

# Uvod u obradu prirodnog jezika

## 2.1. Regularni izrazi

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

# Što su regularni izrazi?

- **Formalni jezik za specificiranje tekstualnih nizova.**
  - Pretpostavimo da u tekstu moramo pronaći riječ **sustav**
  - Ona se može izraziti na nekoliko načina
    - **sustav**
    - **sustavi**
    - **Sustav**
    - **Sustavi**
    - **sustavima**
    - **Sustavima**

# Pisanje regularnog izraza?

- Regularni izraz je oblika `/uzorak/zastavice`
- Regularni izrazi bez zastavica

Izraz	Primjer	Objašnjenje
<code>/a/</code>	Anamarija	prvo pojavljivanje znaka <code>a</code>
<code>/A/</code>	Anamarija	prvo pojavljivanje znaka <code>A</code>
<code>/am/</code>	Anamarija	prvo pojavljivanje niza <code>am</code>

- Zastavica `i` ignorira velike i male znakove

Izraz	Primjer	Objašnjenje
<code>/a/i</code>	Anamarija	prvo pojavljivanje malog ili velikog znaka <code>a</code>
<code>/Am/i</code>	Anamarija	prvo pojavljivanje niza <code>am</code> bez obzira na veličinu znaka, odnosno <code>am</code> , <code>AM</code> , <code>aM</code> ili <code>Am</code>

# Pisanje regularnog izraza?

- Zastavica **g** nastavlja globalnu pretragu

Izraz	Primjer	Objašnjenje
/a/g	Anamarija	sva pojavljivanja znaka <b>a</b>
/a/ig	Anamarija	sva pojavljivanja malog ili velikog znaka <b>A</b>

- Zastavica **i** ignorira velike i male znakove

Izraz	Primjer	Objašnjenje
/a/i	Anamarija	prvo pojavljivanje malog ili velikog znaka <b>a</b>
/Am/i	Anamarija	prvo pojavljivanje niza <b>am</b> bez obzira na veličinu znaka, odnosno <b>am</b> , <b>AM</b> , <b>aM</b> ili <b>Am</b>

# Regularni izrazi: Skup znakova

- skup znakova

Uzorak	Primjer	Objašnjenje
[sS]ustav	Sustavni sustavi	nizovi znakova koji počinju sa znakom <b>s</b> ili <b>S</b> i iza njega slijedi niz znakova <b>ustav</b>
[aeiou]	samoglasnici	svi samoglasnici <b>a e i o u</b>

- negirani skup znakova

znak **^** na početku označava negaciju samo kada je na prvom mjestu iza uglatih zagrada

Uzorak	Objašnjenje	
[^aeiou]	Danas u 3 sata.	svi znakovi koji nisu samoglasnici <b>a e i o u</b>
[^0123456789]	Danas u 3 sata.	svi znakovi koji nisu brođane znamenke <b>0 1 2 3 4 5 6 7 8 9</b>

# Regularni izrazi: Raspon znakova

- raspon znakova

znak – skup znamenaka u rasponu između dva znaka

Uzorak	Primjer	Objašnjenje
[A-Z]	Broj Pi = 3.14.	sva velika slova (bez "naših" znakova)
[a-z]	Broj Pi = 3.14.	sva mala slova (bez "naših" znakova)
[0-9]	Broj Pi = 3.14	sve brojčane znamenke ekvivalentno [0123456789]
[m-s]	Broj Pi = 3.14	sva mala slova od m do s

# Regularni izrazi: Specijalni znakovi

- rezervirani znakovi

znakovi `+ * ? ^ $ \ . [ ] { } ( ) | /` imaju specijalno značenje i potrebno je staviti `\` ispred njih

kod raspona znakova potrebno je staviti `\` ispred `-` ]

Uzorak	Primjer	Objašnjenje
<code>\+</code>	1 <code>+</code> 1 = 2	znak <code>+</code>
<code>[+\-]</code>	3 <code>+</code> 2 <code>-</code> 1 = 4	znak <code>+</code> ili znak <code>-</code>

- specijalni znakovi

Uzorak	Objašnjenje
<code>\t</code>	TAB znak (ASCII 9)
<code>\n</code>	LINE FEED (ASCII 10)
<code>\r</code>	CARRIAGE RETURN (ASCII 13)

# Regularni izrazi: Klase znakova

- točka

Uzorak	Primjer	Objašnjenje
.	Broj Pi = 3.14.	bilo koji znak osim novog reda ekvivalentno <code>[^\n\r]</code>

- klase znakova

Uzorak	Primjer	Objašnjenje
<code>\w</code>	Broj Pi = 3.14.	alfanumerički znakovi i <code>_</code> ekvivalentno <code>[A-Za-z0-9_]</code>
<code>\W</code>	Broj Pi = 3.14.	negacija od <code>\w</code> <code>[^A-Za-z0-9_]</code>
<code>\d</code>	Broj Pi = 3.14.	svi numerički znakovi ekvivalentno <code>[0-9]</code>
<code>\D</code>	Broj Pi = 3.14.	negacija od <code>\d</code> <code>[^0-9]</code>
<code>\s</code>	Broj Pi = 3.14.	prazni znakovi (razmak, novi red, tabulator)
<code>\S</code>	Broj Pi = 3.14.	negacija od <code>\w</code>



# Regularni izrazi: Kvantifikatori

- Kvantifikatori
  - + 1 ili više pojavljivanja
  - \* 0 ili više pojavljivanja
  - ? 0 ili jedno pojavljivanje

Uzorak	Primjer	Objašnjenje
e+	b be bee	jedno ili više pojavljivanja znaka e
r\w+	riba ribi grize rep	nizovi kojima je prvo slovo r i iza njega 1 ili više alfanumeričkih znakova
e*	b be bee	nula ili više pojavljivanja znaka e NAPOMENA: uključuje i prazne znakove
r\w*	riba ribi grize rep	nizovi kojima je prvo slovo r i iza njega 0 ili više alfanumeričkih znakova
past?i	pasi će pasti travu	nula ili jedno pojavljivanje znaka t

# Regularni izrazi: Grupiranje i alternacija

- Grupiranje i reference

Uzorak	Primjer	Objašnjenje
(ha)+	hahaha haa hah!	ha je grupa koji se ponavlja 1 ili više puta
(\w)a\1	pad mam sam gag	\1 se referencira na prvu grupu \w

- Alternacija |

Uzorak	Primjer	Objašnjenje
p(a e u)t	pat pet pit pot put	znak p iza kojeg može biti a e ili u i na kraju t
p(ame i)t	pametno piti	znak p iza kojeg može biti ame ili i i na kraju t

# Regularni izrazi: Sidra

- Početak i kraj
  - $\wedge$  početak linije
  - $\$$  kraj linije

Izraz	Primjer	Objašnjenje
<code>/^\w+/gm</code>	Jedan dva. Tri četiri.	alfanumerički znakovi na početku linije
<code>/\w+\.\$/gm</code>	Jedan dva. Tri četiri.	alfanumerički znakovi i točka na kraju linije

- granice riječi  $\b$   
granica između alfanumeričkog znaka i nealfanumeričkog znaka

Uzorak	Primjer	Objašnjenje
<code>\br</code>	riba ribi grize rep	r je na početku niza alfanumeričkih znakova
<code>[aeiou]\b</code>	riba ribi grize rep	a e i o u je na kraju niza alfanumeričkih znakova

# Primjeri

- Pronađite u tekstu sve instance riječi "on".

on

On je onda u ponoć otišao kući. Sutradan je on ponudio bonus.

[oO]n

On je onda u ponoć otišao kući. Sutradan je on ponudio bonus.

\b[oO]n\b

On je onda u ponoć otišao kući. Sutradan je on ponudio bonus.

# Regularni izrazi: Pogreške

proces koji smo upravo prošli temelji se na utvrđivanju dvije vrste pogrešaka

- lažno pozitivni (TIP I)
  - Odgovarajući nizovi koji se ne bi trebali podudarati.  
(onda, ponoć, bonus)
- lažno negativni (TIP II)
  - Ne označavanje nizova koji bi se trebali označiti.  
(On)

on

On je onda u ponoć otišao kući. Sutradan je on ponudio bonus.

# Regularni izrazi: Pogreške

- Obrada prirodnog jezika uvijek se bavi sljedećim pogreškama
- Smanjenje stupnja pogreške za aplikacije često uključuje dva pristupa rješavanja pogrešaka:
  - povećanje **točnosti** ili **preciznosti**  
(smanjenje lažno pozitivnih)
  - povećanje **pokrivenosti** ili **odziva**  
(smanjenje lažno negativnih)

# Regularni izrazi: Sažetak

- Regularni izrazi igraju iznenađujuće veliku ulogu
  - sofisticirani nizovi regularnih izraza često predstavljaju prvi model za bilo koju obradu teksta
- Za mnogo teže zadatke koriste se klasifikatori strojnog učenja
  - regularni izrazi se koriste kao obilježja u klasifikatorima
  - mogu biti vrlo korisni za obuhvaćanje općenitosti

# Uvod u obradu prirodnog jezika

## 2.2. Tokenizacija (opojavničenje) riječi i korpusi (Word tokenization and corpuses)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning



# Koliko ima riječi u rečenici?

- "Ja radim ovaaj uglav- uglavnom aaa obradu poslovnih podataka"
  - fragmenti, ispunjeni pauzom
- "mačka u šeširu je drugačija od drugih mačaka!"
  - **Lema**: isti korijen, dio govora, smisao grube riječi  
mačka i mačaka = ista lema
  - **Oblik riječi**: puni utjecaj oblika riječi  
mačka i mačaka = različita forma riječi

# Koliko ima riječi u rečenice?

"Oni leže na livadi u Dugom ratu i gledaju na druge livade"

- **Tip riječi:** jedinstveni element rječnika
- **Pojavnica (Token):** primjerak leme u tekstu
- Koliko ima riječi u rečenici
  - 12 pojavnica  
(ili 11 – Dugom ratu – jedna pojavnica)
  - 11 tipova  
(ili 9 – Dugom ratu – jedna pojavnica)  
(ili 8 – livadi i livade – ista lema)

# Koliko ima riječi u korpusu?

$N$  = broj pojava

$V$  = rječnik = skup tipova riječi

$|V|$  = broj tipova riječi u rječniku

Heapsov zakon = Herdanov zakon

$$|V| = kN^\beta \quad 0.67 < \beta < 0.75$$

	Pojavnice = $N$	Tip = $ V $
Centrala telefonskih razgovora	2.4 milijuna	20 tisuća
Shakespeare	884 000	31 tisuća
Google N-grams	1 trilijun	13 milijuna

# Korpus

- Riječi se ne pojavljuju od nigdje!
- Tekst nastaje
  - kod određenog pisca (pisaca),
  - u određeno vrijeme,
  - u određenoj varijaciji,
  - na određenom jeziku,
  - zbog određene funkcije.

# Korpusi variraju po dimenzijama

- **Jezik:** 7097 jezika na svijetu
- **Varijante:** čakavica, kajkavica, ...
- **Žanr:** novine, fikcija, znanstvi radovi, Wikipedia...
- **Demografija autora:** dob, spol, etična skupina...

# Izgradnja korpusa

- **Motivacija**

- Zašto je korpus napravljen?
- Tko ga je napravio?
- Tko je financirao izgradnju?

- **Situacija**

- Radi čega je tekst napisan?

- **Proces skupljanja**

- Ako je podsampliran, kako je podsampliran?
- Je li bilo konzensusa?
- Predobrada?

- **Proces anotacije, varijante jezika, demografija**

# Normalizacija teksta

- **Svaki zadatak obrade prirodnog jezika uključuje normalizaciju teksta:**
  1. Tokenizacija/segmentacija riječi u aktivnom tekstu
  2. Normalizacija formata riječi
  3. Segmentacija rečenica u aktivnom tekstu

# Tokenizacija po praznom znaku

- Jednostavan način tokenizacije
  - za jezike koji koriste razmak između riječi  
arapski, grčki, ćirilični, latinski... sustav pisanja
  - segmentiranje pojavnice između dva prazna znaka
- Unix alati za tokenizaciju po praznom znaku
  - "tr" naredba
  - za danu tekstualnu datoteku, ispiše pojavnice i njihove frekvencije



# Jednostavna tokenizacija u UNIX-u

- Za danu tekstualnu datoteku vraća pojavaice i njihove frekvencije

```
tr -sc "A-Za-zŠĐČĆŽšđčćž0-9" "\n" < alan_ford.txt | sort | uniq -c
```

5 će  
10 ćemo  
1 ćete  
4 ćeš  
17 ću  
1 Čeka  
1 Čekaj  
1 Čemu  
10 Čini  
1 Čitava  
1 Čuj  
1 Čujmo  
2 Čuo  
1 čahure  
...

Zamjena svih  
nealfanumeričkih  
znakova s novim  
redom

Sortiranje


Prebrojavanje  
jedinstvenih

# Jednostavna tokenizacija u UNIX-u

- Za danu tekstualnu datoteku vraća pojavnice i njihove frekvencije

```
tr -sc "A-Za-zŠĐČĆŽšđčćž0-9" "\n" < alan_ford.txt | sort | uniq -c | sort -n -r
```

```
109 je  
101 da  
97 se  
55 sam  
47 u  
46 i  
40 na  
35 za  
34 to  
31 ne  
27 li  
22 A  
20 mi  
...
```



Sortiranje po  
frekvenciji

# Problemi tokenizacije

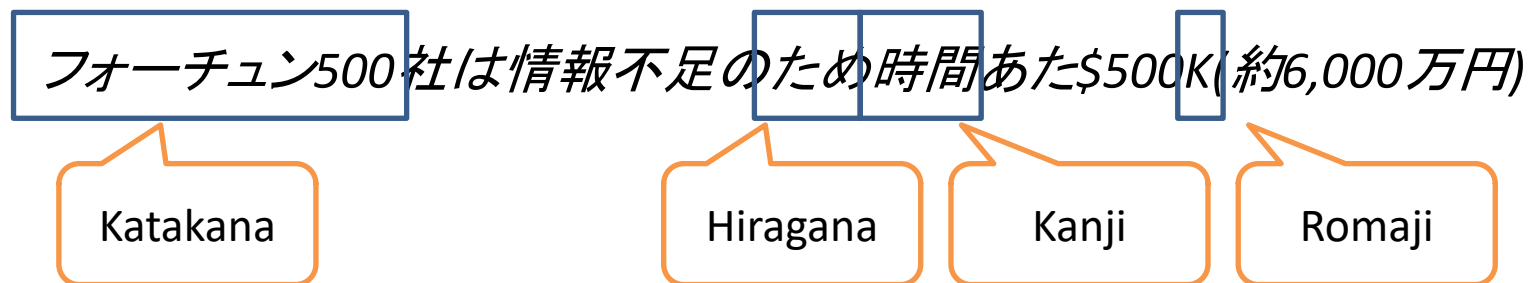
- Ne smiju se izbaciti svi interpunkcijski znakovi
  - km/s, dr. sc
  - cijene (\$59.99)
  - datumi (01.10.2021)
  - URL (<https://www.pmfst.hr>)
  - hashtag (#nlp)
  - email ([netko@pmfst.hr](mailto:netko@pmfst.hr))
- Klitike: riječi koje ne stoje same uz sebe
  - are u we're
- Kada višerječni izrazi postaju tokeni?
  - New York, bed & breakfast

# Tokenizacija kod jezika bez razmaka

- Francuski
  - **L'ensemble** jedna pojava ili dvije?
  - **L ? L' ? Le ?**
  - Težnja je da se **l'ensemble** opojavniči kao **ensemble**
- Njemačke imeničke složenice nisu segmentirane
  - **Lebensversicherungsgesellschaftsangestellter**
  - 'zaposlenik tvrtke za životno osiguranje'
  - pronalaženje informacija (information retrieval) u Njemačkom zahtjeva *razdvajanje složenica*

# Problemi kod jezika bez razmaka

- Kineski i japanski nemaju razmake između riječi
  - 伊万尼塞維奇现在居住在美国东南部的美国加州。
  - 伊万尼塞維奇 现在 居住 在 美国 东南部 的 美国加州
  - Ivanišević danas živi u US jugoistočnoj Kaliforniji
- U japanskom jeziku se pojavljuju riječi pisane drugim abecedama



- Korisnik može sve izraziti u Hiragana abecedi

# Tokenizacija riječi u kineskom

- je zapravo segmentacija riječi (Word segmentation)
- Kineske riječi se tvore od znakova
  - znakovi se tvore najčešće od jednog sloga i jednog morfema
  - prosječna riječ je duga 2.4 znaka
- Standardni algoritam za segmentaciju:
  - Maksimalno podudaranje – pohlepni algoritam (Maximum matching – greedy)

# Maksimalno podudaranje

Algoritam za segmentaciju riječi

- za danu listu riječi i za niz znakova
  1. stavi pokazivač na početak niza.
  2. pronadi najdulju riječ u rječniku koja odgovara niz s početkom u pokazivaču.
  3. pomaknite pokazivač preko riječi u nizu.
  4. idi na 2.

# Maksimalno podudaranje

Thecatinthehat

the cat in the hat

Thetabledownthere

the table down there

theta bled own there

- Nije primjenjivo za engleski jezik
- ali odlično radi za kineski
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
- Moderni probabilistički segmentacijski algoritmi još i bolje



# Uvod u obradu prirodnog jezika

## 2.3. Kodiranje uparivanjem byte-ova (Byte pair encoding)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

# Druga opcija za tokenizaciju

Umjesto

- segmentacije po praznom znaku
- segmentacija po jednom znaku

mogu se iskoristiti **podaci** da nam "kažu" kako tokenizirati

**Segmentacija podriječi**

(jer pojava nice mogu biti dio riječi, kao i sama riječ)

# Tokenizacija podriječi

Tri uobičajena algoritma:

- Kodiranje parova byte-ova (Byte-Pair Encoding (BPE)) (Sennrich et al., 2016)
- Unigram language modeling tokenization (Kudo, 2018)
- WordPiece (Schuster and Nakajima, 2012)

Svi imaju dva dijela:

- Učenje tokena koji uzima korpus za treniranje i inducira rječnik
- Segmentiranje tokena koji uzima sirove testne rečenice i tokenizira ih po rječniku

# BPE učenje tokena

U početku je rječnik skup individualnih znakova

= {A, B, C, D,..., a, b, c, d....}

Ponavljaj:

- izaberi dva najfrekventnija susjedna znaka u korpusu za treniranje (recimo 'A' i 'B')
  - dodaj novi spojeni simbol 'AB' u rječnik
  - zamijeni svaki susjedni 'A' 'B' u korpusu s 'AB' (spajanje)
- dok se ne napravi k spajanja.

# BPE dodatak

Većina algoritama podriječi se izvode nad razmakom odvojenim tokenima

Stoga se prvo dodaje specijalni znak za kraj riječi '\_' prije razmaka u korpusu za treniranje

Slijedi separacija u znakove

# BPE dodatak

Većina algoritama podriječi se izvode nad razmakom odvojenim tokenima

Stoga se prvo dodaje specijalni znak za kraj riječi '\_' prije razmaka u korpusu za treniranje

Slijedi separacija u znakove

low low low low low lowest lowest newer newer newer  
newer newer newer wider wider wider new new

# BPE primjer

low low low low low lowest lowest newer newer newer  
newer newer newer wider wider wider new new

## **korpus**

```
5   l o w _  
2   l o w e s t _  
6   n e w e r _  
3   w i d e r _  
2   n e w _
```

## **rječnik**

```
_ d e i l n o r s t w
```



# BPE primjer

spoji e r s er

**korpus**

```
5   l o w _  
2   l o w e s t _  
6   n e w e r _  
3   w i d e r _  
2   n e w _
```

**rječnik**

```
_ d e i l n o r s t w e r
```

spoji er \_ u er \_

**korpus**

```
5   l o w _  
2   l o w e s t _  
6   n e w e r _  
3   w i d e r _  
2   n e w _
```

**rječnik**

```
_ d e i l n o r s t w e r e r _
```

# BPE primjer

spoji **n e** u **ne**

**korpus**

```
5   l o w _  
2   l o w e s t _  
6   n e w e r _  
3   w i d e r _  
2   n e w _
```

**rječnik**

```
_ d e i l n o r s t w e r e r _ n e
```

# BPE primjer

Sljedeća spajanja su

## Spajanje

(ne, w)

(l, o)

(lo, w)

(new, er\_)

(low, \_)

## Rječnik

\_ d e i l n o r s t w e r e r \_ n e n e w

\_ d e i l n o r s t w e r e r \_ n e n e w l o

\_ d e i l n o r s t w e r e r \_ n e n e w l o l o w

\_ d e i l n o r s t w e r e r \_ n e n e w l o l o w n e w e r \_

\_ d e i l n o r s t w e r e r \_ n e n e w l o l o w n e w e r \_ l o w \_

# BPE primjer

## BPE segmentiranje

Na testnim podacima, pokreni svako spajanje naučeno nad trening podacima

- pohlepno
- u redoslijedu kako se učilo

Stoga, spaji svaki **e r** u **er**, onda svaki **er \_** u **er\_**, itd.

## Rezultat

- testni skup "n e w e r \_" će se tokenizirati kao puna riječ
- testni skup "l o w e r \_" će se tokenizirati kao "low er\_"

# BPE svojstva

Obično uključuje frekventne riječi  
i frekventne podriječi

- koje su često morfemi kao –est ili –er

**Morfem** je najmanja smisljena jedinica jezika

# Uvod u obradu prirodnog jezika

## 2.4. Normalizacija i izvlačenje korijena riječi (Word normalization and stemming)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

# Normalizacija riječi

- postavljanje riječi/tokena u standardni format
  - U.S.A. ili USA
  - uhhuh ili uh-huh
  - Fed ili fed
  - sam, smo, ste

# Promjena veličine slova (Case folding)

- Prilikom pronalaženja informacija često se velika slova prebacuju u mala
  - jer korisnici teže upotrebi malih slova
  - Mogući izuzeci: Veliko slovo u sredini rečenice?
    - npr., Srednja Dalmacija
    - CARNet
  - Za sentimentnu analizu, strojno učenje, ekstrakciju informacija
    - promjena veličine slova pomaže



# Lematizacija

- Smanjivanje infleksija ili varijanta oblika riječi na osnovni oblik (lema)
  - leti, lete, letim, leteći → letjeti
  - ptica, ptice, pticama, ptici → ptica
- *One ptice su visoko letjele → Onaj ptica biti visok letjeti*
- Lematizacija: traženje ispravnog oblika glavne riječi u rječniku

# Morfologija

- **Morfemi**
  - mali smisleni dijelovi riječi
  - **korijen riječi** (stem): temeljni dio
  - **afiksi**: dijelovi koji se dodaju korijenu riječi
    - često imaju gramatičke funkcije

# Izvlačenje korijena riječi (stemming)

- Smanjivanje oblika riječi na njegov korijen u pronalaženju informacija
- Korijen riječi se dobiva grubim cijepanjem afiksa
  - ovisno o jeziku
  - npr. **automati**, **automatski**, **automatizacija** se svodi na **automat**

*Splitska Kinoteka priredila je odličan program do kraja tjedna. Uz poznata filmska ostvarenja tu je i jedna manje razvikana filmska poslastica.*



Splitsk Kinotek priredi je odličan progra do kraj tjedn Uz poznat filmsk ostvarenj tu je i jedn manj razvikan filmsk poslastic

# Porterov algoritam

- Najčešći alat za izvlačenje korijena riječi u Engleskom jeziku

## Korak 1a

sses → ss	caresses → caress
ies → i	ponies → poni
ss → ss	caress → caress
s → ∅	cats → cat

## Korak 1b

(*v*)ing → ∅	walking → walk
	sing → sing
(*v*)ed → ∅	plastered → plaster
...	

## Korak 2 (za duge korijene)

ational → ate	relational → relate
izer → ize	digitizer → digitize
ator → ate	operator → operate
...	

## Korak 3 (za duže korijene)

al → ∅	revival → reviv
able → ∅	adjustable → adjust
ate → ∅	activate → activ
...	

# Uvod u obradu prirodnog jezika

## 2.4. Segmentacija rečenice i stabla odluke (Sentence segmentation and decision trees)

Branko Žitko

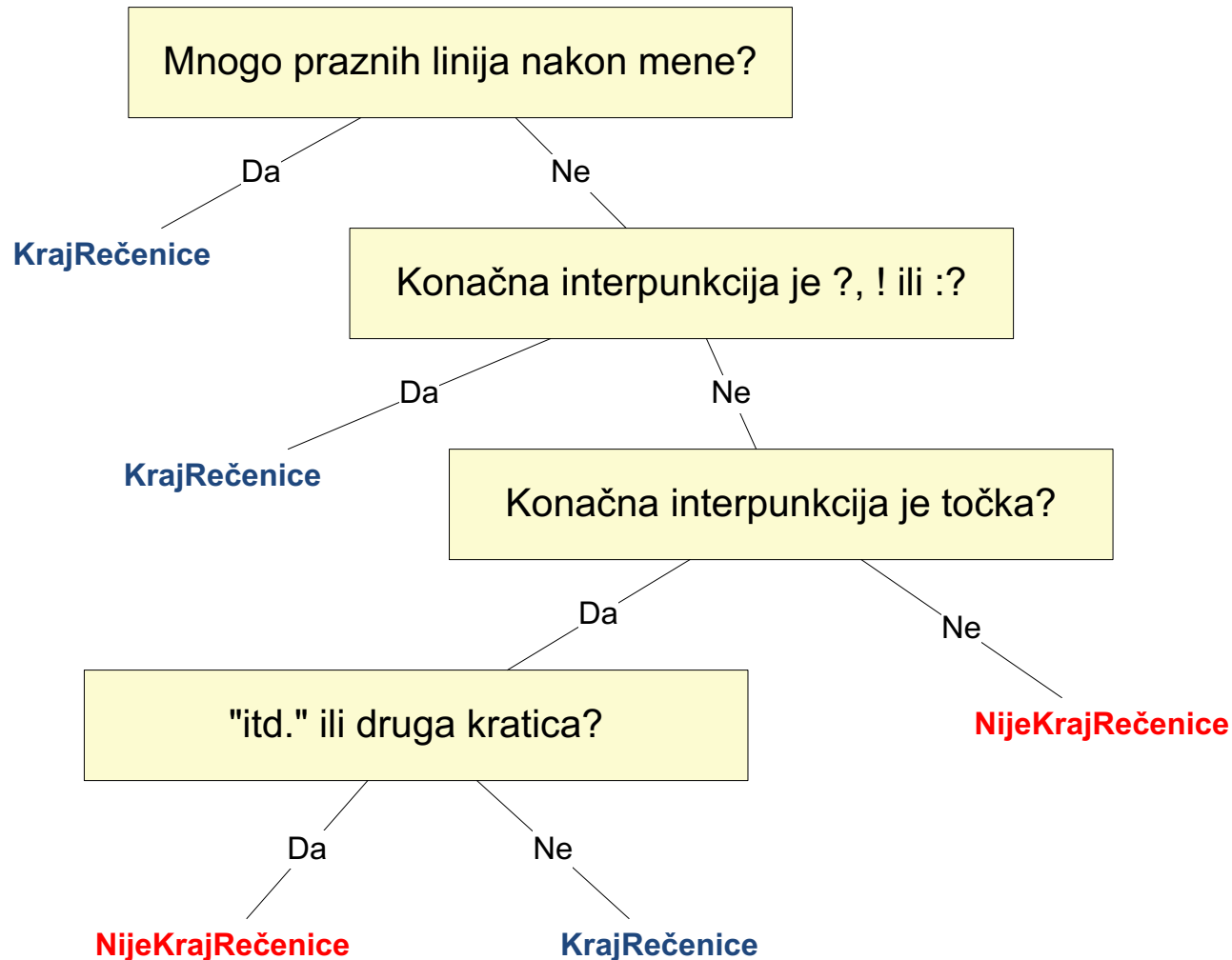
prevedeno od: Dan Jurafsky, Chris Manning

# Segmentacija rečenice

- **!, ?** – uglavnom jednoznačni
- **"."** – višeznačna i može označavati
  - kraj rečenice
  - kratice poput dr. itd.
  - brojeve poput .02% 4.3
- Sagraditi binarni klasifikator
  - koji traži **"."**
  - odlučuje jeli KrajRečenice/NijeKrajRečenice
  - klasifikatori: ručno pisana pravila, regularni izrazi ili strojno učenje

# Stablo odluke

- Odluka je li pojava predstavlja kraj rečenice.



# Profinjenje stabla odluke

- riječi s točkom:
  - mala slova, velika slova, prvo veliko slovo, broj
- riječi nakon točke:
  - mala slova, velika slova, prvo veliko slovo, broj
- Numeričke osobine:
  - duljina riječi s točkom
  - vjerojatnost (riječ s točkom se pojavljuje na kraju rečenice)
  - vjerojatnost (riječ nakon točke se pojavljuje na početku rečenice)