

Uvod u obradu prirodnog jezika

4.1. Uvod u n-grame

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Probabilistički modeli jezika

- Cilj: pridružiti vjerojatnost rečenici
 - Strojno prevođenje
 $P(\text{snažan vjetar večeras}) = P(\text{veliki vjetar večeras})$
 - Ispravljanje pravopisnih grešaka
Ured je oko petnaest **minueta** od moje kuće
 $P(\text{oko petnaest minuta od}) > P(\text{oko petnaest minueta od})$
 - Prepoznavanje govora
 $P(\text{vidio sam Ivanu}) \gg P(\text{vidi osam Ivan u})$
 - Sumarizacija, odgovaranje na pitanja, itd. itd.

Probabilističko modeliranje jezika

- Cilj: izračunati vjerojatnost rečenice ili niza riječi

$$P(W) = P(w_1, w_2, \dots, w_n)$$

- Slični zadaci: vjerojatnost sljedeće riječi

$$P(w_5 | w_1, w_2, w_3, w_4)$$

- Model koji izračunava

$P(W)$ ili $P(w_n | w_1, w_2, \dots, w_{n-1})$ se zove **model jezika**

- Bolji naziv bi bila **gramatika**, ali **model jezika MJ** je standard

Kako izračunati $P(W)$

- Kako izračunati ovu združenu vjerojatnost

$P(\text{njegova, voda, je, tako, čista, da})$

- Osloniti se na pravilo **lanca kod vjerojatnosti**

Pravilo lanca

- Definicija uvjetne vjerojatnosti

$$P(A|B) = \frac{P(A,B)}{P(B)} \quad \text{ili}$$

$$P(A|B)P(B) = P(A, B) \quad \text{ili}$$

$$P(A, B) = P(A|B)P(B)$$

- Više varijabli:

$$P(A, B, C, D) = P(A)P(B|A)P(C|A, B)P(D|A, B, C)$$

- Opće pravilo lanca

$$\begin{aligned} P(x_1 x_2 \dots x_n) &= P(x_1)P(x_2|x_1)P(x_3|x_1 x_2) \dots P(x_n|x_1 x_2 \dots x_{n-1}) \\ &= P(x_1)P(x_2|x_1)P(x_3|x_1^2) \dots P(x_n|x_1^{n-1}) \end{aligned}$$

Primjena lanca vjerojatnosti

$$P(w_1 w_2 \dots w_n) = P(w_1^n) = \prod_{k=1}^n P(w_k | w_1 \dots w_{k-1}) = \prod_{k=1}^n P(w_k | w_1^{k-1})$$

$P(\text{njegova voda je tako čista jer}) =$

$P(\text{njegova}) \times$

$P(\text{voda} | \text{njegova}) \times$

$P(\text{je} | \text{njegova voda}) \times$

$P(\text{tako} | \text{njegova voda je}) \times$

$P(\text{čista} | \text{njegova voda je tako}) \times$

$P(\text{jer} | \text{njegova voda je tako čista})$

Kako procijeniti vjerojatnosti

- Možemo li samo prebrojiti i podijeliti?

$$P(\text{je} | \text{njegova voda je tako čista jer}) = \frac{c(\text{njegova voda je tako čista jer je})}{c(\text{njegova voda je tako čista jer})}$$

- Ne možemo! Jer ima previše mogućih rečenica, ali i previše rečenica koje se ne pojavljuju.

Markovljeva pretpostavka

- pojednostavljenje pretpostavke

$$P(\text{je} | \text{njegova voda je tako čista jer}) \approx P(\text{je} | \text{jer})$$

- ili možda

$$P(\text{je} | \text{njegova voda je tako čista jer}) \approx P(\text{je} | \text{čista jer})$$

Markovljeva pretpostavka

$$P(w_1 \dots w_n) \approx \prod_i P(w_i | w_{i-k}^{i-1})$$

- drugim riječima, aproksimiramo svaku vrijednost u produktu

$$P(w_i | w_1^{i-1}) \approx P(w_i | w_{i-k}^{i-1})$$

Najjednostavniji slučaj: Unigram

$$P(w_1 \dots w_n) \approx \prod_i P(w_i)$$

- neke generirane rečenice iz unigram modela

Hmm Sviđa pred čitava vama

Dakle vodom 315 momče pothvat lopova posljednji nisu manje

Da pucao zapijevajte koga

Šesta dobiti golubarnik ostane

Zapamti slobode

Moj ključ nije izvjesne duboka odvedite

Nesretniče organizaciju mikrofilmom uspjeti zajedničkog ispričam

Otkuda igle kotač znam opasnost tanjurima

Vidiš stoji aviona ostao

Čuj evo kontakt ubijati

Reci ubijen čitavu mušterija sreće

bigram model

- Uvjet prethodne riječi

$$P(w_1 \dots w_n) \approx \prod_i P(w_i | w_{i-1})$$

- neke generirane rečenice iz bigram modela

Trebalo je vjerojatno smaragd od hizmara.

Jedi rižu i mikrofilm s tanjurima.

Nadam se agencija Ford reklamni crtež.

Ostatak hoću natrag.

Tako se okrenuti sklopit ću na električnu stolicu.

Imaš li drugih mana.

Ulovio sam sretan da ga šefe ja putujem u posljednji čas Miss.

Jesi li vi bolji način silaženja s x zrakama.

Oni tipovi.

Lisičine nisu doprli ispušni plinovi.

n-gram modeli

- Možemo proširiti na trigram, 4-gram, 5-gram...
- Općenito, svi ovi modeli su nedovoljni
 - jer jezik ima **daleke ovisnosti**

"Računalo koje sam stavio u dnevnu sobu na petom katu se srušilo"

- Ali često se možemo zadovoljiti s n-gram modelom

Uvod u obradu prirodnog jezika

4.2. Procjena vjerojatnosti N-grama

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Procjena vjerojatnosti bigrama

- Procjena maksimalne izglednosti
(MLE maximum likelihood estimation)

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum_{\mathbf{w}} C(w_{i-1}\mathbf{w})}$$

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

Primjer

- Procjena maksimalne izglednosti

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

<s> ja sam Ivan</s>

<s> Ivan ja sam</s>

<s> ja ne volim kuhana jaja i šunku </s>

$$P(\text{ja}|\text{<s>}) = \frac{2}{3} = 0.67 \quad P(\text{Ivan}|\text{s}) = \frac{1}{3} = 0.33 \quad P(\text{sam}|\text{ja}) = \frac{2}{3} = 0.67$$

$$P(\text{</s>}|\text{Ivan}) = \frac{1}{2} = 0.5 \quad P(\text{Ivan}|\text{sam}) = \frac{1}{2} = 0.55 \quad P(\text{ne}|\text{ja}) = \frac{1}{3} = 0.33$$

Primjer

<s> ja sam Ivan </s>

<s> Ivan ja sam </s>

<s> ja ne volim jesti </s>

<s>	ja	sam	Ivan	</s>	ne	volim	jesti
3	3	2	2	3	1	1	1

	<s>	ja	sam	Ivan	</s>	ne	volim	jesti
<s>		2		1				
ja			2			1		
sam				1	1			
Ivan		1			1			
</s>								
ne							1	
volim								1
jesti					1			

Primjer

<s> ja sam Ivan </s>

<s> Ivan ja sam </s>

<s> ja ne volim jesti </s>

<s>	ja	sam	Ivan	</s>	ne	volim	jesti
3/16	3/16	2/16	2/16	3/16	1/16	1/16	1/16

	<s>	ja	sam	Ivan	</s>	ne	volim	jesti
<s>		2/3		1/3				
ja			2/3			1/3		
sam				1/2	1/2			
Ivan		1/2			1/2			
</s>								
ne							1/1	
volim								1/1
jesti					1/1			

Još primjera

- neke rečenice iz Berkeley Restaurant Project (BeRP)
 - can you tell me about any good cantonese restaurants close by
 - mid priced thai food is what i'm looking for
 - tell me about chez panisse
 - can you give me a listing of the kinds of food that are available
 - i'm looking for a good place to eat breakfast
 - when is caffe venezia open during the day

Broj bigrama

- Od 9222 rečenica ($V = 1446$)

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	5	827	0	9	0	0	0	2
želim	2	0	608	1	6	6	5	1
da	2	0	4	686	2	0	6	211
jedem	0	0	2	0	16	2	42	0
kinesku	1	0	0	0	0	82	1	0
hranu	15	0	15	0	1	4	0	0
ručak	2	0	0	0	0	1	0	0
potrošim	1	0	1	0	0	0	0	0

Bigram vjerojatnosti

- Normalizacija po unigramu

ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
2533	927	2417	746	158	1093	341	278

- Rezultat

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	0,001974	0,32649	0	0,003553	0	0	0	0,00079
želim	0,002157	0	0,655879	0,001079	0,006472	0,006472	0,005394	0,001079
da	0,000827	0	0,001655	0,283823	0,000827	0	0,002482	0,087298
jedem	0	0	0,002681	0	0,021448	0,002681	0,0563	0
kinesku	0,006329	0	0	0	0	0,518987	0,006329	0
hranu	0,013724	0	0,013724	0	0,000915	0,00366	0	0
ručak	0,005865	0	0	0	0	0,002933	0	0
potrošim	0,003597	0	0,003597	0	0	0	0	0

bigram procjene vjerojatnosti rečenice

$$P(<s> \text{ Ja želim domaću hranu} </s>) =$$

$$P(\text{Ja} | <s>)$$

$$\times P(\text{želim} | \text{Ja})$$

$$\times P(\text{domaću} | \text{želim})$$

$$\times P(\text{hranu} | \text{domaću})$$

$$\times P(</s> | \text{hranu})$$

$$= 0.000031$$

Koje vrste znanja?

$$P(\text{domaću} \mid \text{želim}) = 0.0011$$

$$P(\text{kinesku} \mid \text{želim}) = 0.0065$$

$$P(\text{da} \mid \text{želim}) = 0.66$$

$$P(\text{jedem} \mid \text{da}) = 0.28$$

$$P(\text{hranu} \mid \text{da}) = 0$$

$$P(\text{želim} \mid \text{potrošim}) = 0$$

$$P(\text{ja} \mid \langle s \rangle) = 0.25$$

Praktični problemi

- Sve se radi u logaritamskom prostoru
 - izbjegavanje prelijeva ispod granice realnih brojeva
 - zbrajanje je brže nego množenje

$$p_1 \times p_1 \times p_3 \times p_4 = \exp(\log(p_1) + \log(p_2) + \log(p_1) + \log(p_2))$$

Uvod u obradu prirodnog jezika

4.3. Evaluacija i perpleksija

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Evaluacija: koliko je dobar naš model?

- Da li naš model jezika više preferira dobre rečenice ili one loše?
 - pridruživanje veće vjerojatnosti "realnim" ili "često promatranim" rečenicama
 - nego "ne gramatičkim" ili "rijetko promatranim" rečenicama
- Treniramo model na **trening skupu**
- Testiramo performanse nad neviđenim podacima
 - **Testni skup** je neviđen skup podataka različit od trening skupa
 - **Evaluacijska metrika** govori koliko je dobar model nad testnim skupom

Vanjska evaluacija n-gram modela

- Najbolja evaluacija za usporedbu modela A i B
 - Staviti svaki model da radi
 - ispravak pravopisnih grešaka,
 - prepoznavanje govora,
 - strojno prevođenje
 - Prikupiti rezultate preciznosti modela A i B
 - koliko je pogrešnih riječi ispravljeno u točne riječi
 - koliko je prepoznatih riječi
 - koliko riječi je točno prevedeno
 - Usporediti preciznost od modela A i B

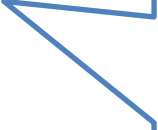
Teškoće vanjske evaluacije n-gram modela

- Vanjska (in-vivo) evaluacija
 - dugotrajna: može trajati danima, tjednima...
- Stoga
 - ponekad koristiti **unutarnju** evaluaciju: **perpleksija**
 - Loša aproksimacija
 - osim ako testni podaci izgledaju kao podaci za trening
 - Općenito korisna samo u pilot eksperimentima
 - Ali je dobra za razmišljanje

Intuicija perpleksije

- Shannonova igra:
 - Koliko dobro ćemo predvidjeti sljedeću riječ?

Uvijek naručujem picu sa sirom i _____



gljivama 0.1
šunkom 0.1
inćunima 0.01
...
rižom 0.0001
...

- Unigrami su očajni kod ove igre...
- Bolji model za tekst
 - je onaj koji pridružuje veću vjerojatnost riječi koja se zapravo pojavila
- Perpleksija je zapravo **težinski faktor grananja**.

Perpleksija

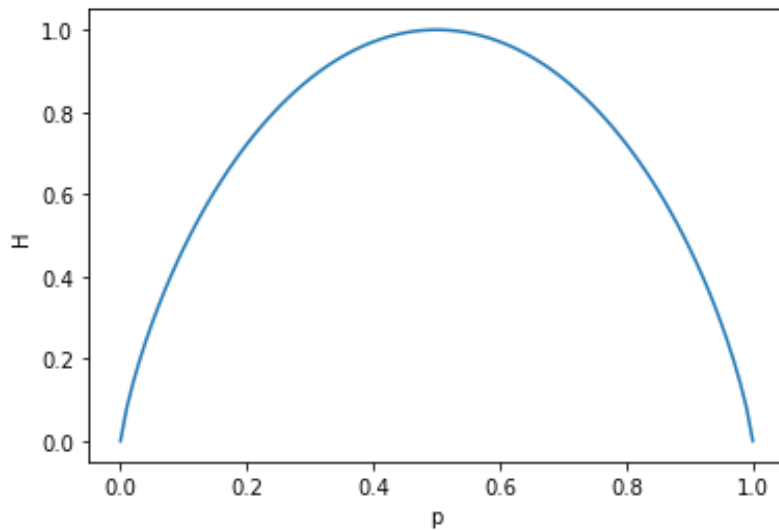
- Najbolji model jezika je onaj koji najbolje predviđa neviđeni testni skup
 - daje najveću vjerojatnost rečenici
- **Perpleksija** je inverzna vjerojatnost testnog skupa W normaliziranog po broju riječi

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

- pravilo lanca $PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$
- za bigram $PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_{i-1})}}$
- minimizacija perpleksije je isto što i maksimizacija uvjetne vjerojatnosti

Perpleksija - teorija informacija

- **Entropija** – mjera neizvjesnosti distribucije p
- $H(p) = -\sum_i p_i \log_2 p_i$
- mjera "bit"
- za distribuciju $p, 1 - p$



Perpleksija - teorija informacija

- **Unakrsna entropija** (cross entropy) – entropija distribucija p, q
- $H(p, q) = -\sum_i p_i \log_2 q_i$
- $H(p, q) = H(p) + D_{KL}(p||q)$
- **Kullback-Leibler divergencija** – udaljenost između distribucija p, q
- $D_{KL}(p||q) = -\sum_i p_i \log_2 \frac{p_i}{q_i}$
- Ako se ne zna p , pretpostavi se da je uniformna distribucija (maksimalna entropija)
- $H(p, q) = -\frac{1}{N} \sum_{i=1}^N \log_2 q_i$

Perpleksija - teorija informacija

- Neka je T testni skup
- Unakrsna entropija modela p
- $H_p(T) = -\frac{1}{|T|} \sum_{i=1}^{|T|} \log_2 p(w_i)$
- Perpleksija modela p – mjera koliko dobro model p predviđa T
- $$PP_p(T) = 2^{H_p(T)} = 2^{-\frac{1}{|T|} \sum_{i=1}^{|T|} \log_2 p(w_i)} = 2^{-\frac{1}{|T|} \log_2 \prod_{i=1}^{|T|} p(w_i)}$$
$$= \left(2^{\log_2 \prod_{i=1}^{|T|} p(w_i)} \right)^{-\frac{1}{|T|}} = \left(\prod_{i=1}^{|T|} p(w_i) \right)^{-\frac{1}{|T|}} = \sqrt[|T|]{\frac{1}{\prod_{i=1}^{|T|} p(w_i)}}$$

Perpleksija kao faktor grananja

- Pretpostavimo da se rečenica sastoji od brojčanih znamenka
- Koja je perpleksija ovih rečenica prema modelu koji pridružuje $P = \frac{1}{10}$ svakoj znamenki?

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \left(\frac{1}{10^N}\right)^{-\frac{1}{N}} = \frac{1}{10}^{-1} = 10$$

Perpleksija kao faktor grananja

- Pretpostavimo da je znamenka 0 frekventnija
- U trening skupu:
 - znamenka 0 se pojavljuje 91 put $P(0) = 0.91$
 - ostale znamenke se pojavljuju točno jedanput $P(w) = 0.01$

0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0

0 0 0 2 0 0 0 0 0 0

0 0 0 0 0 3 0 0 0 0

0 0 4 0 0 0 0 0 0 0

0 0 0 0 0 0 0 5 0 0

0 0 0 6 0 0 0 0 0 0

0 0 0 0 7 0 0 0 0 0

0 0 0 0 0 0 0 0 8 0

9 0 0 0 0 0 0 0 0 0

$$P(0) = \frac{91}{100} \quad P(w \neq 0) = \frac{1}{100}$$

$$P(0|0) = \frac{81}{91} \quad P(w \neq 0|0) = \frac{1}{91}$$

$$P(0|w \neq 0) = \frac{1}{1}$$

$$P(0|) = \frac{1}{1} \quad P(w \neq 0|) = \frac{0}{91}$$

Perpleksija kao faktor grananja

- Pretpostavimo da je znamenka 0 frekventnija
- U trening skupu:
 - znamenka 0 se pojavljuje 91 put
 - ostale znamenke se pojavljuju točno jedanput
- Testni skup je rečenica $W=0000030000$
- Kolika je perpleksija?

- Za unigram

$$PP(W) = P(0000030000)^{-\frac{1}{10}} = \sqrt[10]{0.91^{-9} * 0.01^{-1}} \approx 1.73$$

- Za bigram

$$\begin{aligned} PP(W) &= \left(P(0|)P(0|0)^7 P(0|3)P(3|0) \right)^{-\frac{1}{10}} \\ &= \sqrt[10]{* \frac{1^{-1}}{1} * \frac{81^{-7}}{91} * \frac{1^{-1}}{91} * \frac{1^{-1}}{1}} \approx 1.70 \end{aligned}$$

Manja perpleksija = bolji model

- Wall Street Journal
 - Trening od 38 miliona riječi
 - Test od 1.5 miliona riječi

N-gram	Unigram	Bigram	Trigram
perpleksija	962	170	109

- Što nam više informacija n-gram daje to je manja perplexija (veća izvjesnost)

Uvod u obradu prirodnog jezika

4.4. Generalizacija i nule

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Shannon-ova metoda vizualizacije

- Izaberi slučajni bigram (<s>,w) prema svojoj vjerojatnosti
- Sada izaberi slučajni bigram (w,x) prema svojoj vjerojatnosti
- I tako dalje sve dok se ne dođe do </s>
- Zatim spoji riječi u rečenicu

```
<s> Ja
    Ja želim
        želim da
            da jedem
                jedem kinesku
                    kinesku hranu
                        hranu </s>
```

Ja želim da jedem kinesku hranu.

Aproksimacija Shakespeare-a

Unigram

swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
Every enter now severally so, let
Hill he late speaks; or! a more to leg less first yon enter
Are where exeunt and sighs have rise excellency took of .. Sleep knave we. near; vile like

bigram

What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and namre bankrupt, nor the first gentleman?

Trigram

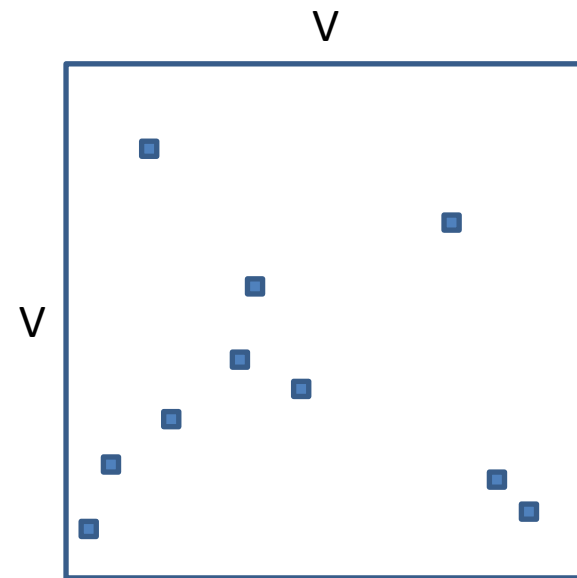
Sweet prince, Falstaff shall die. Hany of Monmouth's grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

4-gram

King Henry. What' I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv 'd
in;
Will yon not tell me who I am?
It cannot be but so.
Indeed the short and the long. Many, 'tis a noble Lepidns.

Shakespeare kao korpus

- $N=884647$ pojava, $V=29066$ riječi
- Shakespeare je producirao 300000 bigram tipova od $V^2=844832356$ mogućih bigrama.
 - Stoga 99.96% mogućih bigrama nikad nisu viđeni (nule u tablici)
- 4-grami su još gori



Opasnosti prekoračenja

- n-grami su dobri za predviđanje riječi samo ako je testni korpus sličan korpusu treninga
 - U stvarnosti, to često nije slučaj
 - Potrebno je napraviti trening robusnih modela i zatim generalizirati!
 - Jedna vrsta generalizacije: Nule!
 - stvari koje se uopće ne pojavljuju u skupu za trening
 - ali se pojavljuju u testnom skupu

Skup za trening	Testni skup
... negirao je optužbe negirao je izvještaje negirao je tvrdnje negirao je zahtjeve negirao je ponude negirao je posudbe ...

$$P(\text{ponude} \mid \text{negirao je}) = 0$$

Bigrami s nultom vjerojatnošću

- Pridruživanje vjerojatnosti 0 na testnom skupu
- Stoga se ne može izračunati perpleksija! (dijeljenje s 0)

Nepoznate riječi

- **Otvoreni** rječnik protiv **zatvorenog** rječnika
- Ako unaprijed znamo sve riječi
 - rječnik V je fiksni
 - zadaci **zatvorenog** rječnika
- Često ne znamo sve riječi
 - riječi izvan rječnika
 - zadaci **otvorenog** rječnika
- Kreira se pojava za nepoznate riječi <UNK> - unknown (OOV out of vocabulary riječi)

Nepoznate riječi

- Treniranje vjerojatnosti OOV riječi kod otvorenog rječnika
 - kreiranje **fiksnog leksikona** L veličine V
 - prilikom normalizacije teksta, svaka riječ iz trening skupa koja nije u L se promijeni u $\langle \text{UNK} \rangle$
 - sada se njihove vjerojatnosti treniraju kao i za ostale riječi
- Prilikom dekodiranja
 - koristi se $\langle \text{UNK} \rangle$ za svaku riječ koja nije u podacima za trening
- Ako nemamo fiksni leksikon, onda se implicitno stvara leksikon mijenjanjem riječi iz trening skupa u $\langle \text{UNK} \rangle$ temeljem njihove frekvencije.
 - Možemo zamijeniti s $\langle \text{UNK} \rangle$ sve riječi iz skupa za treniranje koje se javljaju manje od n puta (n je neki mali broj)

Uvod u obradu prirodnog jezika

4.5. Izgladaivanje: dodaj jedan (Laplaceovo izgladaivanje)

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Ideja izgladivanja (prema Dan Klein-u)

- Kada imamo spremnu statistiku

$P(w | \text{negirao je})$

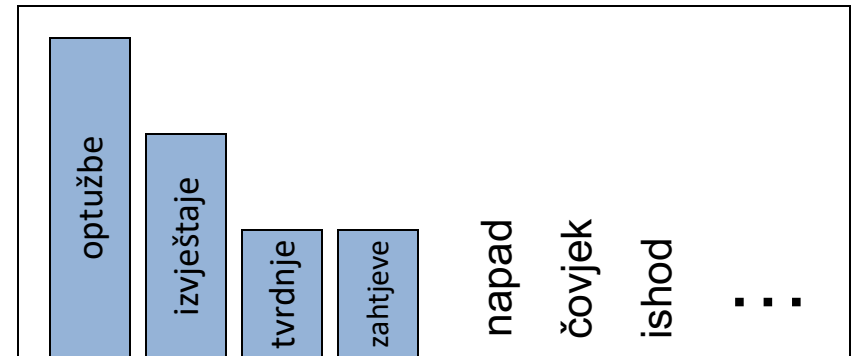
3 optužbe

2 izvještaje

1 tvrdnje

1 zahtjeve

UKUPNO: 7



- Uzmi masu vjerojatnosti radi bolje generalizacije

$P(w | \text{negirao je})$

2.5 optužbe

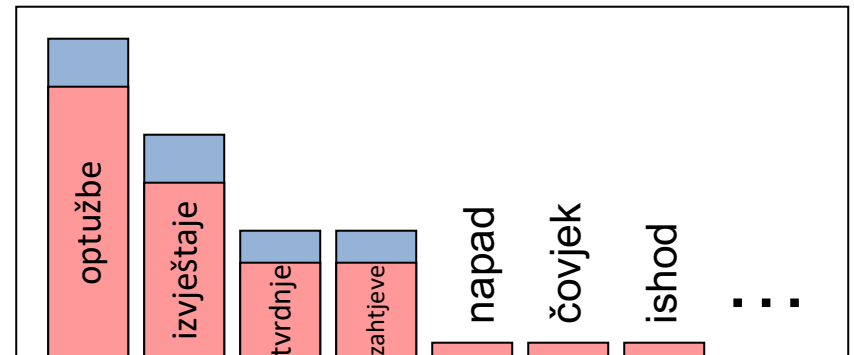
1.5 izvještaje

0.5 tvrdnje

0.5 zahtjeve

2 ostalo

UKUPNO: 7



Dodaj jedan - procjena

- ili Laplaceovo izgladivanje
- Pretvaramo se da smo svaku riječ vidjeli još jedan put
- Dodaj jedan kod svih prebrojavanja!
- Maksimalna izvjesnost (MLE)
 - unigram $P_{MLE}(w_i) = \frac{C(w_i)}{N}$
 - bigram $P_{MLE}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$
- Dodaj 1 (Laplace)
 - unigram $P_{Laplace}(w_i) = \frac{c_i+1}{N+V}$
 - bigram $P_{Laplace}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)+1}{\sum_w (C(w_{i-1}w)+1)} = \frac{C(w_{i-1}w_i)+1}{C(w_{i-1})+V}$

Dodaj jedan - procjena

- Prilagođeni brojač c^*

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V} = \frac{c_i^*}{N} \Rightarrow c_i^* = (c_i + 1) \frac{N}{N + V}$$

$$P_{Laplace}(w_i | w_{i-1}) = \frac{C(w_{i-1}w_i) + 1}{C(w_{i-1}) + V} = \frac{c^*(w_{i-1}w_i)}{C(w_{i-1})} \Rightarrow$$

$$c^*(w_{i-1}w_i) = \frac{(C(w_{i-1}w_i) + 1) \times C(w_{i-1})}{C(w_{i-1}) + V}$$

Berkley restaurant korpus

- s primijenjenim Laplaceovim izgladivanjem

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	6	828	1	10	1	1	1	3
želim	3	1	609	2	7	7	6	2
da	3	1	5	687	3	1	7	212
jedem	1	1	3	1	17	3	43	1
kinesku	2	1	1	1	1	83	2	1
hranu	16	1	16	1	2	5	1	1
ručak	3	1	1	1	1	2	1	1
potrošim	2	1	2	1	1	1	1	1

Laplaceovo izgladivanje

- Unigram

ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
2533	927	2417	746	158	1093	341	278

- $V = 1446$

$$P_{Laplace}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	0,001505	0,207675	0,000251	0,002508	0,000251	0,000251	0,000251	0,000752
želim	0,00126	0,00042	0,255775	0,00084	0,00294	0,00294	0,00252	0,00084
da	0,000775	0,000258	0,001292	0,177474	0,000775	0,000258	0,001808	0,054766
jedem	0,000455	0,000455	0,001364	0,000455	0,007727	0,001364	0,019545	0,000455
kinesku	0,001241	0,00062	0,00062	0,00062	0,00062	0,051489	0,001241	0,00062
hranu	0,006282	0,000393	0,006282	0,000393	0,000785	0,001963	0,000393	0,000393
ručak	0,001671	0,000557	0,000557	0,000557	0,000557	0,001114	0,000557	0,000557
potrošim	0,001155	0,000577	0,001155	0,000577	0,000577	0,000577	0,000577	0,000577

Rekonstruirani brojač

$$c^*(w_{i-1}w_i) = \frac{(C(w_{i-1}w_i) + 1) \times C(w_{i-1})}{C(w_{i-1}) + V}$$

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	3,8	527	0,64	6,4	0,64	0,64	0,64	1,9
želim	1,2	0,39	238	0,78	2,7	2,7	2,3	0,78
da	1,9	0,63	3,1	430	1,9	0,63	4,4	133
jedem	0,34	0,34	1	0,34	5,8	1	15	0,34
kinesku	0,2	0,098	0,098	0,098	0,098	8,2	0,2	0,098
hranu	6,9	0,43	3,9	0,43	0,86	2,2	0,43	0,43
ručak	0,57	0,19	0,19	0,19	0,19	0,38	0,19	0,19
potrošim	0,32	0,16	0,32	0,16	0,16	0,16	0,16	0,16

Usporedba s početnim bigramom

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	5	827	0	9	0	0	0	2
želim	2	0	608	1	6	6	5	1
da	2	0	4	686	2	0	6	211
jedem	0	0	2	0	16	2	42	0
kinesku	1	0	0	0	0	82	1	0
hranu	15	0	15	0	1	4	0	0
ručak	2	0	0	0	0	1	0	0
potrošim	1	0	1	0	0	0	0	0

	ja	želim	da	jedem	kinesku	hranu	ručak	potrošim
ja	3,8	527	0,64	6,4	0,64	0,64	0,64	1,9
želim	1,2	0,39	238	0,78	2,7	2,7	2,3	0,78
da	1,9	0,63	3,1	430	1,9	0,63	4,4	133
jedem	0,34	0,34	1	0,34	5,8	1	15	0,34
kinesku	0,2	0,098	0,098	0,098	0,098	8,2	0,2	0,098
hranu	6,9	0,43	3,9	0,43	0,86	2,2	0,43	0,43
ručak	0,57	0,19	0,19	0,19	0,19	0,38	0,19	0,19
potrošim	0,32	0,16	0,32	0,16	0,16	0,16	0,16	0,16

Dodaj 1 procjena je tupi alat

- Dodaj 1 se praktički ne koristi kod n-grama:
 - vidjet ćemo bolje metode
- Međutim dodaj 1 se koristi kod izgladivanja u drugim modelima obrade prirodnog jezika
 - za klasifikaciju teksta
 - u domenama gdje je mali broj nula

Dodaj k izgladivanje

- Alternativa dodaj 1 izgladivanju je pomicanje manjeg dijela vjerojatnosti sa viđenih na neviđene događaje.
- Umjesto dodaj 1, dodat će se decimalni broj k npr. 0.5, 0.05, 0.1

$$P_{dodaj-k}^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + k}{C(w_{n-1}) + kV}$$

- k se često odabire optimizacijom na razvojnem skupu podataka

Trening podaci

Razvojni podaci

Testni podaci

Uvod u obradu prirodnog jezika

4.6. Interpolacija i odustajanje

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Interpolacija i odustajanje

- Ponekad pomaže korištenje **manjeg** sadržaja
- Uvjeti nad manjim sadržajem za sadržaje o kojima se ne zna
- **Odustajanje (Backoff):**
 - koristi trigram ako imaš dobre dokaze, inače bigram, inače unigram
- **Interpolacija:**
 - pomiješaj unigram, bigram, trigram
- Interpolacija radi bolje

Linearna interpolacija

- jednostavna interpolacija

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) &= \lambda_3 P(w_n | w_{n-2} w_{n-1}) \\ &\quad + \lambda_2 P(w_n | w_{n-1}) \\ &\quad + \lambda_1 P(w_n)\end{aligned}$$

$$\sum_i \lambda_i = 1$$

- lambda je uvjetovan sadržajem

$$\begin{aligned}\hat{P}(w_n | w_{n-2} w_{n-1}) &= \lambda_3 (w_{n-2}^{n-1}) P(w_n | w_{n-2} w_{n-1}) \\ &\quad + \lambda_2 (w_{n-1}^{n-1}) P(w_n | w_{n-1}) \\ &\quad + \lambda_1 (w_n^{n-1}) P(w_n)\end{aligned}$$

Kako odrediti lambda?

- Koristeći **razvojni** korpus



- Izaberi lambde tako da maksimiziraš vjerojatnost razvojnog korpusa
 - namjesti n-gram vjerojatnosti na korpusu za trening
 - zatim odredi lambde tako da daje najveće vjerojatnosti nad razvojnim korpusom

$$\log P(w_1 \dots w_n | M(\lambda_1 \dots \lambda_k)) = \sum_i \log P_{M(\lambda_1 \dots \lambda_k)}(w_i | w_{i-1})$$

Kako odrediti lambda?

- **Metodom maksimiziranja očekivanja (Expectation Maximization)**
- H – razvojni skup
- minimizacija $\frac{-1}{|H|} \sum_{i=1}^{|H|} \log_2 \hat{p}_\lambda(w_i|h_i)$ nad λ gdje je

$$\begin{aligned} \hat{p}_\lambda(w_i|h_i) = \hat{p}_\lambda(w_i|w_{i-2}w_{i-1}) = & \lambda_3 p_3(w_n|w_{n-2}w_{n-1}) \\ & + \lambda_2 p_2(w_n|w_{n-1}) \\ & + \lambda_1 p_1(w_i) \\ & + \frac{\lambda_0}{|V|} \end{aligned}$$

- Izračun očekivanja za $j = 0 \dots 3$

$$c(\lambda_j) = \sum_{i=0}^{|H|} \frac{\lambda_j p_j(w_i|h_i)}{\hat{p}_\lambda(w_i|h_i)}$$

- sljedeći lambda za $j = 0 \dots 3$

$$\lambda_{j,next} = c(\lambda_j) / \sum_{k=0 \dots 3} c(\lambda_k)$$

Kako odrediti lambda?

- **Metoda maksimiziranja očekivanja**

- Sirova distribucija nad rječnikom $V=\{a, b, c, \dots, z\} \mid |V|=26$

$$\begin{aligned} p(a) &= 0.25, & \hat{p}_\lambda(w_i|h_i) &= \hat{p}_\lambda(w_i|w_{i-2}w_{i-1}) = \lambda_3 p_3(w_n|w_{n-2}w_{n-1}) \\ p(b) &= 0.5, & & + \lambda_2 p_2(w_n|w_{n-1}) \\ p(w) &= 1/64 \text{ za } w \in \{c, \dots, r\} & & + \lambda_1 p_1(w_i) \\ p(w) &= 0 \text{ za } w \in \{s, \dots, z\} & & + \frac{\lambda_0}{|V|} \end{aligned}$$

- Razvojni skup **H = babu** (λ_1 unigram, λ_0 uniformno)

- Počni s $\lambda_1 = 0.5$ i $\lambda_0 = 0.5$

$$p'_\lambda(b) = 0.5 * 0.5 + 0.5 / 26 = 0.27$$

$$p'_\lambda(a) = 0.5 * 0.25 + 0.5 / 26 = 0.145$$

$$p'_\lambda(u) = 0.5 * 0 + 0.5 / 26 = 0.02$$

$$c(\lambda_j) = \sum_{i=0}^{|H|} \frac{\lambda_j p_j(w_i|h_i)}{\hat{p}_\lambda(w_i|h_i)}$$

$$c(\lambda_1) = 0.5 * 0.5 / 0.27 + 0.5 * 0.25 / 0.145 + 0.5 * 0.5 / 0.27 + 0.5 * 0 / 0.02 = 2.72$$

$$\begin{aligned} c(\lambda_0) &= 0.5 * 0.04 / 0.27 + 0.5 * 0.04 / 0.145 + 0.5 * 0.04 / 0.27 + 0.5 * 0.04 / 0.02 \\ &= 1.28 \end{aligned}$$

- Normaliziraj $\lambda_1=0.68$, $\lambda_0=0.32$

- Ponavljaj sve dok nove lambde se malo razlikuju od prethodnih lambdi (npr. < 0.01)

Veliki n-grami na Web-u

- Kako se postaviti prema velikim n-gram korpusima
 - Obrezivanje (Pruning)
 - spremi samo n-grame koji imaju broj pojavljivanja $>$ prag
 - ukloniti jedinice od n-grama višeg reda
 - obrezivanje temeljeno entropijom
 - Povećanje efikasnosti
 - efikasne podatkovne strukture kao "prefiksna stabla" (tries)
 - Bloom-ovi filteri: aproksimativni modeli jezika
 - spremanje riječi preko indeksa, a ne stringova
 - Huffmanovo kodiranje za spremanje velikog broja riječi u 2 bajta
 - kvantiziranje vjerojatnosti (4-8 bitova umjesto broja s pomičnim zarezom od 8 bajtova)

Izgladaivanje kod velikih n-grama

- Glupo odustajanje (Stupid backoff)
- Nema popuštanja, ali se koriste relativne frekvencije

$$S(w_i | w_{i-k+1}^{i-1}) = \begin{cases} \frac{c(w_{i-k+1}^i)}{c(w_{i-k+1}^{i-1})} & \text{ako } c(w_{i-k+1}^i) > 0 \\ 0.4 \times S(w_i | w_{i-k+2}^{i-1}) & \text{u suprotnom} \end{cases}$$

$$S(w_i) = \frac{c(w_i)}{N}$$

Ugladivanje n-grama

- Dodaj jedan:
 - Dobro kod kategorizacije teksta
 - Nije dobro za modeliranje jezika
- Najčešće korištena metoda:
 - Proširenje Kneser-Ney interpolacije
- Za velike n-grame:
 - glupo odustajanje

Napredno modeliranje jezika

- Diskriminativni modeli:
 - izbor težina n-grama radi poboljšanja zadatka, a ne radi prilagođavanju skupu za trening
- Modeli temeljeni na parsiranju
- Modeli s predpohranom (Caching models)
 - nedavno korištene riječi imaju veću vjerojatnost da se pojave

$$P_{CACHE}(w|history) = \lambda P(w_i|w_{i-2}w_{i-1}) + (1 - \lambda) \frac{c(w \in history)}{|history|}$$

- ali se pokazala jako lošom metodom kod prepoznavanja govora

Uvod u obradu prirodnog jezika

4.7. Good-Turing izgladivanje

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Napredni algoritmi za izgladivanje

- Ideja većine algoritama izgladivanja
 - Good-Turing
 - Kneser-Ney
 - Witten-Bell
- Upotrijebi broj "stvari" koje smo **jednom vidjeli**
- kako bi procijenili broj "stvari" koje **nikad nismo vidjeli**

Notacija: N_c = frekvencija od frekvencije c

- N_c = broj riječi koje smo vidjeli c puta

Ivan sam ja sam ja Ivan ja nikad ne jedem

ja	3
sam	2
Ivan	2
nikad	1
ne	1
jedem	1

$$N_3 = 1$$

$$N_2 = 2$$

$$N_1 = 3$$

Ideja Good-Turing izgladivanja

- Loviš ribu i ulovio si:
 - 10 srdela, 3 oslića, 2 bukve, 1 zubaca, 1 gofu, 1 komarču = 18 riba
- Koliko je vjerojatno da će sljedeća riba biti zubatac?
 $1/18$
- Koliko je vjerojatno da će sljedeća riba biti neka nova?
 $3/18$ (jer $N_1 = 3$)
- S obzirom na to, koja je vjerojatnost da će sljedeća riba biti zubatac?
 - Mora biti manja od $1/18$
 - Kako to procijeniti

Good Turing proračun

$$P_{GT}^*(\text{riječi s frekvencijom } 0) = \frac{N_1}{N}$$

$$c^* = \frac{(c+1)N_{c+1}}{N_c}$$

- Nije viđeno (npr. Salpa)

$$c = 0$$

$$P_{MLE}(\text{Salpa}) = \frac{0}{18} = 0$$

$$P_{GT}^*(\text{Salpa}) = \frac{N_1}{N} = \frac{3}{18}$$

- Viđeno jednom (npr. Zubatac)

$$c = 1$$

$$P_{MLE}(\text{Zubatac}) = \frac{1}{18} = 0.055$$

$$c^*(\text{Zubatac}) = \frac{(1+1)*N_{1+1}}{N_1} = \frac{2*N_2}{N_1} = \frac{2*1}{3} = \frac{2}{3}$$

$$P_{GT}^*(\text{Zubatac}) = \frac{c^*}{N} = \frac{\frac{2}{3}}{18} = \frac{1}{27}$$

Good-Turing brojevi

- Brojevi iz Church & Gale (1991)
- 22×10^6 riječi iz AP Newswire

$$c^* = \frac{(c + 1)N_{c+1}}{N_c}$$

- relativni popust

$$d_c = \frac{c^*}{c}$$

- apsolutni popust

$$d_c = |c^* - c| \approx 0.75 \text{ za } c > 1$$

c	c^*
0	0.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

Uvod u obradu prirodnog jezika

4.7. Kneser-Ney izgladivanje

Branko Žitko

prevedeno od: Dan Jurafsky, Chris Manning

Apsolutni popust

- Uštedimo na vremenu i jednostavno oduzmimo 0.75 (ili neki d)

$$P_{\text{ApsolutniPopust}}(w_i|w_{i-1}) = \frac{c(w_{i-1}w_i) - d}{c(w_{i-1})} + \lambda(w_{i-1})P(w_i)$$

Diagram illustrating the components of the Absolute Discount formula:

- bigram sa popustom**: Points to the numerator $c(w_{i-1}w_i) - d$.
- težina interpolacije**: Points to the interpolation weight $\lambda(w_{i-1})$.
- unigram**: Points to the unigram probability $P(w_i)$.

- Možda koristiti ekstra vrijednosti od d za bigrame koji se pojavljuju 1 put)
- ali hoćemo li uopće koristiti unigram vjerojatnosti $P(w)$?

Kneser-Ney izgladivanje I

- Bolja procjena vjerojatnosti za unigrame nižeg reda!
 - Shannon-ova igra: *Loše vidim bez svojih čitaćih* _____?
 - "Zeland" je češći od "naočala,,
 - ... ali "Zeland" uvijek slijedi iza "Novi"
- Unigram je koristan ako nismo prije vidjeli bigram!
- Umjesto $P(w)$: "Koliko vjerojatno je w "
- $P_{NASTAVAK}(w)$: "Koliko je vjerojatno da se w pojavi kao novi nastavak"
 - za svaku riječ w prebroji sve bigrame koje upotpunjuje

Kneser-Ney izgladivanje II

- Koliko puta se w pojavljuje kao novi nastavak:

$$P_{NASTAVAK}(w) \propto |\{w_{i-1} : c(w_{i-1}w) > 0\}|$$

- Normaliziran ukupnim brojem bigram tipova

$$|\{(w_{j-1}w_j) : c(w_{j-1}w_j) > 0\}|$$

$$P_{NASTAVAK}(w) = \frac{|\{w_{i-1} : c(w_{i-1}w) > 0\}|}{|\{(w_{j-1}w_j) : c(w_{j-1}w_j) > 0\}|}$$

Kneser-Ney izgladivanje III

- Broj tipova riječi viđenih da prethode w

$$|\{w_{i-1} : c(w_{i-1}w) > 0\}|$$

- Normaliziran brojem riječi koje prethode svim riječima:

$$P_{NASTAVAK}(w) = \frac{|\{w_{i-1} : c(w_{i-1}w) > 0\}|}{\sum_{w'} |\{w'_{i-1} : c(w'_{i-1}w') > 0\}|}$$

- Česta riječ "Zeland" koja se često nalazi iza "Novi" će imati nisku vjerojatnost nastavka

Kneser-Ney izgladivanje IV

$$P_{KN}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1})P_{NASTAVAK}(w_i)$$

- λ je normalizacijska konstanta; količina vjerojatnosti koju smo izbacili

$$\lambda(w_{i-1}) = \frac{d}{c(w_{i-1})} |\{w : c(w_{i-1}w) > 0\}|$$

normalizirani popust

broj tipova riječi koji mogu slijediti iza w_{i-1}
= # tipova riječi koje smo izbacili
= # puta koliko smo primijenili normalizirani popust

Kneser-Ney izgladivanje: rekurzivni oblik

$$P_{KN}(w_i | w_{i-n+1}^{i-1}) = \frac{\max(c_{KN}(w_{i-n+1}^i) - d, 0)}{c_{KN}(w_{i-n+1}^{i-1})} + \lambda(w_{i-n+1}^{i-1})P_{KN}(w_i | w_{i-n+2}^{i-1})$$

$$c_{KN} = \begin{cases} broj(*) \text{ za veći red} \\ brojnastavka(*) \text{ za manji red} \end{cases}$$

- *brojnastavka* = broj jedinstvenih sadržaja s jednom riječju od *