

A horizontal bar with a yellow segment on the left and a red segment on the right.

Language Modeling

NLP II 2025

Assoc. Prof. Attapol Thamrongrattanarit

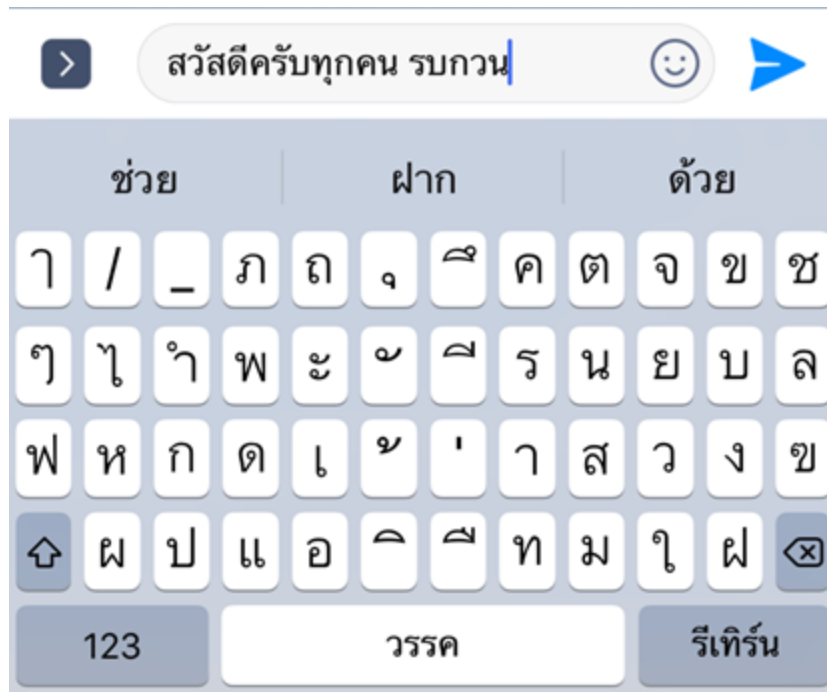


Guess the next word

- You will succeed if you don't give ____
- In the morning, office workers like to stop by my favorite café on their way to ____
- The BTS skytrain will take you from Siam to Ari in 10 ____

Predictive keyboard

The keyboards on smartphones and tablets are nearly unusable without this feature.



A horizontal bar with a gold segment on the left and a red segment on the right.

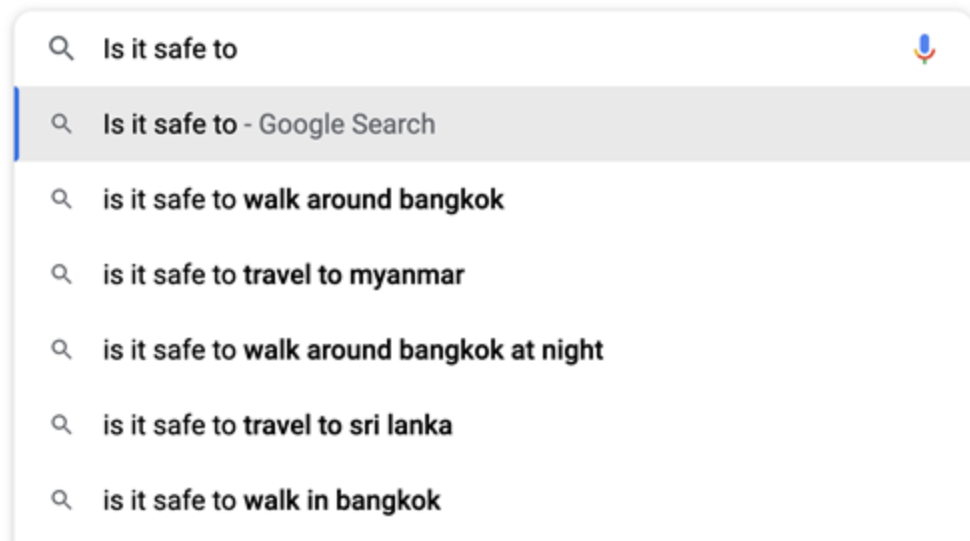
Write the rest of the sentence

- To prepare for the job interview, it is advisable to ____
- The bank account is overdrawn. In other words, you ____
- The official languages of Switzerland are ____



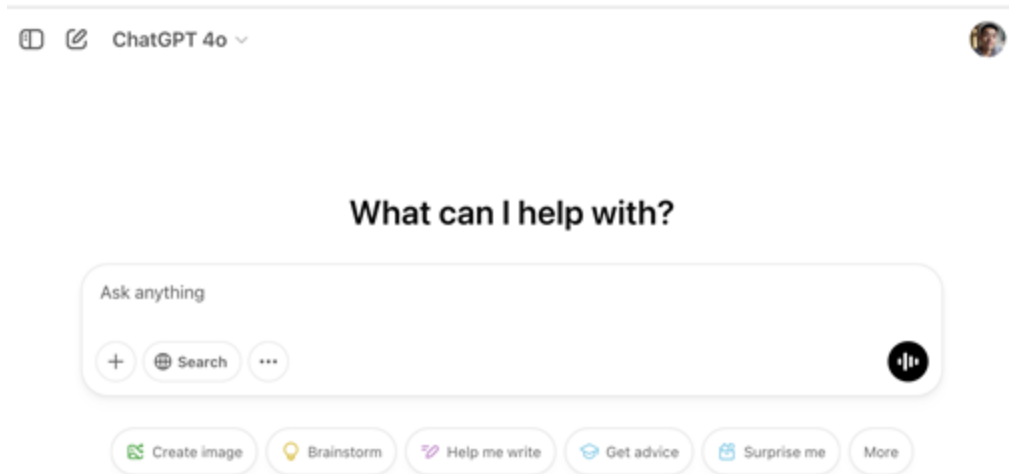
Autosuggest

People and languages are somewhat predictable



Chatbot with a (large) language model

ChatGPT presents a language model that is very good at filling the rest of the text. This is indeed the golden age of language model and generative AI.





Which one sounds 'more English'?

- The police officer told to the reporters about the incident.
The police officer told the reporters about the incident.
- อา นอน ตาก ลม
อา นอน ตา กลม
- I send him a letter.
I send dim a led her.
- Bangkok has many high buildings.
Bangkok has many tall buildings.

A horizontal bar with a gold segment on the left and a red segment on the right.

Language Model (LM)

A model that 'understands' a language (or languages). In computational linguistics, a language model must be able to:

- 1) Compute the probability of a text - fluent text should get higher prob.
- 2) Generate text from the context - we want fluency and 'good' content
- 3) Predict the next word from the context - we want accuracy

Applications of Language Models

- Fluent text generation requires contextual information
 - Chatbot
 - Machine translation
 - Automatic speech recognition (ASR or speech-to-text)
 - Summarization
 - Question answering
 - Grammatical error correction
- Predictive keyboard and typing suggestions

A horizontal bar with a gold segment on the left and a red segment on the right.

Types of Language Models

- n-gram language model uses normalized and smoothed counts of n-grams from a large dataset.
- Neural language model
 - Recurrent Neural Network LM (RNN-LM) uses embeddings instead of counts to encode the context
 - Transformer-based language model is a large neural language model (billions of parameters) trained on a massive dataset (Billions of tokens)



Probability



Probability distribution over vocabulary

- Probability measures how likely a word is to appear
- A vocabulary is a list of words that we want to consider
- A probability distribution $P(W=w)$ over a vocabulary is a function
 - Input: w one of the word in the vocabulary
 - Output: $P(w)$ probability of the word

Probability distribution over vocabulary (toy example)

w	กว้าง	กิน	ก่อน	ขาว	คน	ควร	จ่าย	จริง	ช่วย	ดี
P(W=w)	0.026	0.165	0.072	0.050	0.009	0.009	0.003	0.110	0.050	0.068
w	เด็ก	ได้	นี้	มาก	รัก	ร้าน	เร็ว	สวย	อ่าน	ใหญ่
P(W=w)	0.001	0.192	0.098	0.013	0.011	0.011	0.020	0.041	0.031	0.019

- $0 < P(W=w) < 1$ for any value of w
- The sum of all probabilities in a distribution must equal 1
- We can sample a word from the distribution. A word with a high probability is more likely to be picked.

A horizontal bar with a gold segment on the left and a red segment on the right.

Conditional Probability

- A conditional probability distribution is a probability distribution assuming that we know a new piece of knowledge (condition)
- $P(W \mid \text{context})$ is a conditional probability distribution over the vocabulary given the context on the left.
 - Context should determine what the probability distribution looks like (the next word is dependent on the context).

Probability of a string - calculating from left to right

$$\begin{aligned}
 P(\text{"โปรดมาเที่ยวที่เมืองของฉัน"}) &= P(\text{โปรด มา เที่ยว ที่ เมือง ของ ฉัน}) \\
 &= P(\text{โปรด} \mid \text{context} = \langle s \rangle) \\
 &\quad P(\text{มา} \mid \text{context} = \langle s \rangle \text{ โปรด}) \\
 &\quad P(\text{เที่ยว} \mid \text{context} = \langle s \rangle \text{ โปรด มา}) \\
 &\quad P(\text{ที่} \mid \text{context} = \langle s \rangle \text{ โปรด มา เที่ยว}) \\
 &\quad P(\text{เมือง} \mid \text{context} = \langle s \rangle \text{ โปรด มา เที่ยว ที่}) \\
 &\quad P(\text{ของ} \mid \text{context} = \langle s \rangle \text{ โปรด มา เที่ยว ที่ เมือง})
 \end{aligned}$$

n-gram Language Model



n-gram language model

- A unigram language model computes the probability of a sentence assuming that context does not matter.
- A bigram language model computes the probability of a sentence assuming that the only word to the left matters.
- A trigram model assumes that only two words to the left matter.
- 4-gram 5-gram 6-gram



Unigram Language Model

Example: we want to compute the probabilities of these two sentences

- Bangkok has many high buildings vs
Bangkok has many tall buildings

$$P(\text{Bangkok has many high buildings}) = \\ P(\text{Bangkok}) P(\text{has}) P(\text{many}) P(\text{high}) P(\text{buildings})$$

$$P(\text{Bangkok}) P(\text{has}) P(\text{many}) P(\text{tall}) P(\text{buildings}) = \\ P(\text{Bangkok}) P(\text{has}) P(\text{many}) P(\text{tall}) P(\text{buildings})$$

Estimating the unigram probability

$$P(\text{Bangkok}) = \frac{\text{Count}(\text{Bangkok})}{\text{Count of all unigrams}}$$

$$P(\text{has}) = \frac{\text{Count}(\text{has})}{\text{Count of all unigrams}}$$

unigram	count
...	
buildings	30
...	
...	
Bangkok	7
...	
...	
has	670
high	60
...	
...	
...	
tall	43
...	
...	
...	
...	
...	



Unigram Language Model fails here

$P(\text{Bangkok has many high buildings}) =$

$$\cancel{P(\text{Bangkok})} \cancel{P(\text{has})} \cancel{P(\text{many})} P(\text{high}) P(\text{buildings})$$

$P(\text{Bangkok has many tall buildings}) =$

$$\cancel{P(\text{Bangkok})} \cancel{P(\text{has})} \cancel{P(\text{many})} P(\text{tall}) \cancel{P(\text{buildings})}$$



Unigram Language Model

- $P(w_1, w_2, w_3, \dots, w_n) = P(w_1) P(w_2) P(w_3) \dots P(w_n)$
- $P(w)$ is estimated by counting the frequency of w normalized by the total counts of all unigrams.
- What would happen if we use unigram model to make a predictive keyboard i.e. predictive the next word given the context?
 - *You will succeed if you don't give ____*



Generating from unigram

- fifth an of futures the an incorporated a
- a the inflation most dollars quarter in is mass
- thrift did eighty said hard 'm july bullish
- that or limited the



Bigram model cares about the context

The context is provided by the previous word.

$P(\text{Bangkok has many high buildings}) =$

$P(\text{Bangkok} \mid \text{context} = \langle S \rangle)$

$P(\text{has} \mid \text{context} = \langle S \rangle \text{ Bangkok})$

$P(\text{many} \mid \text{context} = \langle S \rangle \text{ has})$

$P(\text{high} \mid \text{context} = \langle S \rangle \text{ many})$

$P(\text{buildings} \mid \text{context} = \langle S \rangle \text{ high})$

Estimating bigram probabilities

$$P(\textit{Bangkok} \mid \text{context} = \langle S \rangle) = \frac{\text{Count of bigram } (\langle S \rangle \textit{ Bangkok})}{\text{Count of bigram } (\langle S \rangle .)}$$

$$P(\textit{has} \mid \text{context} = \textit{Bangkok}) = \frac{\text{Count of bigram } (\textit{Bangkok has})}{\text{Count of bigram } (\textit{Bangkok} .)}$$

...

$$P(\textit{buildings} \mid \text{context} = \textit{high}) = \frac{\text{Count of bigram } (\textit{high buildings})}{\text{Count of bigram } (\textit{high} .)}$$

Augmenting the data (adding stuff to the data)

We need to add $\langle S \rangle$ and $\langle /S \rangle$ at the beginning and the end of each sentence so that the count of bigram $(X .)$ is equal to the count of unigram (X)

- Without augmentation:

I am Sam

Sam I am

The count of bigram $(\text{Sam} .) = 1$ but the count of unigram $(\text{Sam}) = 2$

- With augmentation

$\langle s \rangle$ I am Sam $\langle /s \rangle$

$\langle s \rangle$ Sam I am $\langle /s \rangle$

The count of bigram $(\text{Sam} .) = 2$ but the count of unigram $(\text{Sam}) = 2$

Estimating bigram probabilities

$$P(\textit{Bangkok} | \text{context} = \langle S \rangle) = \frac{\text{Count of bigram } (\langle S \rangle \textit{ Bangkok})}{\text{Count of unigram } (\langle S \rangle)}$$

of unigram ($\langle S \rangle$)

$$P(\textit{has} | \text{context} = \textit{Bangkok}) = \frac{\text{Count of bigram } (\textit{Bangkok has})}{\text{Count of unigram } (\textit{Bangkok})}$$

unigram ($\textit{Bangkok}$)

...

$$P(\textit{buildings} | \text{context} = \textit{high}) = \frac{\text{Count of bigram } (\textit{high buildings})}{\text{Count of unigram } (\textit{high})}$$

of unigram (\textit{high})

context	word	count
$\langle S \rangle$		
$\langle S \rangle$	Bangkok	10
$\langle S \rangle$		
...		
$\langle S \rangle$	$\langle /S \rangle$	
...		
Bangkok	has	4
...		
Bangkok	$\langle /S \rangle$	
...		
has	high	30
has		
...		
has	tall	50
has	$\langle /S \rangle$	50
...		
high	buildings	2
high	$\langle /S \rangle$	
...		

Count

A horizontal bar with a gold segment on the left and a red segment on the right.

What to count and compute

- A bigram model requires that the counts of
 - all bigrams
 - all unigrams
- A bigram model computes the probability of each word given the context of one word previous

$$P(w_1, w_2, w_3, \dots, w_n) = P(w_1 | <s>) P(w_2 | w_1) P(w_3 | w_2) \dots P(w_n | w_{n-1}) P(w_n | </s>)$$



Generating from a bigram model

- texaco rose one in this issue is pursuing growth in a boiler house said mr. gurria mexico 's motion control proposal without permission from five hundred fifty five yen
- outside new car parking lot of the agreement reached this would be a record november

Trigram and 4-gram LM

- Trigram Language Model

$$P(w_1, w_2, w_3, \dots, w_n) = P(w_1 | \text{START1}, \text{START2}) \\ P(w_2 | \text{START2}, w_1) \\ P(w_3 | w_1 w_2) \\ P(w_4 | w_2 w_3) \dots P(w_n | w_{n-2} w_{n-1})$$

- 4-gram Language Model

$$P(w_1, w_2, w_3, \dots, w_n) = P(w_1 | \text{START1}, \text{START2}, \text{START3}) \\ P(w_2 | \text{START2}, \text{START3}, w_1) \\ P(w_3 | \text{START3} w_1 w_2) \\ P(w_4 | w_1 w_2 w_3) \dots P(w_n | w_{n-3} w_{n-2} w_{n-1})$$



Generating from trigram LM

- people use television and film in a war will never duplicate the disruptions of power
- make sure that there have also just been praying for him to form a coalition government was grossly unfair



Generating from 4-gram LM

- make sure the economy continues to grow and to make reference to her poem called `` Living in Sin . '
- in order to save it from Austrian troops crushing the Hungarian rebellion
- On the other hand I 'm happy that I do believe there was racial discrimination in the Navy .

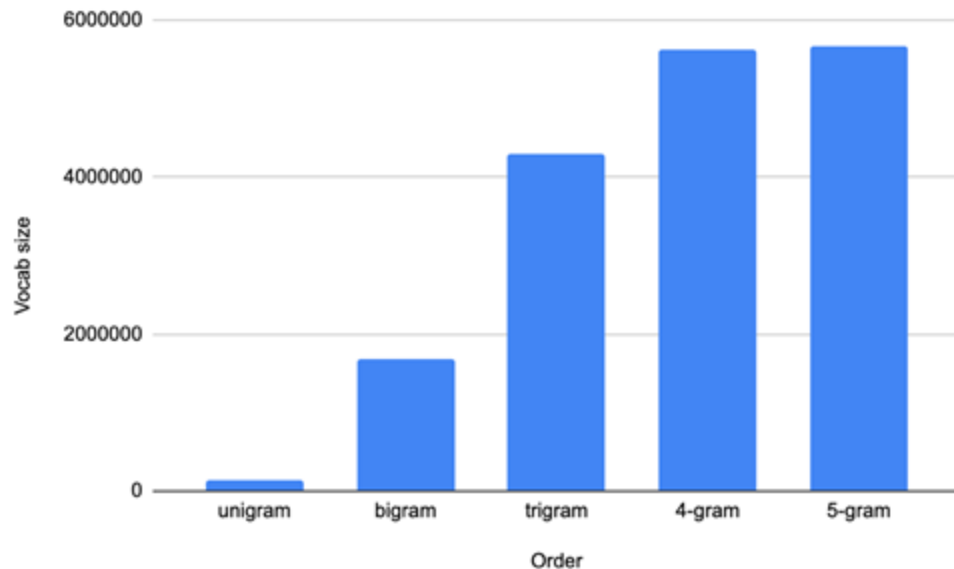


We need higher-order model

- Every language has long-distance dependencies
 - *The **computers** which I had just moved from the storage room in the basement to the lab on the second floor **were** fixed.*
- How many grams do we need?

Why don't we use 9-gram LM? or 10-gram?

5-gram LM trained on 8 million words needs to store 17,434,822 parameters





n-gram models rely on accurate counts

- If we use higher order model such as 7-gram or 8-gram, the context becomes more and more specific and most counts will be close to zero (more on this later). We can notice that the vocab size gets very large.
- More parameters require more data

Estimating n-gram probabilities

Estimate all bigram probabilities

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$$\begin{array}{lll} P(I | <s>) = \frac{2}{3} = .67 & P(\text{Sam} | <s>) = \frac{1}{3} = .33 & P(\text{am} | I) = \frac{2}{3} = .67 \\ P(</s> | \text{Sam}) = \frac{1}{2} = 0.5 & P(\text{Sam} | \text{am}) = \frac{1}{2} = .5 & P(\text{do} | I) = \frac{1}{3} = .33 \end{array}$$

Bigram counts

	I	want	to	eat	chinese	food	lunch	spend
I	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

	Unigram
I	2533
want	927
to	2417
eat	746
chinese	158
food	1093
lunch	341
spend	278

'Normalizing' by dividing by unigram counts

	I	want	to	eat	chinese	food	lunch	spend
I	0.002	0.326	0.000	0.004	0.000	0.000	0.000	0.001
want	0.002	0.000	0.656	0.001	0.006	0.006	0.005	0.001
to	0.001	0.000	0.002	0.284	0.001	0.000	0.002	0.087
eat	0.000	0.000	0.003	0.000	0.021	0.003	0.056	0.000
chinese	0.006	0.000	0.000	0.000	0.000	0.519	0.006	0.000
food	0.014	0.000	0.014	0.000	0.001	0.004	0.000	0.000
lunch	0.006	0.000	0.000	0.000	0.000	0.003	0.000	0.000
spend	0.004	0.000	0.004	0.000	0.000	0.000	0.000	0.000

$$P(i|<s>) = 0.25$$

$$P(\text{food}|\text{english}) = 0.5$$

$$P(\text{english}|\text{want}) = 0.0011$$

$$P(</s>|\text{food}) = 0.68$$

$$P(<s> \text{ i want english food } </s>)$$

$$= P(i|<s>)P(\text{want}|i)P(\text{english}|\text{want})$$

$$P(\text{food}|\text{english})P(</s>|\text{food})$$

$$= .25 \times .33 \times .0011 \times 0.5 \times 0.68$$

$$= .000031$$

Linguistic knowledge and world knowledge in LM

	I	want	to	eat	chinese	food	lunch	spend
I	0.002	0.326	0.000	0.004	0.000	0.000	0.000	0.001
want	0.002	0.000	0.656	0.001	0.006	0.006	0.005	0.001
to	0.001	0.000	0.002	0.284	0.001	0.000	0.002	0.087
eat	0.000	0.000	0.003	0.000	0.021	0.003	0.056	0.000
chinese	0.006	0.000	0.000	0.000	0.000	0.519	0.006	0.000
food	0.014	0.000	0.014	0.000	0.001	0.004	0.000	0.000
lunch	0.006	0.000	0.000	0.000	0.000	0.003	0.000	0.000
spend	0.004	0.000	0.004	0.000	0.000	0.000	0.000	0.000

$$P(i|<s>) = 0.25$$

$$P(\text{food}|\text{english}) = 0.5$$

$$P(\text{english}|\text{want}) = 0.0011$$

$$P(</s>|\text{food}) = 0.68$$



Practical issues

The product of small numbers creates too many zeros. We use the sum of log instead aka log probability of a sentence

$$\log(abc) = \log(a) + \log(b) + \log(c)$$

$$\begin{aligned} P(<s> \text{ i want english food } </s>) \\ &= P(\text{i} | <s>) P(\text{want} | \text{i}) P(\text{english} | \text{want}) \\ &\quad P(\text{food} | \text{english}) P(</s> | \text{food}) \\ &= .25 \times .33 \times .0011 \times 0.5 \times 0.68 \\ &= .000031 \end{aligned}$$

Evaluating language models

A horizontal bar with a gold segment on the left and a red segment on the right.

Evaluation methodology

- Extrinsic evaluation: evaluating on downstream tasks
 - Train the LMs on the same training set
 - Test the LMs on other tasks such as predictive keyboard, machine translation, speech recognition, grammatical error correction
- Intrinsic evaluation
 - Split the dataset into train, dev, test sets
 - Compute perplexity on dev and/or test set(s)



Extrinsic evaluation

- Machine translation - compute translation quality (BLEU score) when using a new LM to rank translations
- Speech recognition - compute word error rate when using a new LM to decode the speech into text
- Predictive keyboard - compute the accuracy and the keystroke discount when using a new LM to predict the next word as the users type on the keyboard

A horizontal bar with a yellow segment on the left and a red segment on the right.

Pros and cons of extrinsic evaluation

Pros

- Real-world relevance
- We can use many metrics from many tasks

Cons

- Expensive and time-consuming
- LMs that perform well in one task might not perform well in another.



Intrinsic Evaluation

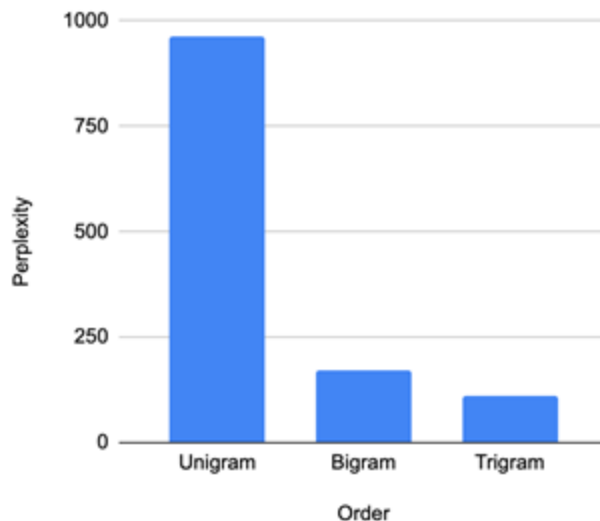
- Not relying on external tasks
- A better LM gives a higher probability to the words in the test set.
- We compute perplexity as a standard measure for LM performance. If a language model is good at predicting the next word, it is less perplexed (နှုတ်) when we reveal the next word.

Perplexity (lower = better)

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

Perplexity vs. Order





Intuition for perplexity

- When comparing two LMs, the better LMs will give a higher probability to the test set than the worse LMs.
- Perplexity is the inverse probability of the test set, so the lower the better.
- Perplexity is normalized by the number of words.

Language Models can compute
probabilities of sentences



N-Gram Language Modeling Tools

- KenLM: Kneser-Ney
- SRILM: Katz, Witten-Bell, Kneser-Ney



Grammar Check

- A good language models should output the higher probability to the sentence that is more 'English-y'
 - I look forward to meeting you
 - I look forward to meet you
- A grammatical English sentence should be more 'English-y'

A horizontal bar with a gold segment on the left and a red segment on the right.

English Grammatical Errors

- 28 types!
- <https://www.aclweb.org/anthology/W14-1701.pdf>

Verbs

Type	Description	Example
Vt	Verb tense	Medical technology during that time [is → was] not advanced enough to cure him.
Vm	Verb modal	Although the problem [would → may] not be serious, people [would → might] still be afraid.
V0	Missing verb	However, there are also a great number of people [who → who are] against this technology.
Vform	Verb form	A study in 2010 [shown → showed] that patients recover faster when surrounded by family members.
SVA	Subject-verb agreement	The benefits of disclosing genetic risk information [outweighs → outweigh] the costs.

Determiner and Nouns


ArtOrDet	Article or determiner	It is obvious to see that [internet → the internet] saves people time and also connects people globally.
Nn	Noun number	A carrier may consider not having any [child → children] after getting married.
Npos	Noun possessive	Someone should tell the [carriers → carrier's] relatives about the genetic problem.
Pform	Pronoun form	A couple should run a few tests to see if [their → they] have any genetic diseases beforehand.
Pref	Pronoun reference	It is everyone's duty to ensure that [he or she → they] undergo regular health checks.



Preposition

*This essay will [**discuss about** → discuss] whether a carrier should tell his relatives or not.*

Wci	Wrong collocation/idiom	Early examination is [healthy → advisable] and will cast away unwanted doubts.
Wa	Acronyms	After [WOWII → World War II], the population of China decreased rapidly.
Wform	Word form	The sense of [guilty → guilt] can be more than expected.
Wtone	Tone (formal/informal)	[It's → It is] our family and relatives that bring us up.
Srun	Run-on sentences, comma splices	The issue is highly [debatable, a → debatable. A] genetic risk could come from either side of the family.
Smod	Dangling modifiers	[Undeniable, → It is undeniable that] it becomes addictive when we spend more time socializing virtually.
Spar	Parallelism	We must pay attention to this information and [assisting → assist] those who are at risk.
Sfrag	Sentence fragment	However, from the ethical point of view.
Ssub	Subordinate clause	This is an issue [needs → that needs] to be addressed.
WOinc	Incorrect word order	[Someone having what kind of disease → What kind of disease someone has] is a matter of their own privacy.
WOadv	Incorrect adjective/adverb order	In conclusion, [personally I → I personally] feel that it is important to tell one's family members.
Trans	Linking words/phrases	It is sometimes hard to find [out → out if] one has this disease.
Mec	Spelling, punctuation, capitalization, etc.	This knowledge [maybe relavant → may be relevant] to them.
Rloc—	Redundancy	It is up to the [patient's own choice → patient] to disclose information.
Cit	Citation	Poor citation practice.
Others	Other errors	An error that does not fit into any other category but can still be corrected.
Um	Unclear meaning	Genetic disease has a close relationship with the born gene . (i.e., no correction possible without further clarification.)

A horizontal bar with a yellow segment on the left and a red segment on the right.

Most common types of errors

Error type	Frequency	Recall
Article	14.8	54.74
Word collocation or idiom	11.8	15.18
Redundancy	10.5	26.47
Verb tense	7.11	20.00
Word form	4.8	46.59

5-gram LM

This essay will discuss about the current issues of climate change

= ... x P(issues | discuss about the current) x

P(of | about the current issues) x

P(climate | the current issues of) x P(change | current issues of climate)

This essay will discuss the current issues of climate change

= ... x P(issues | will discuss the current) x

P(of | discuss the current issues) x

P(climate | the current issues of) x P(change | current issues of climate)

A horizontal bar with a gold segment on the left and a red segment on the right.

Your Turn

- Use language models to detect the following errors:
 - Verb Tense
 - Verb Agreement
 - Preposition
 - Word form such as *The circuit suffers from many electrically problems.*
 - Article such as *dog is chasing cat*

A good language model can generate good text given a context

Sample from a
probability distribution
over the vocab

$P(w \mid \text{'a good robot must'}) =$

trigram model: 'robot must *'

4-gram model: 'good robot must *'

0	a
0	and
0	but
0.07	ethically
0.41	follow
0.36	obey
0.02	orders
...	...
0.08	rules
0.06	set
0	strictly
0	the
0	without

Sample from a
probability distribution
over the vocab

$P(w \mid \text{'a good robot must follow'}) =$

trigram model: 'must follow *'

4-gram model: 'robot must follow *'

0.23	a
0	and
0.01	but
0.02	ethically
0	follow
0.02	obey
0.28	orders
...	...
0.19	rules
0	set
0	strictly
0.25	the
0	without

Sample from a
probability distribution
over the vocab

$P(w \mid \text{'a good robot must follow rules'}) =$

trigram model: 'follow rules *'

4-gram model: 'must follow rules*'

0	a
0.03	and
0.02	but
0.31	ethically
0	follow
0	obey
0	orders
...	...
0	rules
0.27	set
0.37	strictly
0	the
0	without

1
gram

–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have

–Hill he late speaks; or! a more to leg less first you enter

2
gram

–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.

–What means, sir. I confess she? then all sorts, he is trim, captain.

3
gram

–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

–This shall forbid it should be branded, if renown made it empty.

4
gram

–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;

–It cannot be but so.

Generating from n-gram models trained on Shakespeare's works

1
gram

Months the my and issue of year foreign new exchange's september
were recession exchange new endorsed a acquire to six executives

2
gram

Last December through the way to preserve the Hudson corporation N.
B. E. C. Taylor would seem to complete the major central planners one
point five percent of U. S. E. has already old M. X. corporation of living
on information such as more frequently fishing to keep her

3
gram

They also point to ninety nine point six billion dollars from two hundred
four oh six three percent of the rates of interest stores as Mexico and
Brazil on market conditions

Generating from n-gram models trained on newspaper (Wall Street Journal)



Limitations for n-gram models

- n-gram language models only have the information on the strings that are already seen. The generated text stay within the training set.
- Languages are infinite. We are guaranteed to find an ngram that we have not seen before. Example:
 - นักท่องเที่ยว ขุด พบ เรือดำน้ำ
 - หมา พุดเตีล หาว ตั้ง



Consider the trigram
model: *denied the* —

We collect the trigrams:

denied the allegations

5

denied the speculation

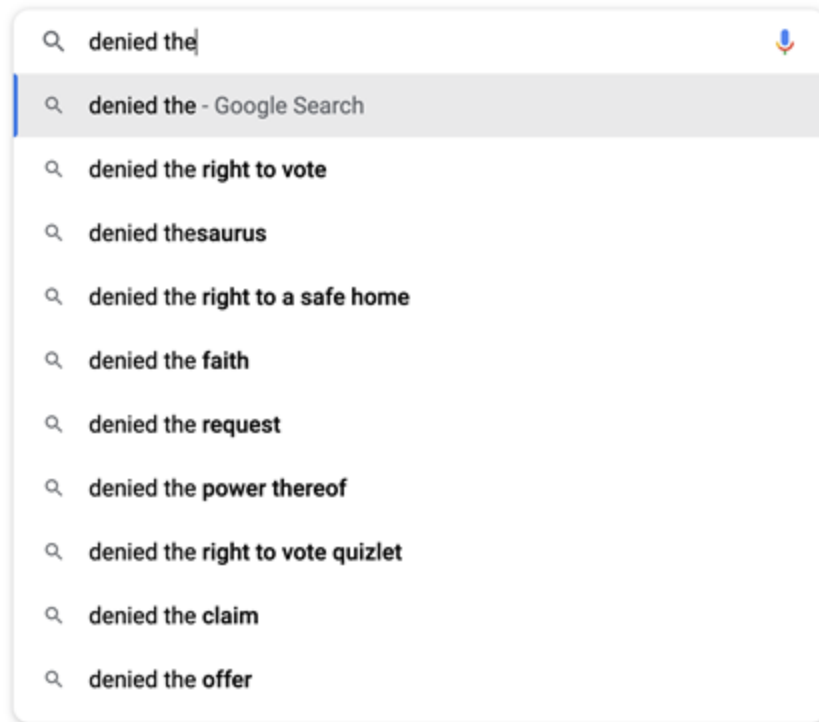
2

denied the rumors

1

denied the report

1



Problems

if this is zero, then probability is zero.
we must smooth the count

$$P(\text{right} \mid \text{officials deny the}) = \frac{\text{count}(\text{officials deny the right})}{\text{count}(\text{officials deny the})}$$

we have to store
millions of counts

if this zero, then divide by 0 = Infinity!
we must backoff to the lower order
model such as 4gram → trigram

A horizontal bar with a gold segment on the left and a red segment on the right.

Good n-gram LM use all of these

- Katz LM uses back-off techniques.
- Witten-Bell uses linear interpolation (and also back-off in a way)
- Kneser-Ney uses both back-off and interpolation.



Generalization and zeros

- N-gram models struggle with choosing how much context we should consider
 - Too much context and the model performs poorly because we do not have enough data. (sparsity problem)
 - Too little context and the model performs poorly because of long-range dependencies in language.
- N-gram models are bound by the n-grams found in the training set only.
- N-gram models are still useful for modeling simpler sequences such as syllables in a word (syllable n-gram), characters in a name (character n-gram), etc.