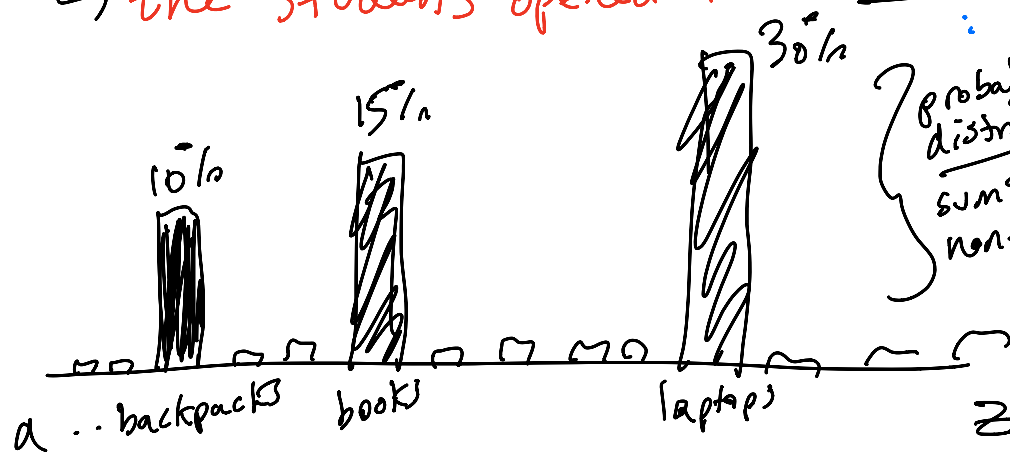
Language models

↳ given a prefix, a LM predicts the next word

↳ the students opened their ______
books
laptops
backpacks
⋮



probability distribution
sums to 1, non-negative

given $w_1 w_2 w_3 \ldots w_{n-1}$ we want to compute

$$p(w_n \mid w_{1 \ldots n-1})$$

↳ any model that computes this is called a language model

---

n-gram models:

p(backpacks | "the students opened their") we have a training dataset that's fairly large

let's extract all occurrences of the prefix

the students opened their books
the students opened their backpacks
the students opened their laptops
the students opened their laptops
the students opened their laptops
the students opened their backpacks

6 occurrences of this prefix

$$\left.\begin{aligned} p(\text{books} \mid \text{prefix}) &= \frac{1}{6} \\ p(\text{laptops} \mid \ldots) &= \frac{1}{2} \\ p(\text{backpacks} \mid \ldots) &= \frac{1}{3} \end{aligned}\right\}$$

maximum likelihood estimate, maximise the prob. of training dataset

problems?

↳ dataset size and diversity

↳ generalization

the students lazily opened their ____

↳ storage

prefix	w_1	w_2	w_3	...	w_{100k}
p_1			100	.	
p_2	0	0		.	
p_3	.				

; 0 0 5000

↳ Sparse
↳ table size is infinite

Storage issue can be addressed by truncating the prefix

$p(\text{backpacks} \mid \text{the students opened their})$
$\approx p(\text{backpacks} \mid \text{students opened their})$
$\approx p(\text{backpacks} \mid \text{opened their}) \Rightarrow$ trigram model
$\approx p(\text{backpacks} \mid \text{their}) \Rightarrow$ bigram
$\approx p(\text{backpacks}) \Rightarrow$ unigram

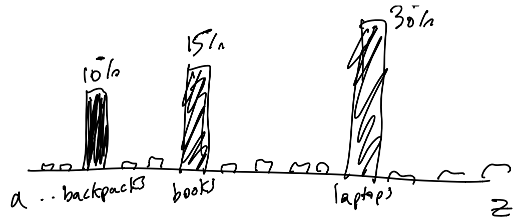
vocabulary contains V words
storage cost for n-gram model is V^n

---

decoding: how do we generate a multi-word response from a LM given a prefix?

evaluation: how do we know how good a given LM is?

the students opened their



↳ how do I generate the next word?

↳ just choose the word with the highest prob : greedy decoding

↳ randomly choose a word from vocab where prob of choosing w_i is proportional to $P_{LM}(w_i \mid w_{1 \ldots i-1})$: sampling

↳ temperature sampling

↳ changes the "peakiness" of distribution

↳ truncation sampling

↳ do not consider any words w_i where $p(w_i \mid prefix) < X$

30% 40% 70%

the students opened their laptops > and

2% navigated 80% to 40% Google . 30% <EOS>

evaluation : perplexity

let's say we estimated our LM

on a training dataset

now, we want to see how well

it does at estimating prob. of a test set

test set : $t_1 \, t_2 \, t_3 \, \dots \, t_n$

we want the LM to guess each word

in the test set without seeing it

we want :

$$P_{LM}(t_3 \mid t_1, t_2) \Rightarrow \text{to be high}$$

$$P_{LM}(t_4 \mid t_1, t_2, t_3) \Rightarrow \text{to be high}$$

$$\vdots$$

$$P_{LM}(t_i \mid t_{1 \dots, i-1}) \Rightarrow \text{to be high}$$

extremely important!

$$\text{test perplexity} = \exp\Big(\overbrace{-\frac{1}{n}\sum_{i}^{n} \log p_{LM}(t_i \mid t_{1,\dots,i-1})}^{\text{average negative log-likelihood}}\Big)$$

4. allows to interpret as "branching factor"

averaging over all words in test set

1. if LM is confident + correct, this prob is very high

if confident + wrong, prob is very low

2. log will be a tiny negative number

log will be a very large neg. number

perplexity (PPL):

- on average, how many equally likely words is the LM choosing between?
  - ↳ if low PPL: model is more certain
  - ↳ if high PPL: model is less certain

PPL of a model that gets 100%
prob. on the test set is 1

↳ is this achievable?

let's say you estimate a bigram
model on some training dataset.
you want to eval test ppl.
your test set contains a two-word
phrase that never occurred in training
dataset.

↳ what is your test PPL?