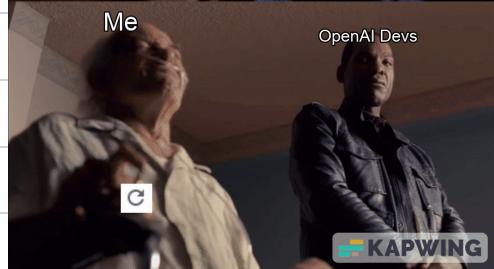
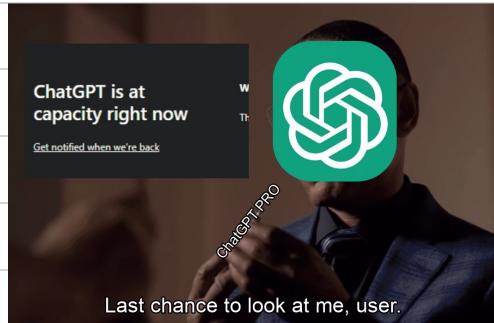


What people think programming  
is like



What programming is actually like



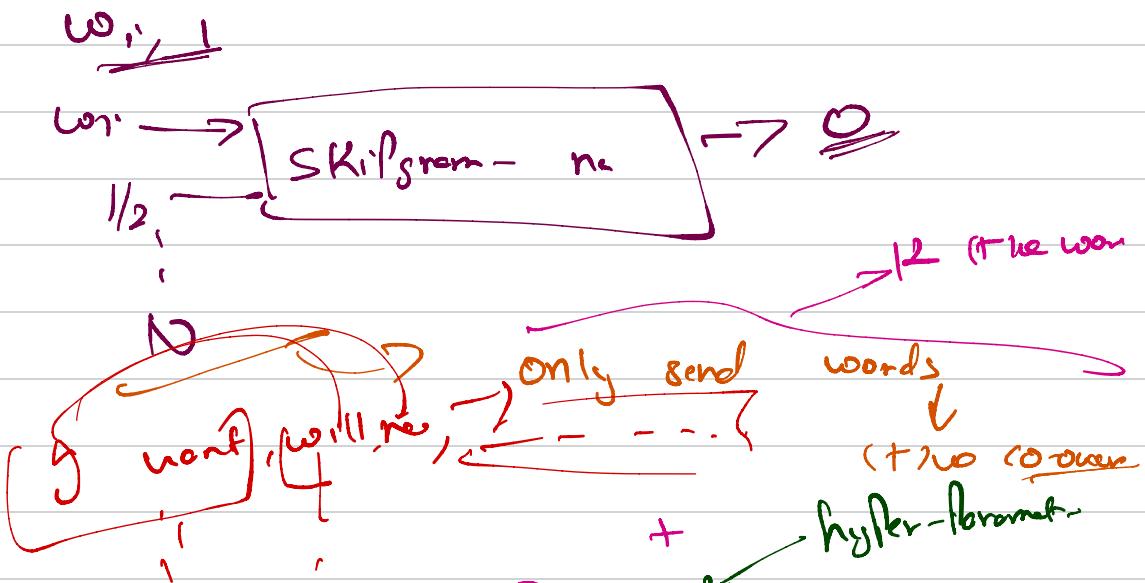
## Agenda

1. Probabilistic Models
2. N-Gram Introduction
3. Evaluation Metrics
4. Ambiguity in NLP

# Slip-gram with negative sampling

(Co-occurrence matrix  $\rightarrow$  U · U<sup>T</sup> & Perk)

$$w_i \rightarrow \begin{matrix} 1, 2, 3 \\ 0 0 0 \\ - - - \end{matrix} \quad N \quad 1$$



Send ( $5 \times 12$ ) - negative samples

	me	has	want	will	<u>new</u>	samples
1	0	0	56	23	15	$12 = 3(15)$
2	0	0	0	0	0	$12 = 0$
the	0	0	5	17	0	$\rightarrow k=2$

Price  $\leftarrow K \mid 5 \times 1$   
 For  $\leftarrow K \mid w_j \mid 5 \times 1$  w. ton  
 $\left\{ \begin{array}{l} \text{new } 1 \rightarrow K \\ \text{new } 0 \rightarrow 5 \times K \end{array} \right.$   
 $\underline{\underline{1012}} \rightarrow K$

$w_1$   
 $w_2$   
 $\frac{1}{2}$   
 $\frac{1}{2}$

## LANGUAGE MODELLING

Task:  $w_1, w_2, \dots, w_K$  predict  $w_{K+1}$

Hint: Probability

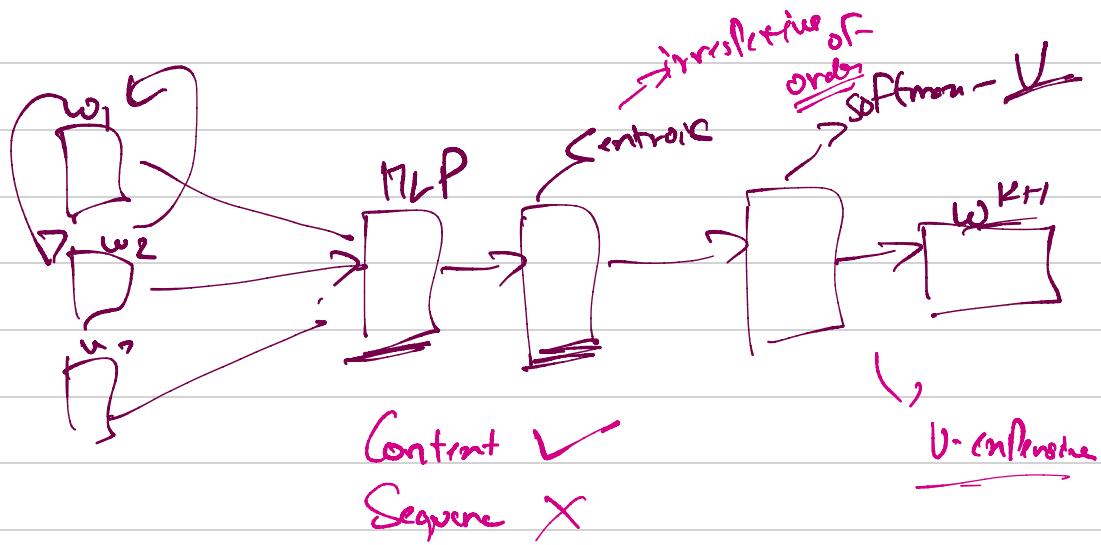
Ideas:

① Conditional Probability

$P(w_{K+1} \mid w_K, w_{K-1}, w_{K-2}, \dots)$   $\rightarrow$  Sequence is Previous

U-U-Fast  
+  
ChroD

① Context =  $P(w_{t+1} | w_t) \rightarrow$



## N-GRAMS

$P(w_{t+1} | w_t) \rightarrow$  bi-gram  
→ how many words  
am I using to  
create Probability model

g little to go then

$P(\text{little} | g) / P(\text{to} | \text{little}), P(\underline{\text{go}} / \text{to})$

$P(w_{12+1} | w_{12}, w_{12-1}) \rightarrow \underline{\text{trigrams}}$

①

②

③

$w(w_{12+1} | w_{12}, w_{12-1} \dots w_{12-n}) = \underline{n\text{-gram}}$

n words

$\rightarrow$  Considering Bi-gram

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(w_{12+1} | w_{12}) = \frac{P(w_{12+1} \cap w_{12})}{P(w_{12})}$$

$$= \frac{P(w_{12+1}, w_{12})}{P(w_{12})}$$

$$\underline{P(\omega_k)} = \frac{\text{Counts}}{\# \text{ words}}$$

$\omega_k, \omega_{k+1}$   
or  
 $\omega_{k+1}, \omega_k$

$$\underline{P(\omega_{k+1}, \omega_k)} = \frac{\# \text{ (}\omega_{k+1}, \omega_k\text{)}}{\text{total - count}}$$

$$P(\omega_{k+1} | \omega_k) = \frac{P(\omega_{k+1}, \omega_k)}{P(\omega_k)} = \frac{\# \text{ (}\omega_{k+1}, \omega_k\text{)}}{\# \text{ (}\omega_k\text{)}}$$

Seq is  
Preserves

$$P(g_0 | g) \Rightarrow \# \frac{\text{ (}g_0, g\text{)}}{\text{Total # of words}}$$

OR

$P(g_0, g)$   
# of bigrams wrt g was first word

# millions of now,

(g, go )  
, will )  
( , stat. /  
( : ) ]

highest cond. prob.

# words in V,  $\rightarrow$  multi

Redis online cache

• - 'g'  $\rightarrow$  'love'

DC1)

Google  $\downarrow$   
 $\rightarrow$  Item-gram  
 $\xrightarrow{\text{Problems}}$   
 $\rightarrow$  Not enough content  
 $\rightarrow$  Sparsity

(  $w_1, w_2, w_3, w_4, w_5, \dots$  )  $w_6$

~~51. x 10K CS~~ } 1012 One Problem

→ underflow ??

$$\log(a \cdot b) = \log a + \log b$$

$$\log(P(w_i)) \approx P(w_i \cdot r_i) - P(w_{ic}) / \alpha_i$$

↑-↑-↑

word in Text  
not Part  
of class

In Fraction

Smaller Value

Replace smoothing



MF  
+  
GMM

$$\text{test\_woc}' = \frac{o + \delta}{\text{Total no of woc}}$$

≠ o

## How to Evaluate

→ Bigram model ??  
Trigram model :-

I like Margherita pizza any day

Bigram →

$$p(i) * p(\text{like}/I) * p(\text{Margherita}/\text{like}) * p(\text{pizza}/\text{Margherita})$$

$$\cdot p(\text{any}/\text{pizza}) * p(\text{day}/\text{any})$$

→ Prob of

Prob → If entire sentence

in some seq- using

b-gram mod

my sentence  
to emit

some for Trigram

$$p(i) * (\text{like}) * p(\text{Margherita}/\text{like}, I) * p(\text{pizza}/\text{Margherita}, \text{like}) * \dots$$

Markovian property: Probability of next state depends only on my current state, basically bi-gram model.

# Perlmutter Confusion 2

I like this jug of water

$$n=2 \xrightarrow{\text{bigram}} P(\text{like}/\text{I}) \times P(\text{this}/\text{the}) \times P(\text{jug}/\text{a}) \dots$$

$$\frac{1}{0.0001} = 20??$$

(first-next-word)  $\rightarrow \underline{P_n}$

10

P<sub>n</sub> (trigram-model, y')

y  $\rightarrow$  so

y-go  $\rightarrow$  other

f go Hn → now

f go here now

The car hit the chicken while it was moving