdopodobieństwo drzewa rozbioru (1)

- P(*t*) – prawdopodobieństwo drzewa rozbioru jest iloczynem prawdopodobieństw generujących to drzewo reguł
- P(*s*) – prawdopodobieństwo zdania jest sumą prawdopodobieństw drzew, których liście tworzą dane zdanie

Prawdopodobieństwo drzewa rozbioru (2)





P(*t1*) = 1.0 × 0.7 × 0.4 × 0.5 × 0.6 × 0.7
× 1.0 × 0.2 × 1.0 × 0.7 × 0.1
= 0.0008232

P(*t2*) = 1.0 × 0.7 × 0.6 × 0.5 × 0.6 × 0.2
× 0.7 × 1.0 × 0.2 × 1.0 × 0.7 × 0.1
= 0.00024696

Narzędzia dla języka polskiego

- **Morfologik** - analizator morfologiczny i korektor gramatyczny
  – http://morfologik.blogspot.com/
- **Morfeusz** – analizator morfologiczny
  – http://sgjp.pl/morfeusz/
- **TaKIPI** – tager
  – http://plwordnet.pwr.wroc.pl/narzedzia-i-zasoby/narzedzia/takipi

Serwisy poświęcone maszynowemu przetwarzaniu języka polskiego

- **http://clip.ipipan.waw.pl/** - Computational Linguistics in Poland
- **http://clarin-pl.eu/en/home-page/** **-** Celem CLARIN jest udostępnianie zasobów językowych oraz elektronicznych narzędzi do automatycznego przetwarzania języka naturalnego badaczom we wszystkich dyscyplinach naukowych, a w szczególności z dziedziny nauk humanistycznych i społecznych.

Morfeusz – przykład działania

0	1	Mam	**mama** **mamić** **mieć**	subst:pl:gen:f impt:sg:sec:imperf fin:sg:pri:imperf
1	2	próbkę	**próbka**	subst:sg:acc:f
2	3	analizy	**analiza**	subst:sg:gen:f subst:pl:nom.acc.voc:f
3	4	morfologicznej	**morfologiczny**	adj:sg:gen.dat.loc:f:pos
4	5	.	**.**	interp

- Wiersz tabeli - jedna interpretacja morfologiczna
- Interpretacja ma podany lemat
  - Prawa kolumna — znaczniki opisujące wartości kategorii gramatycznych charakteryzujące poszczególne formy
  - Znaczniki pozycyjne - pierwsza pozycja określa klasę gramatyczną („część mowy”), następne pozycje reprezentują wartości kategorii gramatycznych przysługujących danej klasie, np. rzeczownik (subst) : liczba : przypadek : rodzaj

POS tagi dla języka polskiego

Tabela 1. Klasy leksemów i ich rozbicie na fleksemy. W wypadku leksemów złożonych z tylko jednego fleksemu wspólna nazwa leksemu i fleksemu zajmuje dwie kolumny tabeli.

leksem	fleksem	ozn.
rzeczownik	rzeczownik	subst
	forma deprecjatywna	depr
przymiotnik	przymiotnik	adj
	przymiotnik przyprzymiotnikowy	adja
	przymiotnik poprzyimkowy	adjp
przysłówek odprzymiotnikowy i/lub stopniowalny		adv
liczebnik		num
zaimek nietrzecioosobowy		ppron12
zaimek trzecioosobowy		ppron3
zaimek SIEBIE		siebie
czasownik	forma nieprzeszła	fin
	forma przyszła czasownika BYĆ	bedzie
	aglutynant czasownika BYĆ	aglt
	pseudoimiesłów	praet
	rozkaźnik	impt
	bezosobnik	imps
	bezokolicznik	inf
	imiesłów przys. współczesny	pcon
	imiesłów przys. uprzedni	pant
	odsłownik	ger
	imiesłów przym. czynny	pact
	imiesłów przym. bierny	ppas
czasownik typu WINIEN (forma teraźniejsza)		winien
predykatyw		pred
przyimek		prep
spójnik		conj
kublik (partykuło-przysłówek)		qub
ciało obce nominalne		xxs
ciało obce luźne		xxx

Źródło: M.Wloliński System znaczników morfosyntaktycznych w korpusie IPI PAN

Korpusy

- *„Korpus to dowolny zbiór tekstów, w którym czegoś szukamy. O korpusach w tym znaczeniu mówią najczęściej językoznawcy, ale także archiwiści, historycy i informatycy”* **– wydawnictwo PWN**

- *„Korpus - zbiór tekstów służący badaniom lingwistycznym, np. określaniu częstości występowania form wyrazowych, konstrukcji składniowych, kontekstów w jakich pojawiają się dane wyrazy. Korpusy językowe znalazły szerokie zastosowanie we współczesnej leksykografii. Są też wykorzystywane jako zbiory danych uczących i testowych w metodach uczenia maszynowego stosowanych w przetwarzaniu języków naturalnych.”* **– Wikipedia**

Korpus - cechy

Korpus – zbiór tekstów reprezentatywnych dla języka, zapisany w formie elektronicznej, o ile to możliwe zawierający metadane

- niezbilansowany – niereprezentatywny dla języka, np. zawierający jedynie teksty o pewnej tematyce, albo też
- zbilansowany - reprezentatywny dla całego języka
- jednojęzykowy vs. wielojęzyczny (bitext)
- anotowany – zawierający metadane, w szczególności POS tags i/lub informacje o rozbiorze zdania

Korpus jest zwykle statyczny i jako taki jest „fotografią” języka w pewnej chwili – np. Brown corpus – język angielski z lat 60-tych XX wieku

Korpusy w języku polskim

- Narodowy Korpus Języka Polskiego http://nkjp.pl/
- Korpus PWN http://korpus.pwn.pl/

Modelowanie języka

Model języka – model probabilistyczny pozwalający obliczyć prawdopodobieństwo zdania

- Jeśli $w_{1:n}$ oznacza ciąg wyrazów $w_1 w_2 \ldots w_n$.
- Jaka jest wartość $P(w_{1:n})$?

Możemy próbować określać prawdopodobieństwo wystąpień:

- poszczególnych liter
- poszczególnych wyrazów

Obliczenie prawdopodobieństwa wystąpienia słowa w zdaniu nie jest zadaniem prostym (ogólnie zależy od znaczenia wypowiadanego zdania), ale analiza poprzedzających słów może wiele pomóc:

- kolokacje
- części mowy i struktura zdania
- dziedzina semantyczna

Reguła łańcuchowa (1)

Jak obliczyć $P(w_{1:n})$?

Możemy wykorzystać regułę łańcuchową, wtedy:

$P(w_{1:n})$

$=P(w_{1:n-1})P(w_n|w_{1:n-1}) = P(w_{1:n-2})P(w_{n-1}|w_{1:n-2})P(w_n|w_{1:n-1})$ = itd. =

$= P(w_1)P(w_2|w_1)\ P(w_3|w_{1:2})\ P(w_4|w_{1:3})\ \ldots\ P(w_{n-1}|w_{1:n-2})P(w_n|w_{1:n-1})$

Sue swallowed the large green _____.
w_n

historia dla w_n

$w_{1:n-1}$: historia dla w_n

Problem – w naszym zbiorze danych (korpusie) będzie prawdopodobnie bardzo mało wystąpień $w_{1:n-1}$

Reguła łańcuchowa (2)

Uproszczenie: traktujemy proces generacji słów jako proces Markowa i przyjąć założenie Markowa (ang. *markov assumption*): tylko N najbliższych słów ma wpływ na to jakie będzie w_n :

$$P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-N+1:n-1})$$

Bigram: bierzemy pod uwagę tylko poprzednie słowo
Trigram: bierzemy pod uwagę dwa poprzedzające słowa
Tetragram: ... cztery itd.

Wtedy $$P(w_{1:n}) \approx \prod_{k=1,n} P(w_k | w_{k-N+1:k-1})$$

Tworzenie modelu (1)

Najprostszym podejściem do budowania modelu języka jest posłużenie się MLE (ang. maximum likelihood estimation**)** i policzenie wystąpień odpowiednich n-gramów w korpusie:

- korpus: <s> a b a b </s>
- MLE P(a|b)= ½, P(b|a)=1, P(a|<s>)= ½, P(</s>|b) = ½.

Przykład (Manning, Shuetze):

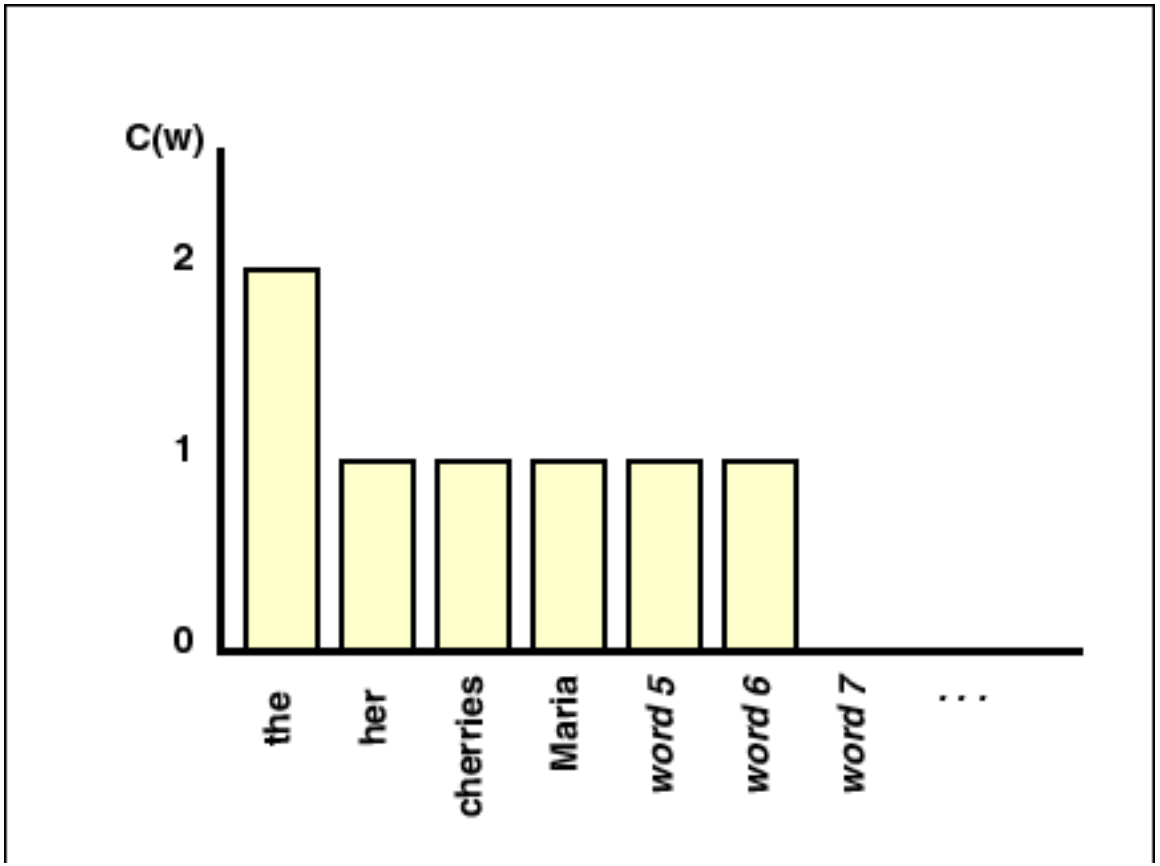
Korpus – powieści Jane Austen

N = 617,091 słów

V = 14,585 słów

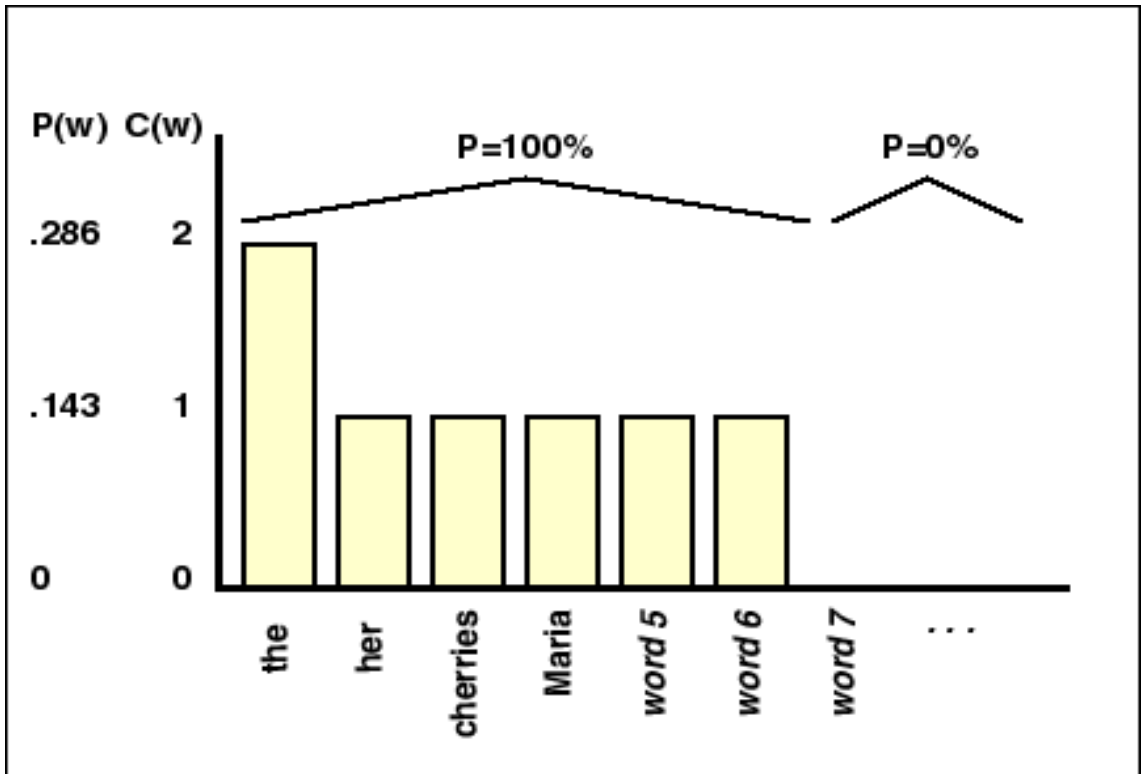
Tworzenie modelu (2)

Liczba wystąpień trigramu “inferior to _____” w korpusie:



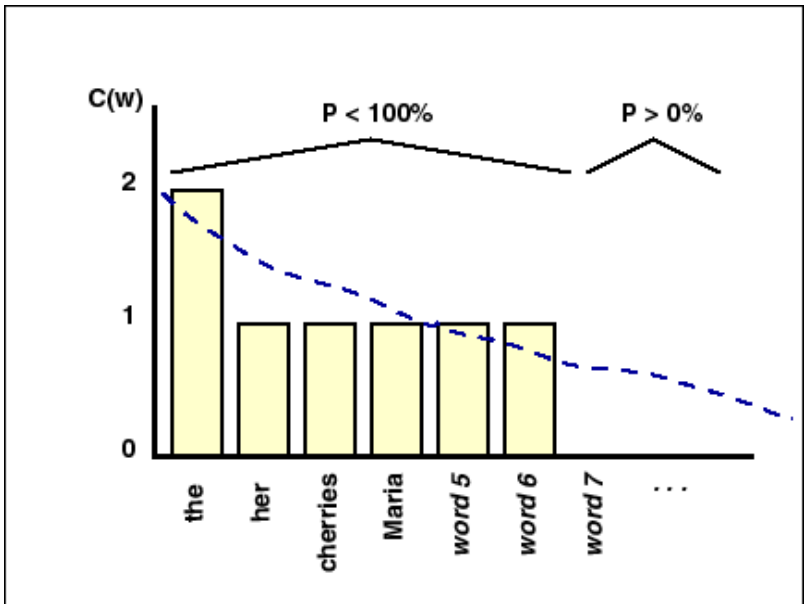
Tworzenie modelu (3)

- Zgodnie z MLE nie zaobserwowane wystąpienia trigramów otrzymują zerowe prawdopodobieństwa
- Typowo korpus jest jednak ograniczony i brak wystąpienia pewnego ciągu wyrazów może być przypadkowy



Wygładzanie (1)

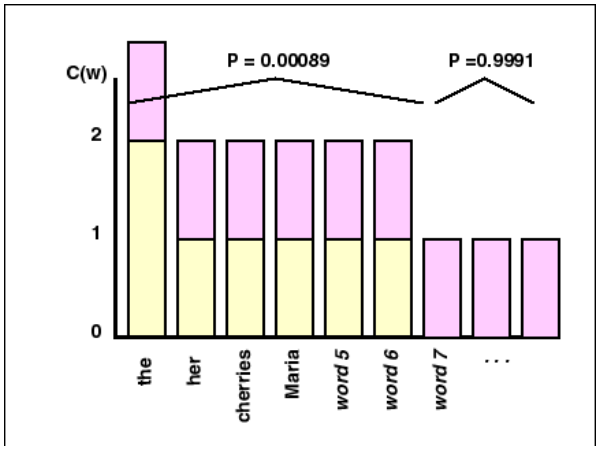
- Rzeczywisty rozkład prawdopodobieństwa wygląda zapewne tak:

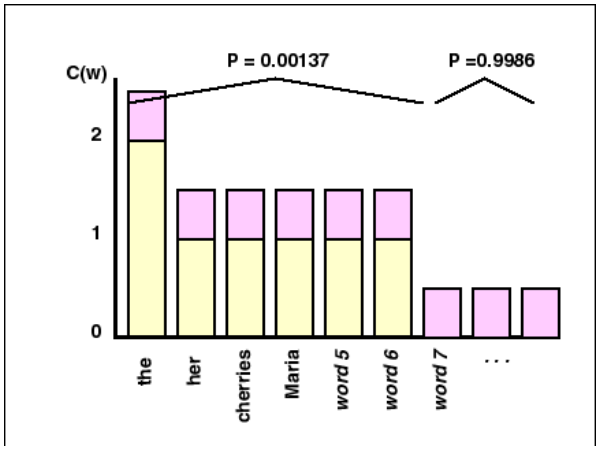


- Należy zatem :
  a) Zmniejszyć (ang. *discount*) nieco „masę prawdopodobieństwa” przypadającą na obserwowane przypadki
  b) Rozdzielić (ang. reallocate) uzyskany nadmiar na pozostałe przypadki

Wygładzanie (2)

- Wersja Laplace'a – uznajemy, iż każdy n-gram występuję przynajmniej 1 raz, lub wersja Jeffrey's-Parks – dopuszczamy wystąpienia „ułamkowe"





- Ogólnie:

$$P_{Lid}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + \lambda}{N + B\lambda}$$

gdzie

C = liczba wystąpień n-gramu w danych trenujących

N = liczba wystąpień wszystkich n-gramów w danych trenujących

B = liczba różnych n-gramów

MLE: $\lambda = 0$, LaPlace: $\lambda = 1$, Jeffreys-Perks: $\lambda = ½$

Modele oparte o sieci neuronowe (1)

- **Skip-gram model** - przewidywanie kontekstu danego słowa; wejście: słowo wyjście: kontekst słowa (zbiór słów)
- **Continuous Bag of Words (CBOW) model –** przewidywanie słowa na podstawie kontekstu; wejście: kontekst (sekwencja słów), wyjście: słowo.

Modele oparte o sieci neuronowe (2)

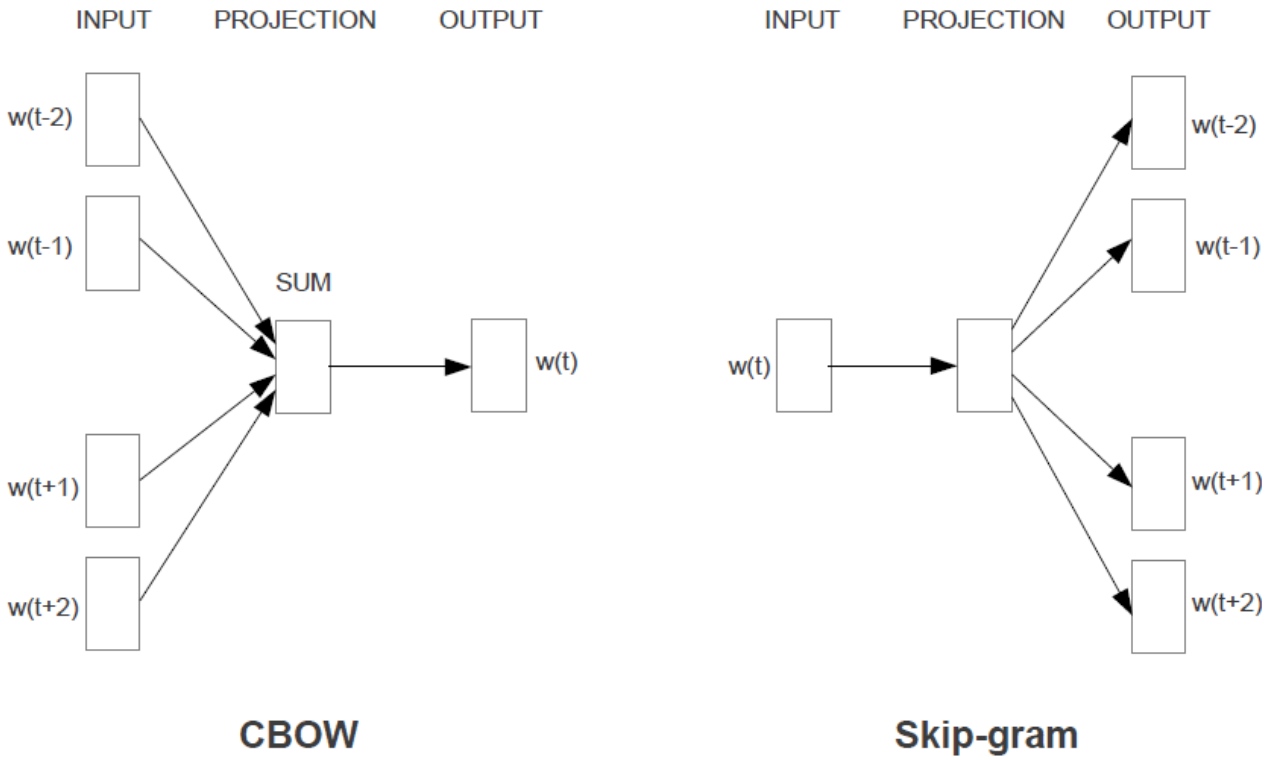


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Ocena modelu

- Czy model przypisuje wyższe prawdopodobieństwo dobrym zdaniom?
  - *Dobre zdanie = rzeczywiste (m.in. Sensowne, poprawne gramatycznie) lub często obserwowane zdanie*
- Zbiór treningowy / testowy, metryki ewaluacji
- Porównywanie modeli w zastosowaniu do określonych zadań (np. tłumaczenie, poprawa błędów)
- Porównanie efektywności wykonania zadań działających na testowanych modelach

Najlepszy model najlepiej przewiduje nieznany zbiór danych – daje najwyższe prawdopodobieństwo zdań, które występują w zbiorze testowym

Przykładowe zastosowania

OCR / rozpoznawanie mowy

wiele wypowiedzi brzmi podobnie np.

- I went to a party
- Eye went two a bar tea

Rudolph the red nose reindeer.
Rudolph the Red knows rain, dear.
Rudolph the Red Nose reigned here.

Poprawianie błędów ortograficznych

- … I think they’re okay …
- … I think there okay …
- … I think their okay …

Najbardziej prawdopodobne ze zdań-kandydatów

Tłumaczenie automatyczne

On voit Jon à la télévision

- Jon appeared in TV.
- In Jon appeared TV.
- Jon appeared on TV.

Analiza stylu pisania (wykrywanie plagiatów, autorstwa tekstów itp.)

Generowanie dużej ilości danych tekstowych ☺

Reprezentacje dokumentów tekstowych

Zliczanie słów

- reprezentacje unigramowe (bag-of-words)
  - binarne
  - częstościowe

Zliczanie sekwencji słów

- reprezentacja n-gramowe
- reprezentacje mieszane

reprezentacje pozycyjne

Reprezentacje unigramowe

Niech dany będzie dokument D=(w_1, w_2, ..., z_1, ..., w_n, z_m). Unigramową reprezentacją binarną dokumentu D nazywamy wektor **R** taki, że:

$$R_i = \begin{cases} 1\ gdy\ \exists j;\ w_j = v_i,\ v_i \in V \\ 0\ wpw. \end{cases}$$

Niech dany będzie dokument D = (w_1, w_2, ..., z_1, ..., w_n, z_m). Unigramową reprezentacją częstościową dokumentu D nazywamy wektor **R** taki, że:

$$R_i = \frac{\sum_{j=1}^{n} \begin{cases} 1\ gdy\ w_j = v_i, v_i \in V \\ 0\ wpw. \end{cases}}{n}$$

Reprezentacje bazujące na modelu Markowa

- n-gramowe
- mieszane

→ *„I would like to make phone…”*

Niech dany będzie dokument D=(w_1, w_2, ..., z_1, ..., w_n, z_m). Reprezentacją n-gramową dokumentu D nazywamy macierz M taką, że:

1. kolejne wiersze x macierzy odpowiadają kolejnym wariacjom r_x obejmującym n-1 słów ze słownika V
2. kolejne kolumny y macierzy odpowiadają kolejnym słowom v_y ze słownika V
3. elementy macierzy przyjmują wartości:

$$M_{x,y} = \sum_{j=1}^{o-n} \begin{cases} 1 \; gdy \; (w_j, w_{j+1}, \ldots, w_{j+n-1}) = r_x \wedge w_{j+n} = v_y \\ 0 \; wpw. \end{cases}$$

Reprezentacja n-gramowa – przykład

Przykład – bigram dla tekstu:

Twas brillig, and the slithy toves
Did gyre and gimble in the wabe

	twas	brillig	and	the	slithy	toves	did	gyre	gimble	in	wabe
twas	0	1	0	0	0	0	0	0	0	0	0
brillig	0	0	1	0	0	0	0	0	0	0	0
and	0	0	0	1	0	0	0	0	1	0	0
the	0	0	0	0	1	0	0	0	0	0	1
slithy	0	0	0	0	0	1	0	0	0	0	0
toves	0	0	0	0	0	0	1	0	0	0	0
did	0	0	0	0	0	0	0	1	0	0	0
gyre	0	0	1	0	0	0	0	0	0	0	0
gimble	0	0	0	0	0	0	0	0	0	1	0
in	0	0	0	1	0	0	0	0	0	0	0
wabe	0	0	0	0	0	0	0	0	0	0	0

N-gramy

Dla większych wartości n to podejście staje się niepraktyczne

Załóżmy, iż słownik zawiera 20000 słów

wtedy:

n	**Liczba klas**
2 (bigrams)	**400,000,000**
3 (trigrams)	**8,000,000,000,000**
4 (tetragrams)	**1.6×10^{17}**

Model wektorowy - podsumowanie

- **Model przestrzeni wektorowej (ang**. vector space model) dokument d_j jest reprezentowany przez wektory o długości n, gdzie n jest liczbą wyrazów występujących w rozważanym repozytorium:

$$\overrightarrow{d_j} = (d_{j1}, d_{j2}, d_{j3}, \dots, d_{jn})$$

$d_{ji} = 0\ dla\ k_{ji} = 0$

$d_{ji} = k_{ji}\ dla\ k_{ji} > 0\ (\ modelu\ binarny\ d_{ji} = 1)$

gdzie k_{ji} jest liczbą wystąpień wyrazu i w dokumencie j.

- **Modele n-gramowe**: dokument jest reprezentowany przez wektor o długości n, gdzie n jest liczbą n-gramów w rozważanym repozytorium.

Reprezentacja pozycyjna

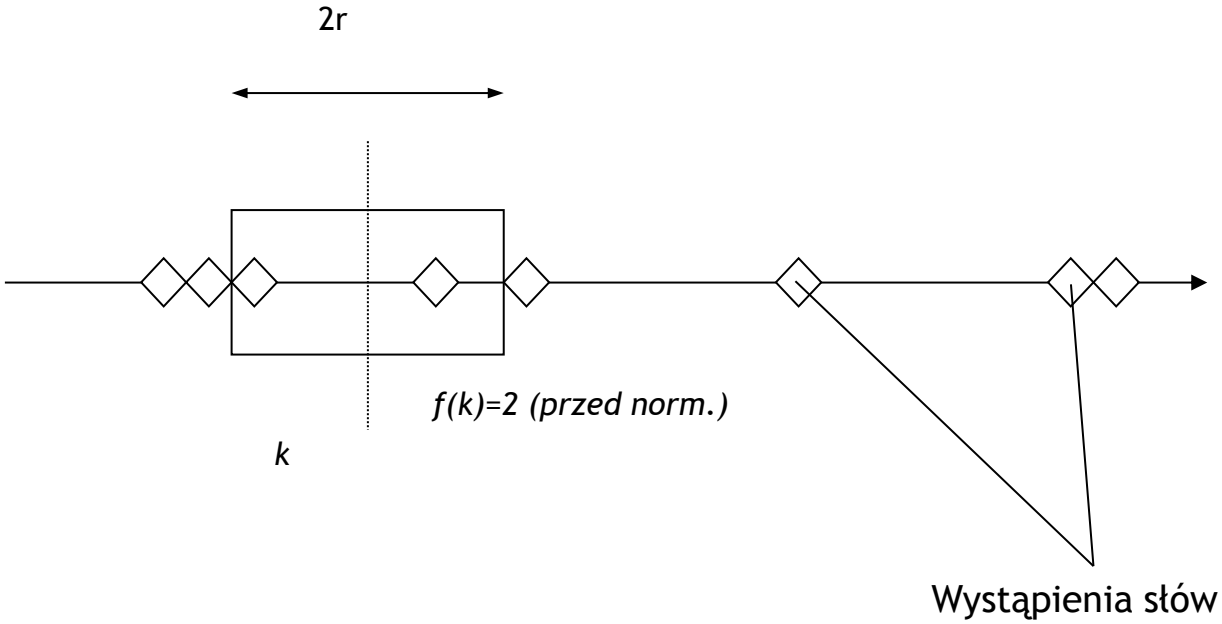
Niech dany będzie dokument D=(w_1, w_2, ..., z_1, ..., w_n, z_m). Reprezentacją pozycyjną dokumentu D nazywamy dwójkę (F, S) gdzie F jest zbiorem funkcji gęstości rozkładu słów f_{Vi} o następujących własnościach:
1) dziedziną funkcji f_{Vi} jest zbiór {1...n}
2) wartości funkcji f_{Vi} określone są następująco:

$$f_{v_i}(k) = \frac{\sum_{j=k-r}^{k+r} \begin{cases} 1 \; gdy \; w_j = v_i, v_i \in V \\ 0 \; wpw. \end{cases}}{\alpha_i} \qquad \sum_{1}^{n} f_{v_i} = 1$$

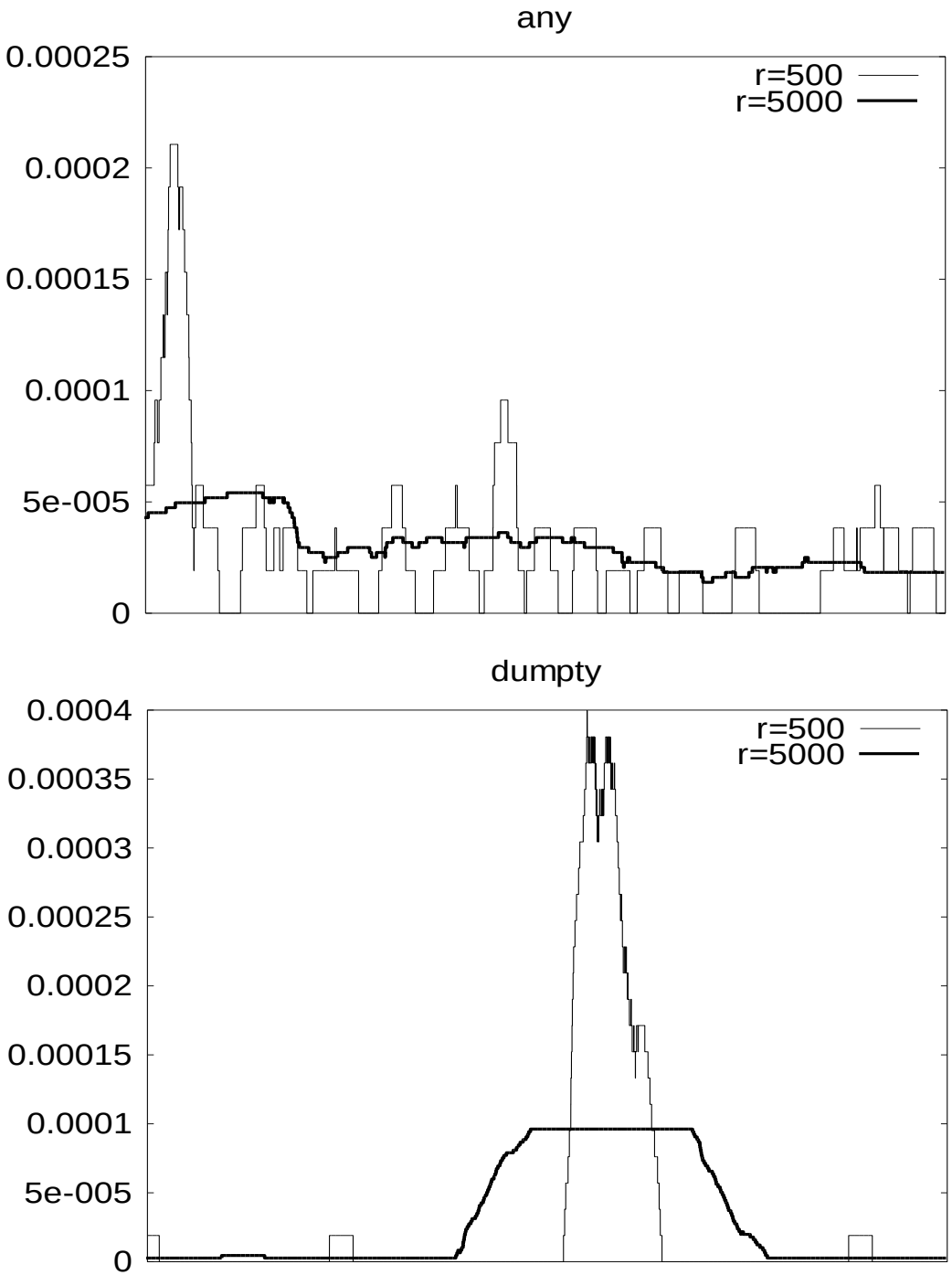
S- wektor skalujący o takich samych wartościach jak dla reprezentacji unigramowej.
Parametr r może być interpretowany jako reprezentacja rozmycia (ang. fuziness). Jeśli r =n to reprezentacja równoważna reprezentacji unigramowej . Jeśli r = 0 – reprezentacja pozwala na dokładne odtworzenie dokumentu.

Reprezentacja pozycyjna

Schemat objaśniający



Przykłady funkcji gęstości



Reprezentacja dokumentów o bogatej strukturze

Atrybuty nie muszą być wyłącznie częstościami słów/sekwencji słów

- topologia
- metadane (np. autorzy, data powstania)
- podobieństwo tekstów (klasyczny model dokumentów)
- częstość odwiedzin
- słowa kluczowe

<table>
<tr><td>Tekst</td><td rowspan="2">Elementy medialne (obraz, dźwięk itp.)</td><td rowspan="2">Osadzone aplikacje</td></tr>
<tr><td>Kroje pisma</td></tr>
<tr><td colspan="3">Hiperpołączenia z innymi dokumentami</td></tr>
<tr><td colspan="3">Układ stron i paginacja</td></tr>
</table>

Klasyfikacja oparta o formatowanie dokumentów

PAPERS

LISTS

PRESS

Klasa	Precyzja	Zupełność
PAPERS	0.9	0.9
LISTS	0.642857	0.9
PRESS	1	0.6

Reprezentacja oparta na kategoriach

$$R = \begin{bmatrix} |\{w \in D \wedge w \in P\}| \\ |\{w \in D \wedge w \in N\}| \\ |\{w \in D \wedge w \notin P \wedge w \notin N\}| \end{bmatrix}$$

gdzie: R — wektor reprezentacji,
D — tekst,
w — słowo w tekście,
P — słownik zawierający słowa pozytywne,
N — słownik zwierający słowa negatywne.

Przykład

„Benchmark U.S. crude-oil futures gain as the European Union slaps Iran with an oil-import embargo, heightening concerns over potential supply disruptions"

$$S = \{crude[n], gain[p], slap[n], embargo[n], heighten[p], concern[n], disruption[n]\}$$

Miary podobieństwa dokumentów (1)

- d_i, d_j - reprezentacja dokumentów w przestrzeni V,
- $w(d_l,t_p)$ - waga termu t_p w dokumencie d_l.

- Manhattan $$mn(d_i,d_j) = \sum_{k\in 1..|V|} | w(d_i,t_k) - w(d_j,t_k) |$$
- Euklidesowa $$eu(d_i,d_j) = \sqrt{\sum_{k\in 1..|V|} \left[w(d_i,t_k) - w(d_j,t_k) \right]^2}$$
- Kosinusowa $$\cos(d_i,d_j) = \frac{< d_i, d_j >}{| d_i \| d_j |} = \frac{\sum_{k=1}^{|V|} w(d_i,t_k) * w(d_j,t_k)}{\sqrt{\sum_{k=1}^{|V|} w(d_i,t_k)^2} \sqrt{\sum_{k=1}^{|V|} w(d_j,t_k)^2}}$$

Wskaźniki podobieństwa dokumentów

- d_i, d_j - reprezentacja dokumentów w przestrzeni V,
- $w(d_l,t_p)$ - waga termu t_p w dokumencie d_l.

- Dice's

$$dice(d_i,d_j) == \frac{2\sum_{k=1}^{|V|} w(d_i,t_k) * w(d_j,t_k)}{\sum_{k=1}^{|V|} w(d_i,t_k) + \sum_{k=1}^{|V|} w(d_j,t_k)}$$

- Jaccard's

$$jacc(d_i,d_j) == \frac{\sum_{k=1}^{|V|} w(d_i,t_k) * w(d_j,t_k)}{\sum_{k=1}^{|V|} w(d_i,t_k) + \sum_{k=1}^{|V|} w(d_j,t_k) - \sum_{k=1}^{|V|} w(d_i,t_k) w(d_j,t_k)}$$

Przetwarzanie reprezentacji

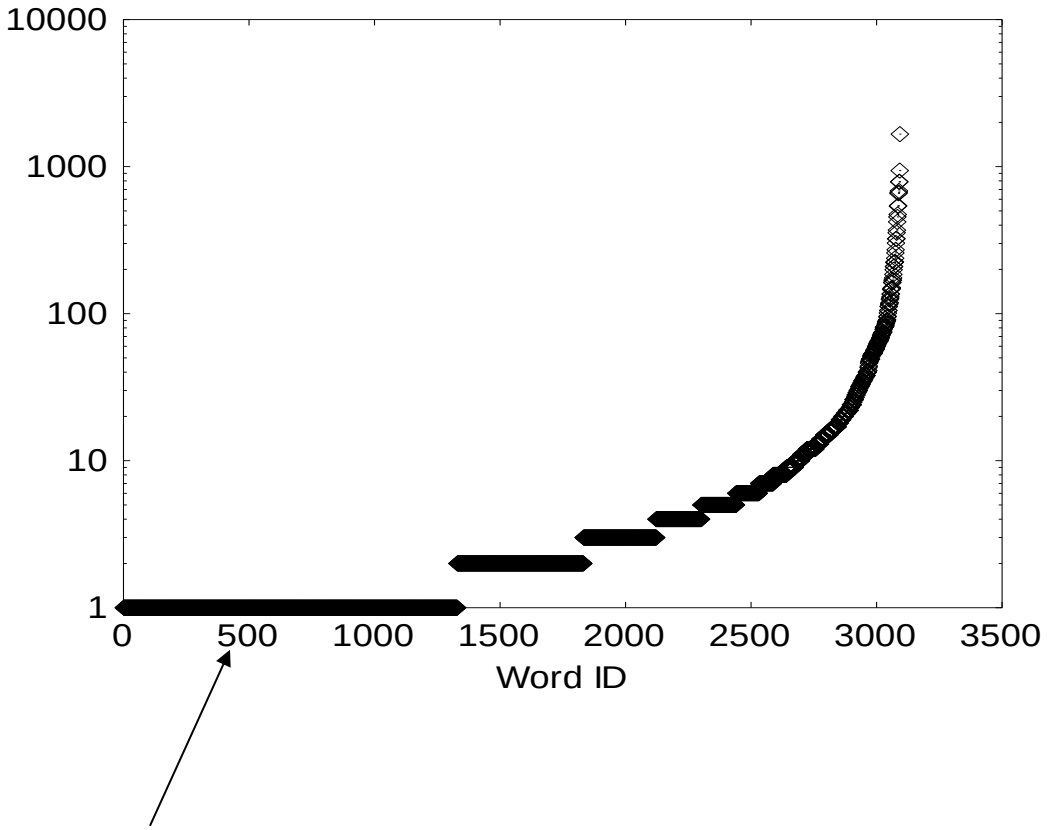
- ***Powiększanie rozmiaru reprezentacji***
  - Różne metody wygładzania

- ***Ograniczanie rozmiaru reprezentacji***
  - Funkcje istotności atrybutów
  - Wybór atrybutów
  - Przekształcanie przestrzeni atrybutów

Po co ograniczać rozmiar reprezentacji?

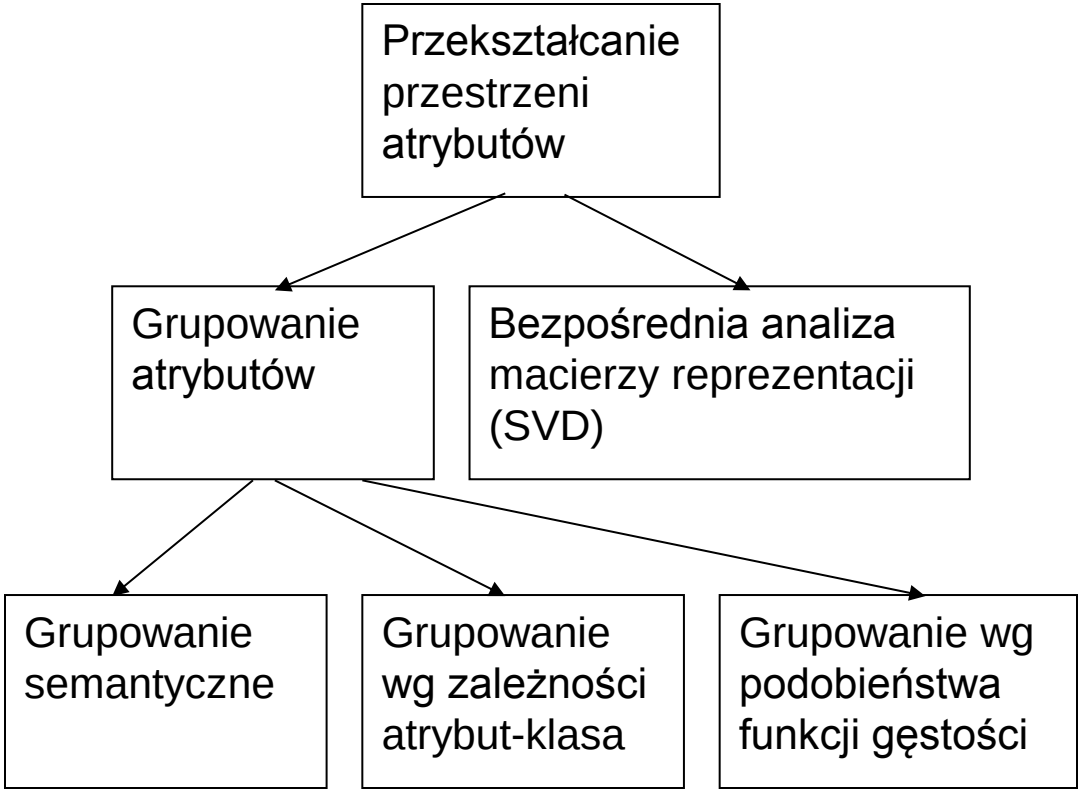
Prawo Zipfa

Słowo	Częstość
the	1664
and	940
to	789
a	788
it	683
you	666
I	658
she	543
of	538
said	473



„Hapax legomena”

Przekształcanie przestrzeni atrybutów



Ograniczanie wielkości reprezentacji

Funkcje istotności atrybutów – rodzina TF/IDF

term frequency $tf_{i,j}$ – określa częstość wystąpień atrybutu w_i w dokumencie d_j
document frequency df_i – określa liczbę dokumentów. w których występuje atrybut w_i
N – określa liczbę wszystkich dokumentów w systemie

$$\gamma_{lln}(w_i, d_j) = \left(1 + \log(tf_{ij})\right) \cdot \log\left(\frac{N}{df_i}\right)$$

$$\gamma_{lln}(w_i, d_j) = \left(1 + \log(tf_{ij})\right) \cdot \log(N) = \log(N) + \log(tf_{ij})$$ ← Atrybut w jednym dokumencie

$$\gamma_{lln}(w_i, d_j) = \left(1 + \log(tf_{ij})\right) \cdot \log\left(\frac{N}{N}\right) = \left(1 + \log(tf_{ij})\right) \cdot 0 = 0$$

Atrybut we wszystkich dokumentach

Funkcje istotności atrybutów - analiza funkcji gęstości

Np. wartość takiej funkcji równa 0 oznacza całkowicie równomierny rozkład wystąpień słowa, zaś dla maksymalnej koncentracji (tj. dla pojedynczego wystąpienia słowa w dokumencie) wartość równa jest 1.

Waga TF-IDF

$$TF - IDF\ (t, d, D) = tf(t, d) * idf(t, D)$$

gdzie:

$$\mathrm{tf(t, d)} = \frac{f(t, d)}{\max\{f(w, d) : w\ \epsilon\ d\}}$$

- t – wyrażenie, dla którego obliczany jest parametr
- d – dokument dla którego obliczany jest parametr
- $f(t, d)$ – częstotliwość wystąpienia słowa w dokumencie
- $f(w, d) : w\ \epsilon\ d$ – zbiór częstotliwości słów występujących w dokumencie

$$idf(t, D) = log\frac{|D|}{|\{d\ \epsilon\ D : t\ \epsilon\ d\ \}|},$$

- $|D|$ - liczba wszystkich dokumentów w zbiorze
- $|\{d\ \epsilon\ D\ :\ t\ \epsilon\ d\ \}|$ - liczba dokumentów w których występuje wyrażenie t

Funkcje istotności atrybutów – Information Gain

Information Gain określa, które atrybuty są tymi, które w najlepszy sposób różnicują klasy ze zbioru trenującego

$$IG(w_i) = -\sum_{j=1}^{l} P(k_j) \cdot \log P(k_j) + P(w_i) \cdot \sum_{j=1}^{l} P(k_j \mid w_i) \cdot \log P(k_j \mid w_i) +$$

$$+ P(\overline{w_i}) \cdot \sum_{j=1}^{l} P(k_j \mid \overline{w_i}) \cdot \log P(k_j \mid \overline{w_i})$$

$P(w_i)$ - prawdopodobieństwo wystąpienia atrybutu w_i w losowo wybranym dokumencie z systemu;

$P(k_j)$ - prawdopodobieństwo, iż losowo wybrany dokument należy do klasy k_j;

$P(k_j \mid w_i)$ - prawdopodobieństwo, iż dokument wybrany z dokumentów zawierających atrybut w_i należy do klasy k_j;

$P(\overline{w_i})$ - prawdopodobieństwo nie wystąpienia atrybutu w_i w losowo wybranym dokumencie z systemu;

$P(k_j \mid \overline{w_i})$ - prawdopodobieństwo, iż dokument wybrany z dokumentów nie zawierających atrybutu w_i należy do klasy k_j.

Rozkład SVD macierzy (1)

Każdą macierz rzeczywistą *A* można przedstawić w postaci **rozkładu SVD**:

$$A = U\Sigma V^T$$

gdzie:

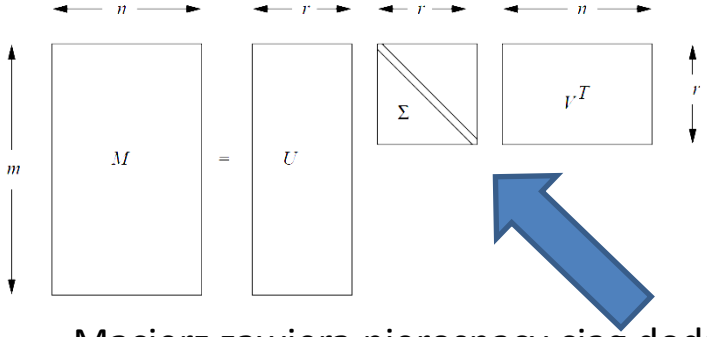
- *U* i *V* - macierze ortonormalne
- *Σ* - macierz diagonalna składająca się z nieujemnych wartości szczególnych macierzy *A*, zwyczajowo uporządkowane nierosnąco

- ***U* – kolumny U - wektory własne AA^T - lewe wektory szczególne macierzy A**
- ***Σ* – pierwiastki z niezerowych wartości własnych A^TA oraz AA^T**
- ***V* – kolumny V - wektory własne A^TA - prawe wektory szczególne macierzy A**

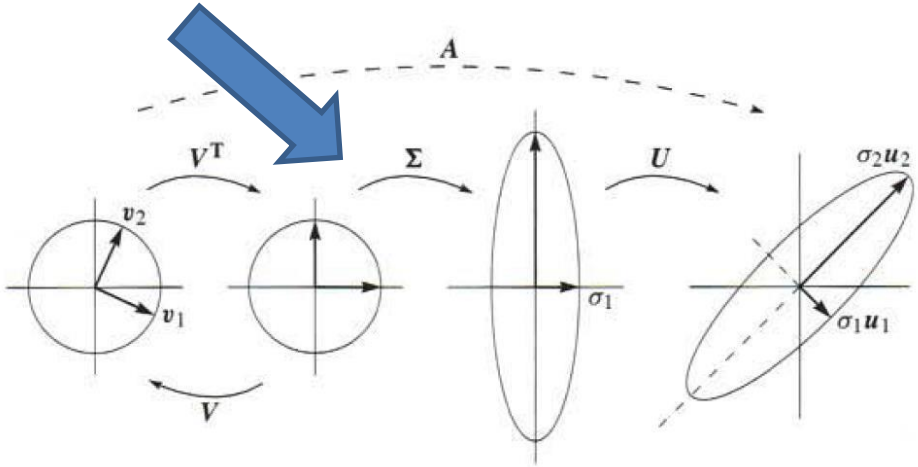
$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

gdzie: u_i – i-ta kolumna macierzy *U*, v_i – i-ta kolumna macierzy *V*, σ_i – i-ta wartość szczególna

Rozkład SVD macierzy (2)



Macierz zawiera nierosnący ciąg dodatnich wartości szczególnych. Wartości te odzwierciedlają „ważność” ukrytych cech opisujących dane zawarte w M. W praktyce wystarczy się ograniczyć do pierwszych najważniejszych cech reprezentujących 80% sumy wszystkich wartości szczególnych.



Analiza dokumentów tekstowych metodą LSA

termy / frazy w dokumentach

```
d1: modem the steering linux. modem, linux the modem. steering the modem. linux!
d2: linux; the linux. the linux modem linux. the modem, clutch the modem. petrol.
d3: petrol! clutch the steering, steering, linux. the steering clutch petrol. clutch the petrol; the clutch.
d4: the the the. clutch clutch clutch! steering petrol; steering petrol petrol; steering petrol!!!!
```

	d1	d2	d3	d4
linux	3	4	1	0
modem	4	3	0	1
the	3	4	4	3
clutch	0	1	4	3
steering	2	0	3	3
petrol	0	1	3	4

linux, modem – kojarzą się nam z pojęciem „komputer”
sprzęgło, sterowanie, benzyna – kojarzą się z pojęciem „samochód”

Analiza dokumentów tekstowych metodą LSA

```
d1: c a a b c b c
d2: a b c a b c c
d3: d e f f d
d4: f d e d f
```

Mamy 3 ukryte cechy (w tym dwie ważne).

macierz
dokumentów i termów

	d1	d2	d3	d4
a	2	2	0	0
b	2	2	0	0
c	3	3	0	0
d	0	0	2	2
e	0	0	1	1
f	0	0	2	2

$$\begin{vmatrix} 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 2 \end{vmatrix} = \begin{vmatrix} 0.57 & 0.00 & -0.82 \\ 0.45 & 0.00 & 0.32 \\ 0.68 & 0.00 & 0.48 \\ 0.00 & 0.67 & 0.00 \\ 0.00 & 0.33 & 0.00 \\ 0.00 & 0.67 & 0.00 \end{vmatrix} X \begin{vmatrix} 6.22 & 0.00 & 0.00 \\ 0.00 & 4.24 & 0.00 \\ 0.00 & 0.00 & 0.58 \end{vmatrix} X \begin{vmatrix} 0.66 & 0.75 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.71 & 0.71 \\ 0.75 & -0.66 & 0.00 & 0.00 \end{vmatrix}$$

A **U** Σ $\mathbf{V^T}$

LSA - interpretacja

V*∑	C1	C2	C3
d1	4.10	0	0.44
d2	4.67	0	-0.38
d3	0.00	3	0.00
d4	0.00	3	0.00

Interpretacja V*∑ - macierz opisująca dokumenty w przestrzeni ukrytych cech.
Kolor wskazuje na grupy dokumentów. Dokumenty, które są blisko w przestrzeni cech ukrytych dotyczą zbliżonej tematyki.

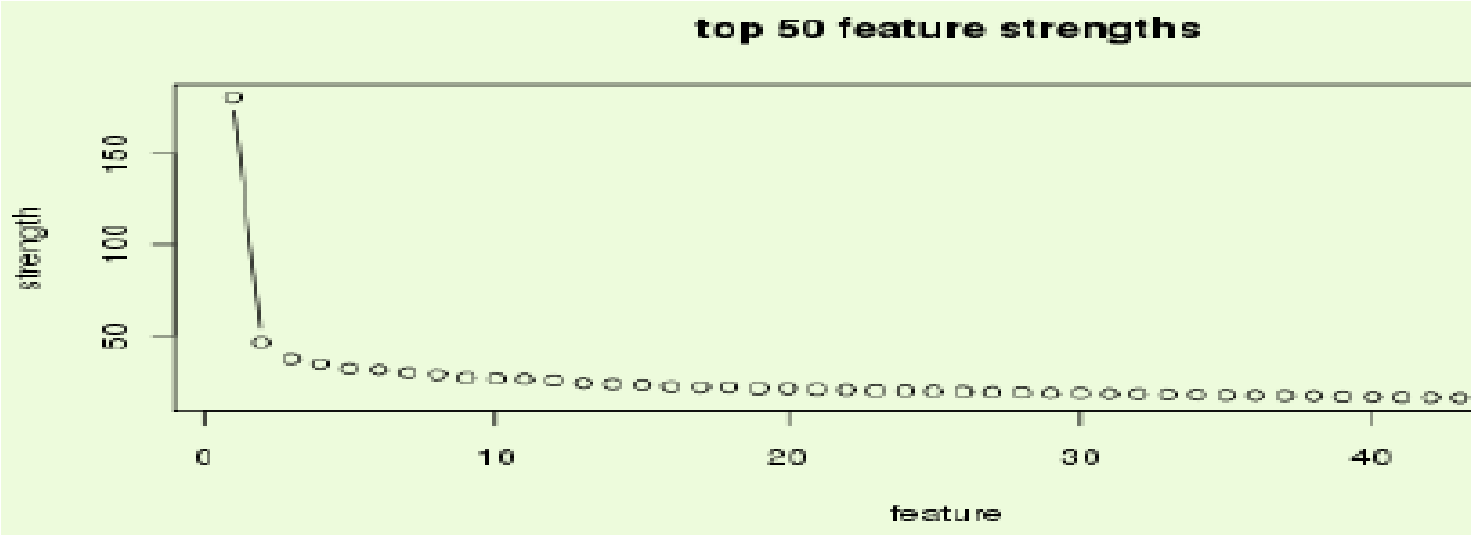
U*∑	C1	C2	C3
a	3.57	0.00	-0.47
b	2.82	0.00	0.18
c	4.23	0.00	0.28
d	0.00	2.83	0.00
e	0.00	1.41	0.00
f	0.00	2.83	0.00

Interpretacja U*∑ - macierz opisująca termy/frazy w przestrzeni ukrytych cech.
Kolor wskazuje na grupy fraz. Termy/frazy, które są blisko w przestrzeni cech ukrytych pochodzą ze zbliżonej tematyki.

LSA – przykład dane rzeczywiste

- Zbiór danych: po 100 artykułów

- autoblog (a automotive discussion blog) http://www.autoblog.com/
- perez hilton (a hollywood gossip blog) http://perezhilton.com/
- the register (a tech review blog) http://www.theregister.co.uk/



LSA – przykład dane rzeczywiste (2)

terms related to the first feature

of the 5700 terms present in the corpus which terms are strongest for the first feature?

rank	1	2	3	4	5	6	7	8	9	10
term	the	of	to	and	in	for	that	is	with	it
strength	138	46	45	43	32	25	25	22	16	16

at the tail end there are the hapax legomenon with near zero scores including terms like...
un, sydney, soa, jailed, worker, diplomat

- Słowa te mogą być wykorzystane do rozpoznawania języka.
- Dla analizy tekstów z jęz. angielskiego nie mają one znaczenia – nie pozwalają na odróżnienie dokumentów – występują powszechnie.