Q. Take a query and show how MLE approach works with a document as a LM.

→ Let's say I've a document d

d : "Statistics is the foundation of Machine Learning"

So, probability distribution of each word (ignoring stopwords)

Statistics →	1/4
foundation →	1/4
Machine →	1/4
Learning →	1/4

Now, I have a query q : "What is at the core of Machine Learning"

Our task is to rank different documents like above one based on one question: "Given document D, what is the probability it generated query q?"

Assuming each word is independent of each other,

$$P(q/D) = \cancel{\#} \prod_{i=1}^{n} P(w_i/D)$$

Let's remove stop words

q: "Core Machine Learing"

Now, since probabilities are independent, we take a product of them to get MLE

But if we don't have a particular query word in our document, its probability will be 0. This will negate the entire purpose of our document ranking - just because one word was missing.

$$\therefore P(q/d) = P(\text{core}/d)\, P(\text{Machine}/d)\, P(\text{Learning}/d)$$

$$0 \times 1/4 \times 1/4 = 0$$

To address MLE problem, we use smoothing/expansion methods that take some part of the known probabilities and assign it to unknown words.

2. Why would you want to expand a document model with a corpus model? How would you do that?

A document is a limited collection of words (100 words or even 40000 words). It is possible that certain documents may have a greater/lesser representation of particular words based on the topic each document is about. This is the very differentiating factor to rank them.

d1
d2
d3
⋮
d100 } Corpus

However, there may also be some missing words in our document. Now, easy way is to assign them a fixed constant value to prevent MLE from turning zero.

But if a document is about 'Antarctica', a missing words like 'North Pole' is more relevant than 'mars'. So, we need to have a mechanism where we can generalize the word probabilies of a document. Also, we want to reduce some impact of highly-repetitive words from our document.

So, we use probabilies of words from the corpus (expanding the document) to stabalize word frequencies of our document.

Let's say document d: "Statistics is the Foundation of Machine Learning"

Probability distribution of d

Statistics	1/4
Foundation	1/4
Machine	1/4
Learning	1/4
Core	0
iPhone	0

I also have a corpus, which is the probability distribution of all the document words together

Statistics	0.12
Foundation	0.13
Machine	0.10
Learning	0.09
Core	0.15
iPhone	0.03
⋮	0.14

Corpus probabilty distribution $(d_1+d_2+\dots d_n)$

So, I expand the model by taking weighted sum of probabilities

$$\boxed{P(t/d_{new}) = \lambda P(t/d) + (1-\lambda)\,P(t/corpus)}$$

Example, new 'more balanced' probabilities of

1. Statistics $\Rightarrow 0.6 \times 1/4 + 0.4 \times 0.12 = 0.17$
2. Core $\Rightarrow 0.6 \times 0 + 0.4 \times 0.15 = 0.06$ (instead of zero)

Once, done for all words, we use MLE to calculate $P(q/d)$.

This ensures non-zero probabilies for missing words, adding repetition relevancy to missing word probabilities and more balanced word probabilies within the document