N-Gram

N-gram

Given the Corpus:

1. He is a nice guy
2. It’s nice to get your letter
3. I am going to meet you
4. I would like a meat sandwich

Estimate the probability of the following two sentences (with smoothing):

1. Nice to meet you
2. Nice to meat you

Start & End tags

Add Start & End Tag:

<s> He is a nice guy <e>

<s> It’s nice to get your letter <e>

…

The bigrams:

<s> He, He is, is a, a nice, nice guy, guy <e>

<s> It’s, It’s nice, nice to, to get, get your, your letter, letter <e>

…

Estimate the probability

P(nice to meet you)

= P(nice|<s>)P(to|nice)P(meet|to)P(you|meet)P(<e>|you)

p(nice|<s>)	p(to|nice)	p(meat|to)	p(meet|to)	p(you|meat)	p(you|meet)	p(<e>|you)
C(<s> nice) / (C(<s>he) + C(<s>it's) + C(<s>I)						
0/4						

Estimate the probability

p(nice|<s>)	p(to|nice)	p(meat|to)	p(meet|to)	p(you|meat)	p(you|meet)	p(<e>|you)
C(<s> nice) / (C(<s>he) + C(<s>it's) + C(<s>I)	C(nice to) / C(nice guy) + C(nice to)	C(to meat) / C(to get) + C(to meet)	C(to meet)/ C(to get) + C(to meet)	C(meat you)/ C(meat sandwich)	C(meet you)/ C(meet you)	C(you <e>)/ C(you <e>)
0/4	1/2	0/2	1/2	0/1	1/1	1/1

Smoothing

Techniques to tackle data sparseness problem

Decreasing the probability of seen events

Adding a little bit of probability mass over for unseen events

Probabilistic distribution becomes smoother

Smoothing by adding one to frequencies of all events

p(nice|<s>)	**p(nice|<s>) smoothed**
C(<s> nice) / (C(<s>he) + C(<s>it's) + C(<s>I)	C(<s> nice) + 1 / (C(<s>he) + 1 + C(<s>it's) + 1 + C(<s>I) + 1
0/4	1/7

Smoothing by Laplace's law

If the size of the vocabulary is given as V, we can apply the Laplace smoothing.

For example, given V = 20

p(nice|<s>)	**p(nice|<s>) smoothed by Laplace's law**
C(<s> nice) / (C(<s>he) + C(<s>it's) + C(<s>I)	C(<s> nice) + 1 / (C(<s>he) + C(<s>it's) + C(<s>I) + **V**
0/4	1/ 24

Convolutional Neural Network

Question 1

Perform the convolution operation (no padding, stride=1)

0	1	0	0	0
1	1	1	1	0
0	1	1	0	1
1	0	1	0	0
1	1	0	1	0

Input matrix

-1	-1	-1
-1	8	-1
-1	-1	-1

Filter (kernel)

Question 1 - Answer

Perform the convolution operation (no padding, stride=1)

0	1	0	0	0
1	1	1	1	0
0	1	1	0	1
1	0	1	0	0
1	1	0	1	0

Input matrix

-1	-1	-1
-1	8	-1
-1	-1	-1

Filter (kernel)

3	3	5
2	3	-5
-6	4	-4

Convolved feature

Question 2

Perform the average pooling (pool size = 2x2, stride=1)

0	1	0	0	0
1	1	1	1	0
0	1	1	0	1
1	0	1	0	0
1	1	0	1	0

Input matrix

Question 2 - Answer

Perform the average pooling (stride=2)

0	1	0	0	0
1	1	1	1	0
0	1	1	0	1
1	0	1	0	0
1	1	0	1	0

Input matrix

0.75	0.75	0.5	0.25
0.75	1	0.75	0.5
0.5	0.75	0.5	0.25
0.75	0.5	0.5	0.25

Result

Information Retrieval

Given Inverted Index Matrix, and the query: w1 and w3 and not w4

	D1	D2	D3	D4
w1	0	0	1	1
w2	1	0	0	0
w3	1	1	1	1
w4	1	0	0	1

w1 and w3 and not w4

= 0011 & 1111 & 0110

=0010 (Document D3 will be retrieved)

	D1	D2	D3	D4
w1	0	0	1	1
w2	1	0	0	0
w3	1	1	1	1
w4	1	0	0	1

TF-IDF

d1 = big cats are nice and funny

d2 = small dogs are better than big dogs

d3 = small cats are afraid of small dogs

d4 = big cats are not afraid of small dogs

d5 = funny cats are not afraid of small dogs

Fill in the TF IDF matrix

TF-IDF

d1 = big cats are nice and funny

d2 = small dogs are better than big dogs

d3 = small cats are afraid of small dogs

d4 = big cats are not afraid of small dogs

d5 = funny cats are not afraid of small dogs

$$w_j^i = {tf_j}^i \cdot idf_j = {tf_j}^i \cdot \log \frac{N}{df_j}$$

	big	cat	nice	and	funny	small	dog
d1	$\log_2(5/3)$						
d2							
d3							

	big	cat	nice	and	funny	small	dog
d1	$\log_2(5/3)$	$\log_2(5/4)$	$\log_2(5/1)$	$\log_2(5/1)$	$\log_2(5/2)$	0	0
d2	$\log_2(5/3)$	0	0	0	0	$\log_2(5/4)$	$2\log_2(5/4)$
d3	0	$\log_2(5/4)$	0	0	0	$2\log_2(5/4)$	$\log_2(5/4)$