N-gram Language Modeling

$$P(\bar{w}) = P(w_1, \cdots, w_m) = P(w_1)\,P(w_2|w_1)\,P(w_3|w_1 w_2) \cdots$$

n-gram LM:

$$P(\bar{w}) = \prod_{i=1}^{m} P(w_i \mid \underbrace{w_{i-n+1}, \cdots, w_{i-1}}_{\text{previous } n-1 \text{ words}})$$

$n \approx 3-7$

2-gram LM: $P(w_1|\langle s\rangle)\,P(w_2|w_1)\,P(w_3|w_2)\cdots$

w_3 cond. indep. of $w_1 \mid w_2$

n-gram LM $\Leftrightarrow$ n-1-order Markov model

3-gram LM: $P(w_1|\langle s\rangle\langle s\rangle)\,P(w_2|\langle s\rangle w_1)\,P(w_3|w_1 w_2)\cdots$

2-gram: multinomial distributions, $|V| \times |V|$ params

$P(w \mid the) =$ 0.001 house

0.0005 dog

0.0005 cat

⋮

very flat distribution!

Parameter estimation MLE from a large corpus

$$P(dog \mid the) = \frac{count(the, dog)}{count(the)}$$

Why ① Generation: machine translation

② Grammatical error correction

③ Way to build "word2vec++"