1

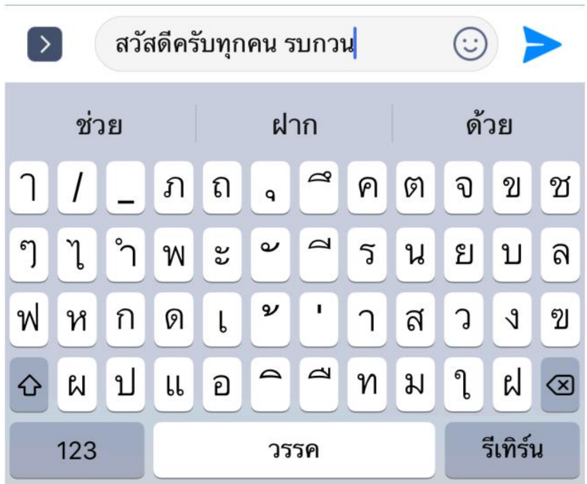
LANGUAGE MODEL

บริบททางภาษา

2

- He had **beef** for lunch vs He had **a beef** for lunch
- อา|นอน|ตาก|ลม vs อาน|อน|ตา|กลม
- I send him a letter vs I send dim a led her
- กรุงเทพมีตึกสูงเยอะ
  - Bangkok has many **high** buildings
  - Bangkok has many **tall** buildings

3



LM ใช้ทำอะไร

4

- หาความน่าจะเป็น/ความเป็นไปได้ของประโยค
- ไวยากรณ์โดยไม่ต้องเขียนกฎโดยตรง
- ทำนายคำถัดไปโดยใช้บริบท

5

N-Gram Language Model

Model แบบง่ายที่สุด

6

- Unigram Language Model

  P(w1, w2, w3, …, wn) = P(w1) P(w2) P(w3) .. P(wn)

- Bangkok has many high buildings

  P(Bangkok,has,many,high,buildings) =

  P(Bangkok)P(has)P(many) P(high) P(buildings)

- Bangkok has many tall buildings

  P(Bangkok,has,many,tall,buildings) =

  P(Bangkok) P(has) P(many) P(tall) P(buildings)

Language Model แบบมีบริบท

7

- Bigram Language Model
- P(w1, w2, w3, …, wn) = P(w1 | START) P(w2 | w1) P(w3 | w2) .. P(wn | wn-1)
- Bangkok has many high buildings

  P(Bangkok,has,many,high,buildings) =

  P(Bangkok | START) P(has | Bangkok) P(many | has)

  P(high | many) P(buildings | high)
- Bangkok has many tall buildings

  P(Bangkok,has,many,tall,buildings) =

  P(Bangkok | START) P(has | Bangkok) P(many | has)

  P(tall | many) P(buildings | tall)

Bigram Language Model

8

texaco, rose, one, in, this, issue, is, pursuing, growth, in,
a, boiler, house, said, mr., gurria, mexico, 's, motion,
control, proposal, without, permission, from, five, hundred,
fifty, five, yen

outside, new, car, parking, lot, of, the, agreement, reached
this, would, be, a, record, november

Trigram and 4-gram LM

9

- Trigram Language Model

P(w1, w2, w3, …, wn) = P(w1 | START1, START2)
P(w2 | START2, w1)
P(w3 | w1 w2)
P(w4 | w2 w3) .. P(wn | wn-2 wn-1)

P(tall | has many)

- 4-gram Language Model

P(w1, w2, w3, …, wn) = P(w1 | START1, START2, START3)
P(w2 | START2, START3, w1)
P(w3 | START3 w1 w2)
P(w4 | w1 w2 w3) .. P(wn | wn-3 wn-2 wn-1)

โมเดลมันก็ไม่ฉลาดอยู่ดี

10

- Long distance dependencies (e.g. relative clauses)
  - The computers that I bought from the new mall are/is broken.

- 5-gram ดีๆส่วนใหญ่มักจะเพียงพอ

การประมาณค่า Unigram Probability

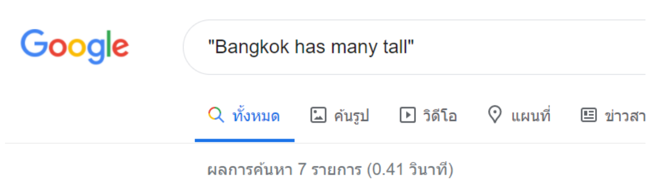
11

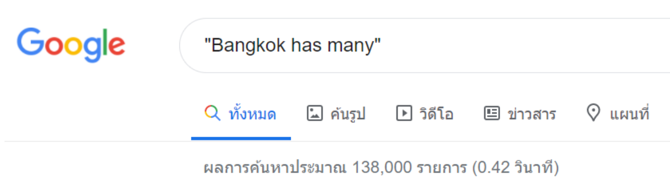
P(Bangkok) = C(Bangkok) / จำนวนคำทั้งหมด

การประมาณค่า Conditional Probability

12

$$P(\text{tall} \mid \text{Bangkok has many}) \approx \frac{C(\text{Bangkok has many tall})}{C(\text{Bangkok has many})} = 7 / 138\text{k}$$





13

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67$ $\quad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33$ $\quad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$

$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5$ $\quad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5$ $\quad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$

Raw bigram counts

14

- Out of 9222 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Raw bigram probabilities

15

- Normalize by unigrams:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

- Result:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Bigram estimates of sentence probabilities

16

P(<s> I want english food </s>) =

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

× P(</s>|food)

= .000031

What kinds of knowledge?

17

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

Practical Issues

18

- We do everything in log space
  - Avoid underflow
  - (also adding is faster than multiplying)

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

19

Smoothing Language Model

Zeros

20

- Training set:
  ... denied the allegations
  ... denied the reports
  ... denied the claims
  ... denied the request

  P(“offer” | denied the) = 0

- Test set
  ... denied the offer
  ... denied the loan

Zero probability bigrams

21

- Bigrams with zero probability
  - mean that we will assign 0 probability to the test set!
- And hence we cannot compute perplexity (can't divide by 0)!

Add-one estimation

22

- Also called Laplace smoothing
- Pretend we saw each word one more time than we did
- Just add one to all the counts!
- MLE estimate: $$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$
- Add-1 estimate: $$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$$

Berkeley Restaurant Corpus: Laplace smoothed bigram counts

23

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Laplace-smoothed bigrams

24

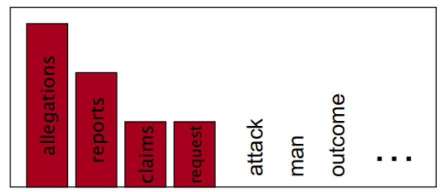
$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

25

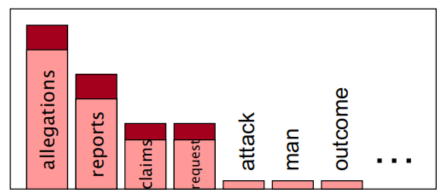
- When we have sparse statistics:

  P(w | denied the)
  3 allegations
  2 reports
  1 claims
  1 request
  7 total



- Steal probability mass to generalize better

  P(w | denied the)
  2.5 allegations
  1.5 reports
  0.5 claims
  0.5 request
  2 other
  7 total



Kneser-Ney Smoothing

26

- Better estimate for probabilities of lower-order unigrams!
  - Shannon game: *I can't see without my reading* ___glasses / Francisco___ ?
  - "Francisco" is more common than "glasses"
  - … but "Francisco" always follows "San"

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

λ is a normalizing constant; the probability mass we've discounted

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

the normalized discount

The number of word types that can follow w_{i-1}
= # of word types we discounted
= # of times we applied normalized discount

27