$\boxed{\text{N-gram Language Modeling}}$

- Placing distributions over sentences

$$P(\bar{w}) = P(w_1, w_2 \dots w_m) = P(w_1)\, P(w_2 \mid w_1)\, P(w_3 \mid w_2 w_1) \dots$$

- N-gram LM

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

- 2-gram LM: $P(w_1 \mid \text{<s>})\, P(w_2 \mid w_1)\, P(w_3 \mid w_2)$

  * w_3 independent of $w_1 \mid w_2$

* N-gram LM $\longleftrightarrow$ n-1 order Markov Model

- 3-gram LM: $P(w_1 \mid \text{<s><s>})\, P(w_2 \mid \text{<s>}\, w_1)\, P(w_3 \mid w_2 w_1) \dots$

- Multinomial Distributions (2-gram LM example):

V = size of vocab $\qquad |V| \times |V|$ parameters

$P(w \mid \text{"the"}) =$

0.001	House	
0.0005	Dog	Very Flat distribution
0.0005	Cat	

Parameter estimation: MLE from large corpus

MLE $\quad P(\text{dog} \mid \text{the}) = \dfrac{\text{count(the, dog)}}{\text{count(the)}}$

Does not tell you what words come up next. Only gives rough probability dist. of what words are likely to come next.

Why: (applications)
1. Generation: Machine translation
2. Grammatical Error Correction
3. Way to build "word2vec ++"

$\boxed{\text{Smoothing in N-gram LM's}}$

- 5-gram models work well!

(2) $P(w \mid \text{to})$ (3) $P(w \mid \text{go to})$ (5) $P(w \mid \text{hate to go to}) \rightarrow$ class

(5) $P(w \mid \text{want to go to}) \rightarrow$ Austin

$P(\text{Austin} \mid \text{to}) > 0 \rightarrow$ seen in data

$P(\text{Austin} \mid \text{want to go to}) = 0 \rightarrow$ if corpus isn't huge

* How do we give some portion of probability to these novel instances?

- Absolute Discounting : Reserve mass from seen 5-grams to allocate to unseen 5-grams

$$P_{AD}(\text{Austin} \mid \text{want to go to}) = \frac{\text{count}(\text{want to go to Austin}) - k}{\text{count}(\text{want to go to})} + \lambda \overbrace{P_{AD}(\text{Austin} \mid \text{to go to})}^{\text{4-gram}}$$

* λ set to make this normalized

Ex.

			$k=0.2$
want to go to (count = 4)	Maui	2 →	1.8
	Class	1 →	0.8
	Campus	1 →	0.8

$\lambda = \frac{0.6}{4}$ ← # word types seen in this context times k

* Can do this recursively! $P_{AD}(\text{Austin} \mid \text{to go to}) = \frac{\text{count}(\text{tgtA})}{\text{count}(\text{tgt})} + \lambda' \overbrace{P_{AD}(\text{Austin} \mid \text{go to})}^{\text{3-gram}}$

keep unrolling, and you eventually end up with $P(\text{Austin})$, which is > 0 is Austin is in corpus