Sampling. Modele N-gramowe

Paweł Rychlikowski

Instytut Informatyki UWr

29 października 2025

Modele N-gramowe. Przypomnienie

Definicja

N-gramem nazywamy ciąg kolejnych słów o długości N. 1-gramy to unigramy, 2-gramy to bigramy, 3-gramy to trigramy.

Za pomocą N-gramów tworzymy model języka, w którym staramy się przewidzieć kolejne słowo (N-te) na podstawie $N-1$ słów poprzednich.

n-gram Language Models: Example

Suppose we are learning a 4-gram Language Model.

~~as the proctor started the clock, the~~ *students opened their* _____

discard

condition on this

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

For example, suppose that in the corpus:

- "students opened their" occurred 1000 times
- "students opened their books" occurred 400 times
  - → P(books | students opened their) = 0.4
- "students opened their exams" occurred 100 times
  - → P(exams | students opened their) = 0.1

Should we have discarded the "proctor" context?

25

Sparsity Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their w"* never occurred in data? Then *w* has probability 0!

(Partial) Solution: Add small δ to the count for every $w \in V$. This is called *smoothing*.

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

Sparsity Problem 2

Problem: What if *"students opened their"* never occurred in data? Then we can't calculate probability for *any w*!

(Partial) Solution: Just condition on *"opened their"* instead. This is called *backoff*.

Note: Increasing *n* makes sparsity problems *worse*. Typically, we can't have *n* bigger than 5.

Sparsity Problems with n-gram Language Models

Sparsity Problem 1

Problem: What if *"students opened their w"* never occurred in data? Then *w* has probability 0!

(Partial) Solution: Add small δ to the count for every $w \in V$. This is called *smoothing*.

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

Sparsity Problem 2

Problem: What if *"students opened their"* never occurred in data? Then we can't calculate probability for *any w*!

(Partial) Solution: Just condition on *"opened their"* instead. This is called *backoff*.

Note: Increasing *n* makes sparsity problems *worse*. Typically, we can't have *n* bigger than 5.

Storage Problems with n-gram Language Models

Storage: Need to store count for all *n*-grams you saw in the corpus.

$$P(\boldsymbol{w}|\text{students opened their}) = \frac{\text{count}(\text{students opened their } \boldsymbol{w})}{\text{count}(\text{students opened their})}$$

Increasing *n* or increasing corpus increases model size!

27

n-gram Language Models in practice

- You can build a simple trigram Language Model over a 1.7 million word corpus (Reuters) in a few seconds on your laptop*

Business and financial news

today the _____

get probability distribution

company	0.153
bank	0.153
price	0.077
italian	0.039
emirate	0.039
…	

Sparsity problem: not much granularity in the probability distribution

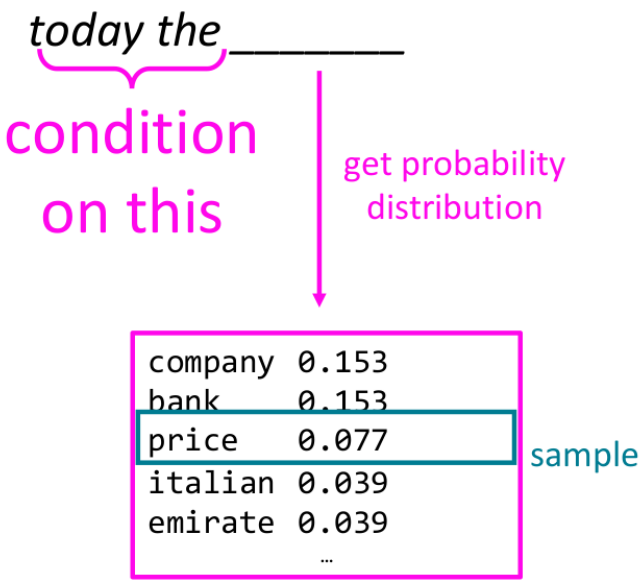
Otherwise, seems reasonable!

*** Try for yourself:** https://nlpforhackers.io/language-models/

28

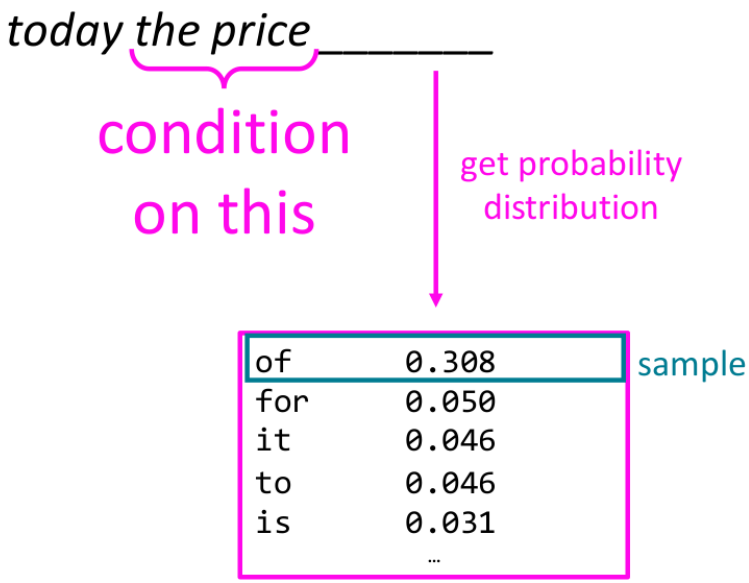
Generating text with a n-gram Language Model

You can also use a Language Model to generate text



Generating text with a n-gram Language Model

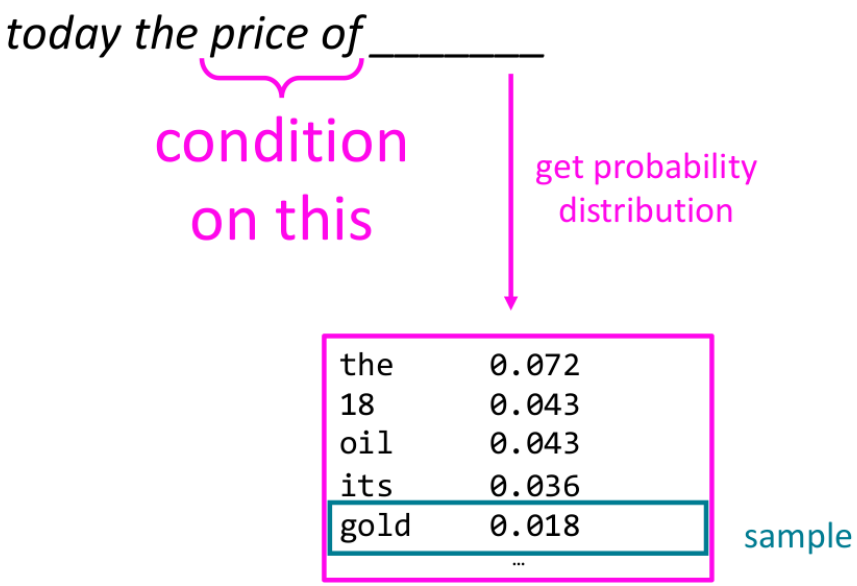
You can also use a Language Model to generate text



30

Generating text with a n-gram Language Model

You can also use a Language Model to generate text



31

Generating text with a n-gram Language Model

You can also use a Language Model to generate text

today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

Surprisingly grammatical!

…but **incoherent.** We need to consider more than three words at a time if we want to model language well.

But increasing *n* worsens sparsity problem, and increases model size…

32

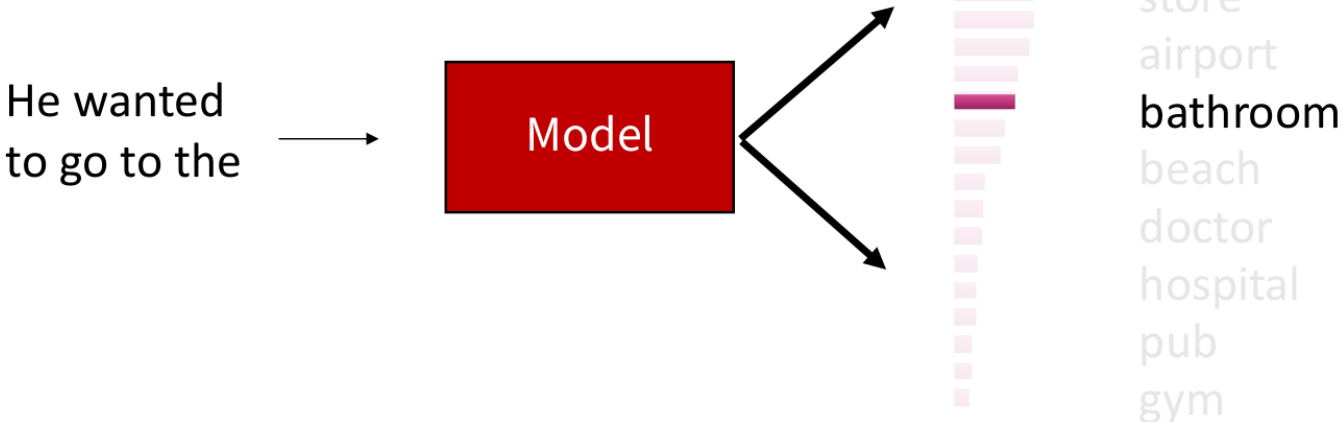
- Popatrzmy na generację w modelu Papuga
- (będziemy pokazywać 10 najbardziej prawdopodobnych opcji)

Time to get random : Sampling!

- Sample a token from the distribution of tokens

$$\hat{y}_t \sim P(y_t = w \mid \{y\}_{<t})$$

- It's *random* so you can sample any token!



30

Decoding: Top-*k* sampling

- Problem: Vanilla sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass (statistics speak: we have "heavy tailed" distributions)
  - Many tokens are probably *really wrong* in the current context
  - Why are we giving them *individually* a tiny chance to be selected?
  - Why are we giving them *as a group* a high chance to be selected?

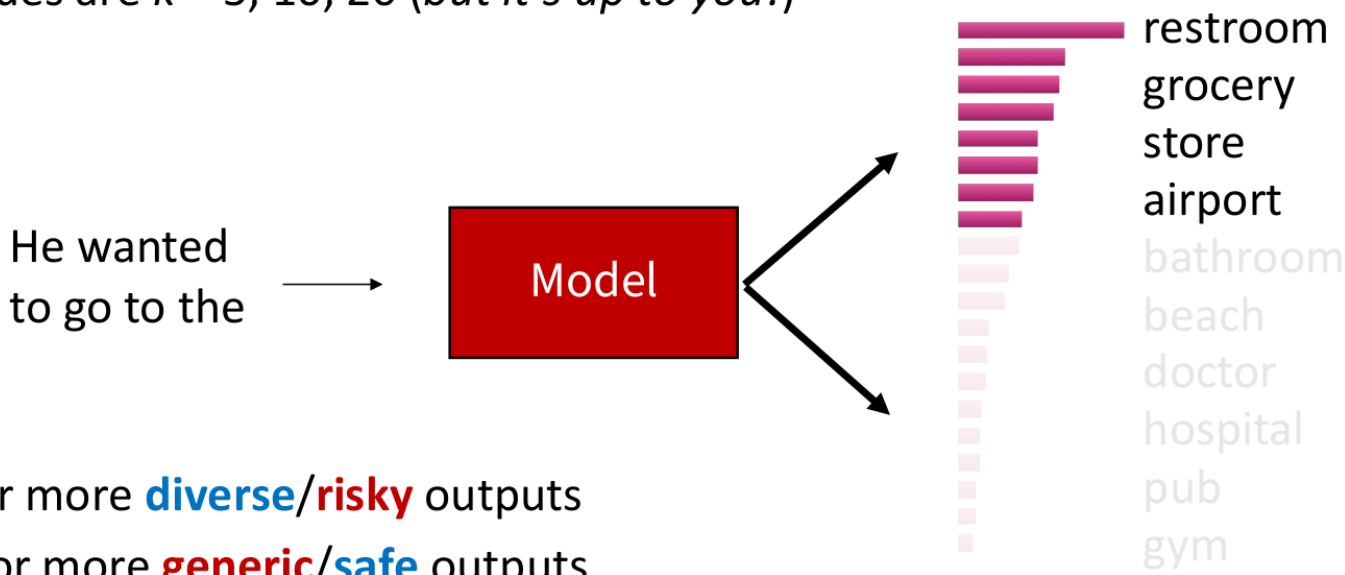
- Solution: Top-*k* sampling
  - Only sample from the top *k* tokens in the probability distribution

31

(Fan et al., ACL 2018; Holtzman et al., ACL 2018)

Decoding: Top-*k* sampling

- Solution: Top-*k* sampling
  - Only sample from the top *k* tokens in the probability distribution
  - Common values are *k* = 5, 10, 20 (*but it's up to you!*)

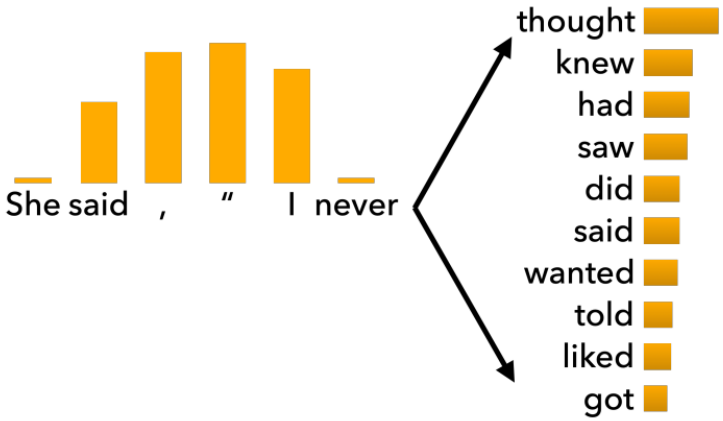


  - Increase *k* for more **diverse**/**risky** outputs
  - Decrease *k* for more **generic**/**safe** outputs

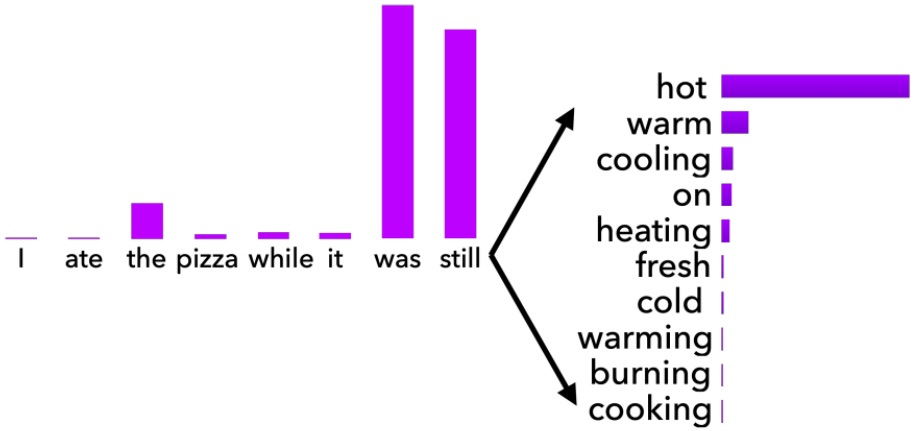
32

(Fan et al., ACL 2018; Holtzman et al., ACL 2018)

Issues with Top-*k* sampling



Top-*k* sampling can cut off too ***quickly***!



Top-*k* sampling can also cut off too ***slowly***!

(Holtzman et. al., ICLR 2020)

Decoding: Top-*p* (nucleus) sampling

- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited *k* removes many viable options
  - When the distribution P_t is peakier, a high *k* allows for too many options to have a chance of being selected

- Solution: Top-*p* sampling
  - Sample from all tokens in the top *p* cumulative probability mass (i.e., where mass is concentrated)
  - Varies *k* depending on the uniformity of P_t

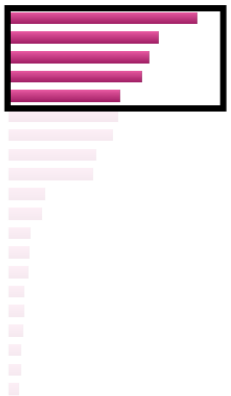
34

(Holtzman et. al., ICLR 2020)

Decoding: Top-*p* (nucleus) sampling

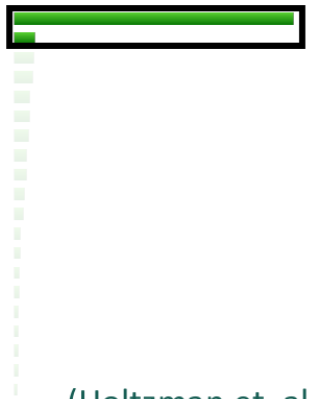
- Solution: Top-*p* sampling
  - Sample from all tokens in the top *p* cumulative probability mass (i.e., where mass is concentrated)
  - Varies *k* depending on the uniformity of P_t

$P_t^1(y_t = w \mid \{y\}_{<t})$



$P_t^2(y_t = w \mid \{y\}_{<t})$

$P_t^3(y_t = w \mid \{y\}_{<t})$



(Holtzman et. al., ICLR 2020)

Scaling randomness: Softmax temperature

- Recall: On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$

$$P_t(y_t = w) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t:

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

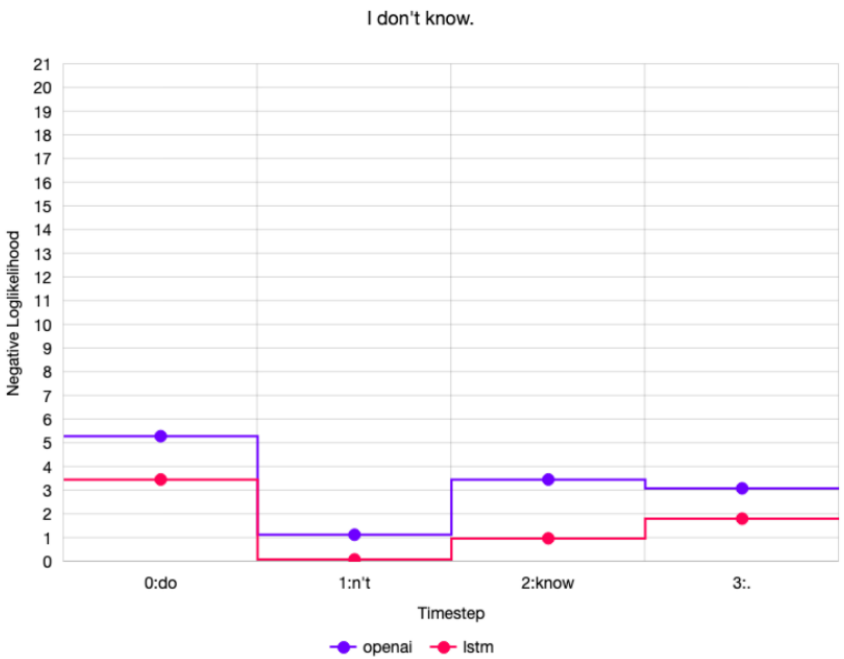
- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - **More** diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
  - **Less** diverse output (probability is concentrated on top words)

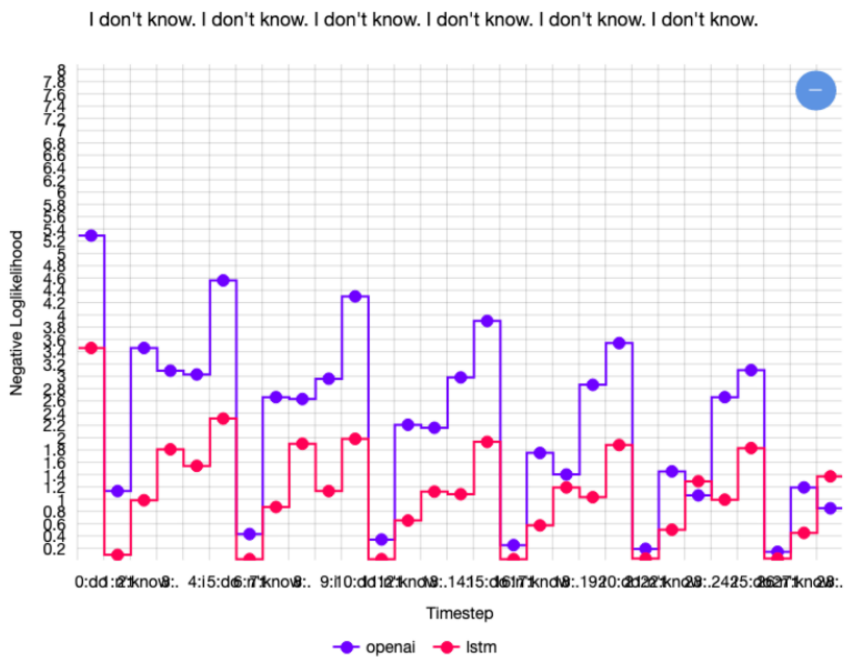
Note: softmax temperature is not a decoding algorithm!
It's a technique you can apply at test time, in conjunction with a decoding algorithm (such as beam search or sampling)

Powtarzalność

- Problemem w generacji jest powtarzalność (to znaczy, że model generuje powtarzające się ciągi)
- Spróbujmy zaobserwować ten fenomen w przypadku papugi.

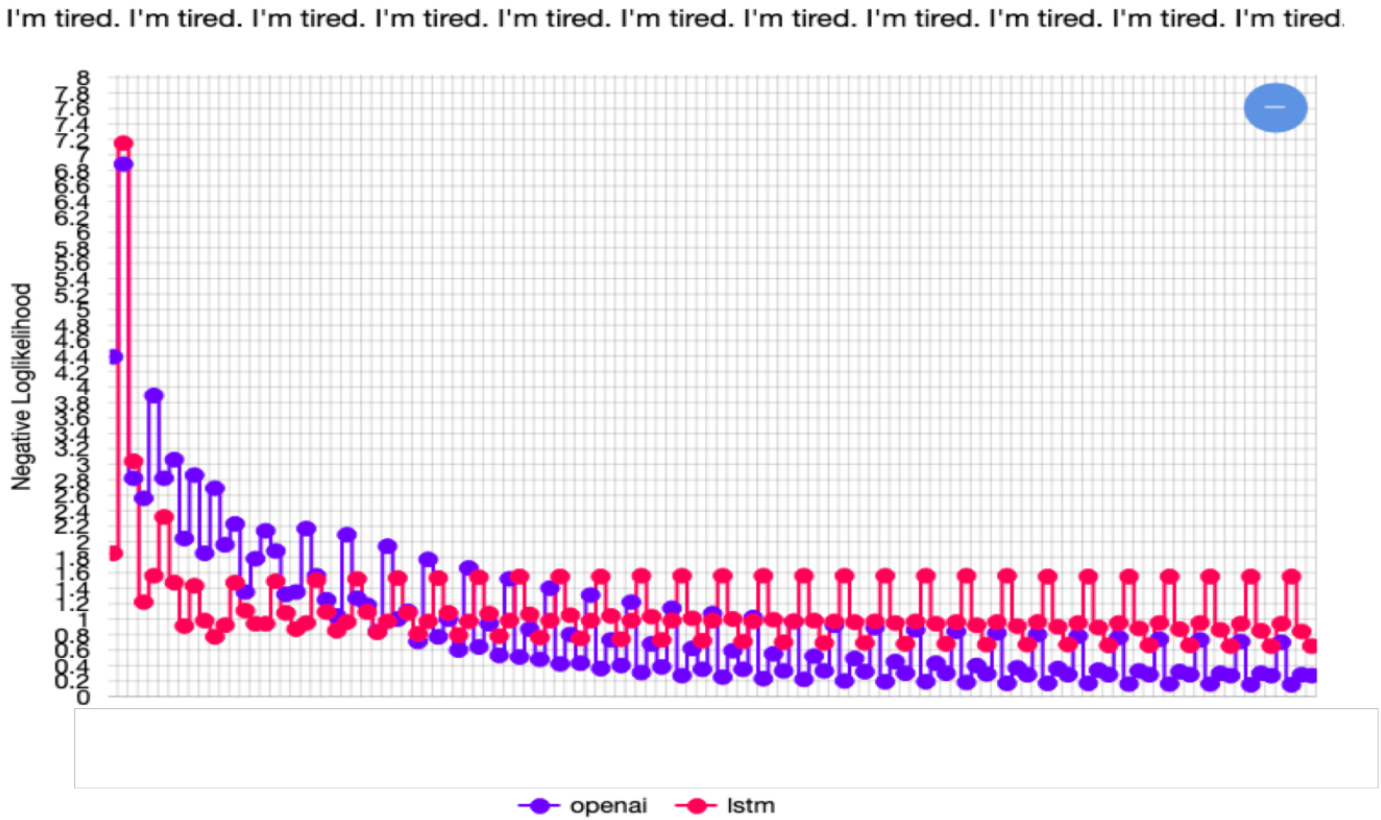
Why does repetition happen?





(Holtzman et. al., ICLR 2020)

And it keeps going...



27

(Holtzman et. al., ICLR 2020)

Ocena modeli językowych

Są generalnie dwa sposoby oceniania modeli językowych (i, tak naprawdę, wszystkiego innego też):

1. Wewnętrzna (instristic) – mamy jakąś mniej lub bardziej naturalną miarę jakości modelu
2. Zewnętrzna (exstrinsic) – sprawdzamy, jak model poradzi sobie z pewnym zadaniem (które jest naszym celem, i w którym mamy naturalną miarę jakości)

Z miarą zewnętrzną spotykamy się od pierwszej pracowni.

Perplexity (miara nieokreśloności)

Intuicje

1. To co się zdaża, powinno mieć wysokie prawdopodobieństwo.
2. Gdy dobrze przewidujemy kolejne słowo (na podstawie pełnego prefiksu), to jesteśmy w stanie dobrze kompresować tekst (dlaczego?)
3. Oczywiście powinniśmy dzielić korpus (na część przeszłą (zdarzyła się) i przyszłą (zdarzy się, chcemy jej dać spore prawdopodobieństwo, ale jej nie znamy)

Perplexity (2)

Wzór

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

gdzie N jest wielkością części testowej korpusu

Pytanie

Jakie jest perplexity całkiem losowego ciągu cyfr?(odpowiedź: **10**)

Można rozumieć perplexity jako średni ważony **współczynnik rozgałęzienia** języka.