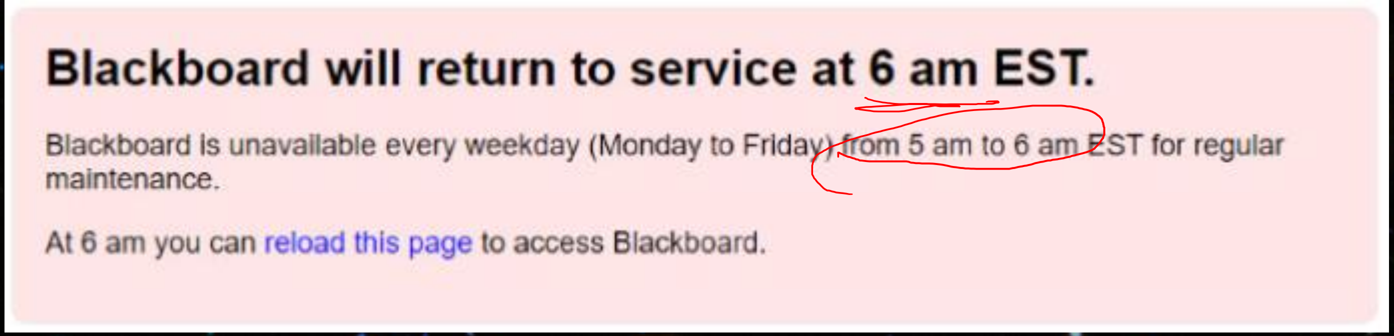
Proposal Marks are Out!

Assign 1, Individual, Tomorrow 7 AM EST



Blackboard will return to service at 6 am EST.

Blackboard is unavailable every weekday (Monday to Friday) from 5 am to 6 am EST for regular maintenance.

At 6 am you can reload this page to access Blackboard.

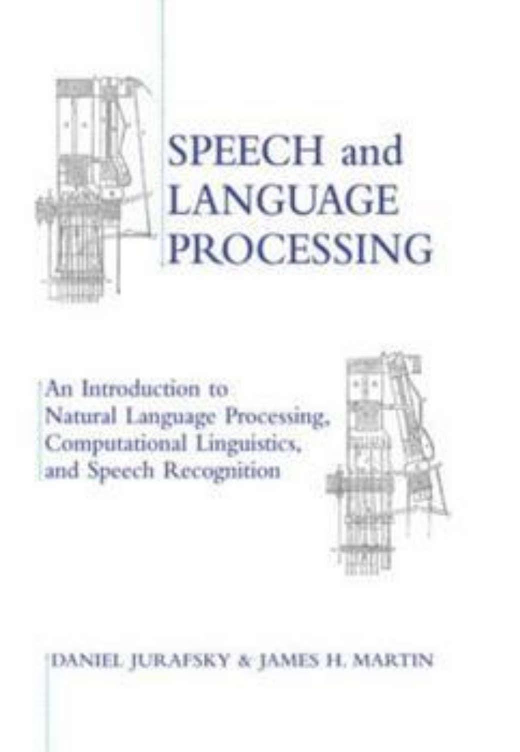
Office Hour

2:30 PM – 3:30 PM

Lambton Tower 5111

Memento, Christopher Nolan
Guy Pearce, Carrie-Anne Moss, Joe Pantoliano
September 2000
Budget $4.5 m
Box office $40 m

n-Gram Language Models



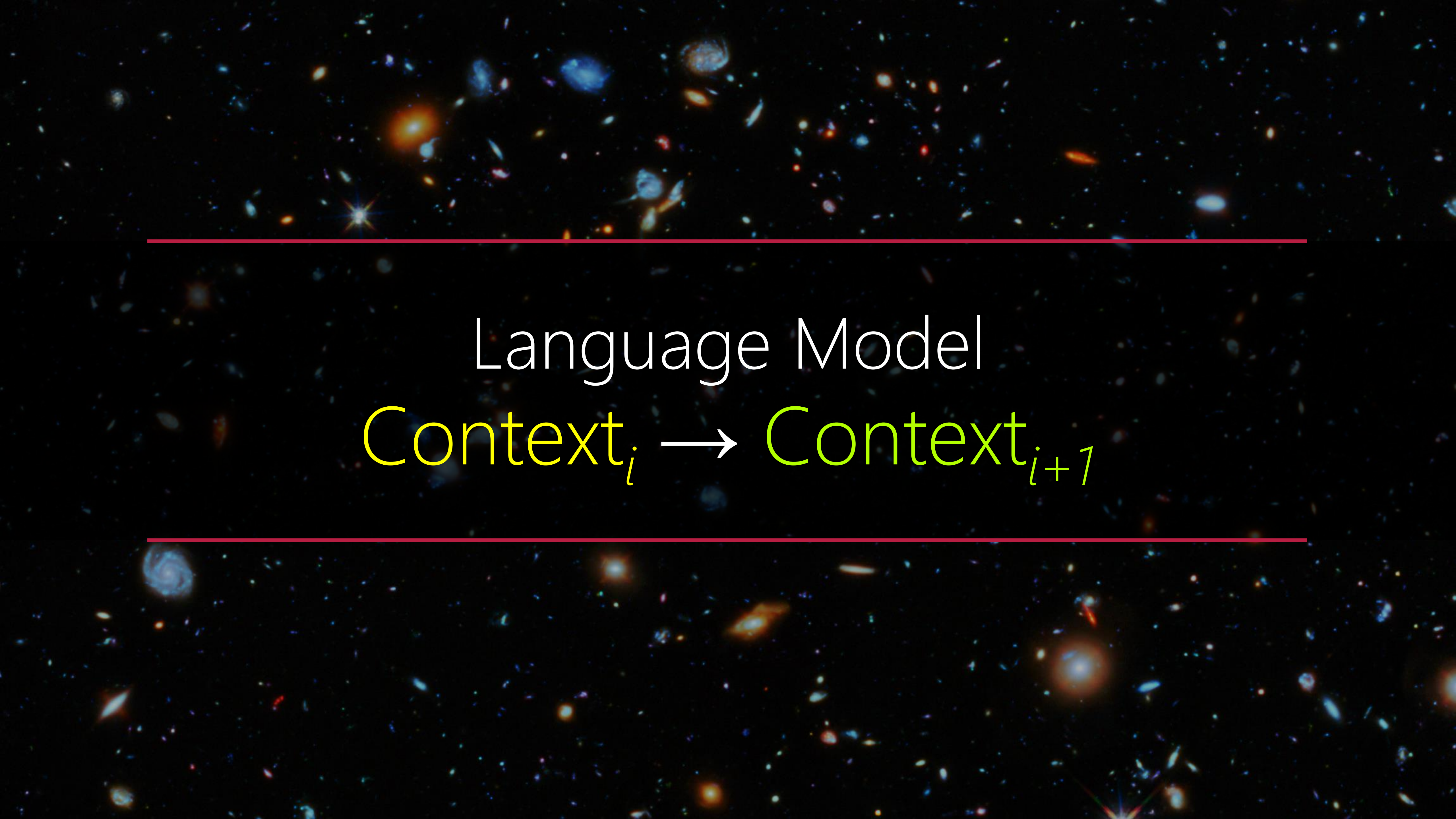
SPEECH and LANGUAGE PROCESSING

An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition

DANIEL JURAFSKY & JAMES H. MARTIN

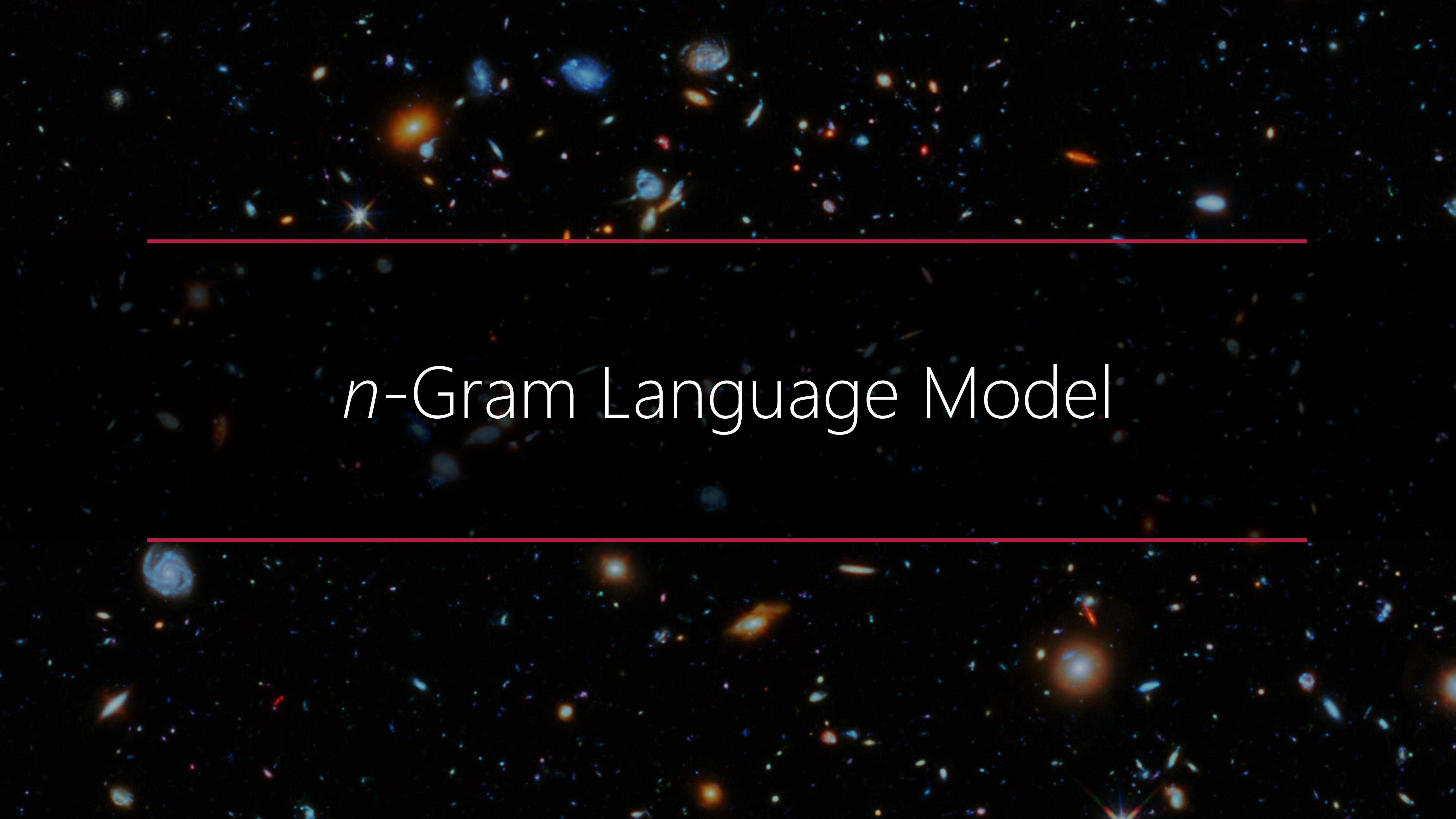
Language Modeling

CH04

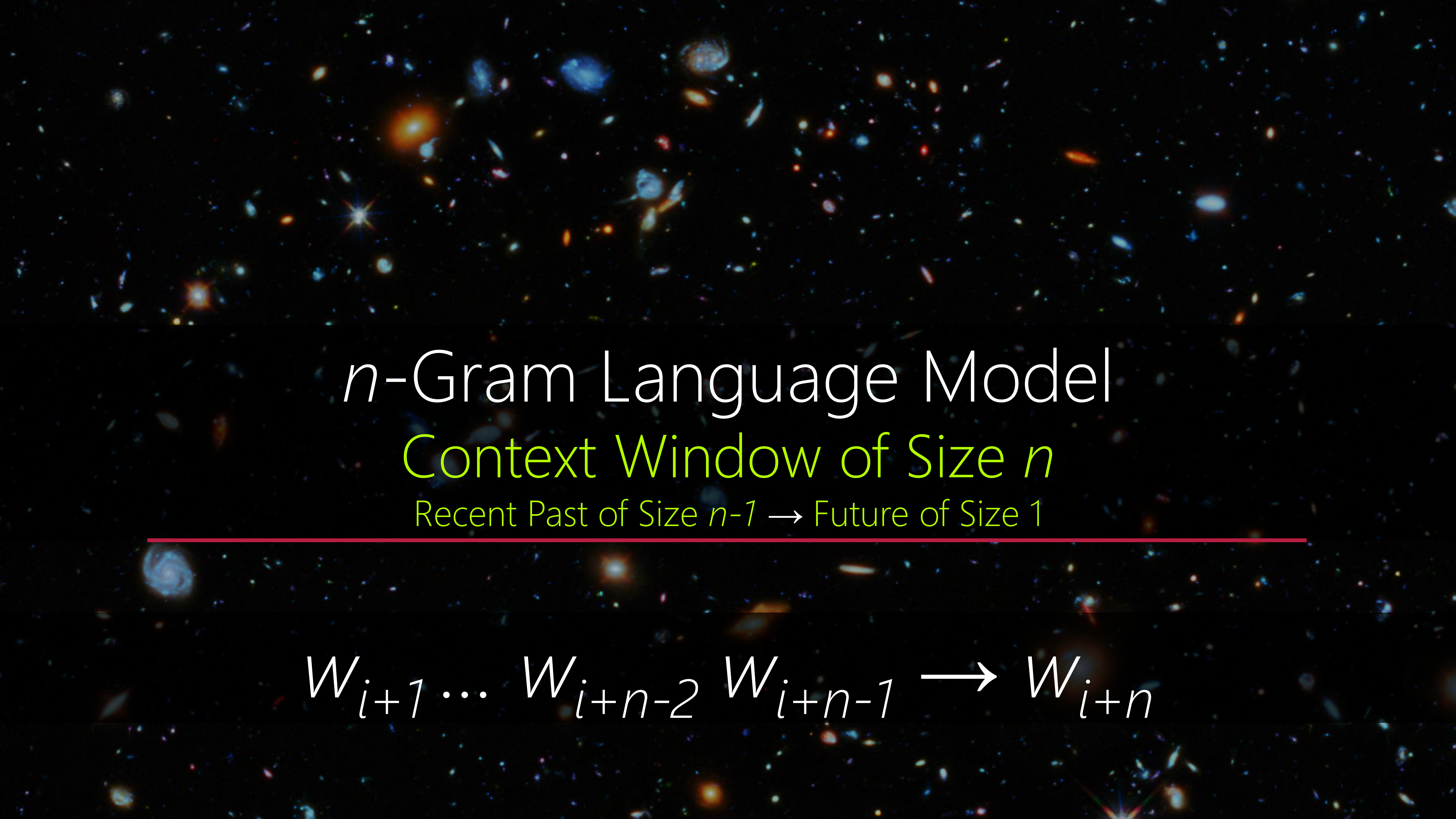


Language Model

$$\text{Context}_i \rightarrow \text{Context}_{i+1}$$



n-Gram Language Model



n-Gram Language Model

Context Window of Size *n*

Recent Past of Size *n-1* → Future of Size 1

$$w_{i+1} \cdots w_{i+n-2} \; w_{i+n-1} \longrightarrow w_{i+n}$$

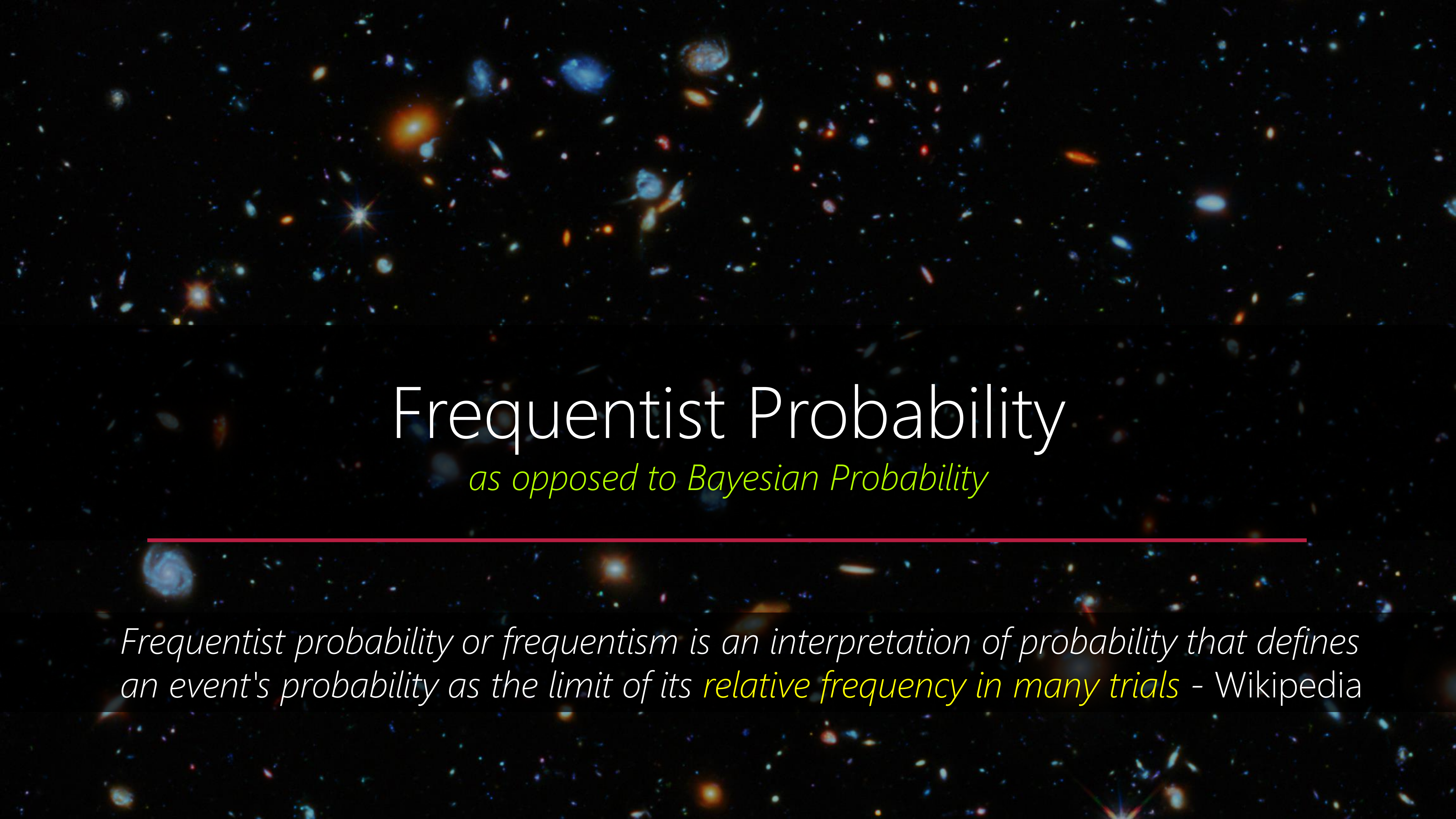
n-Gram Language Model

$\rightarrow W_{i+1}$ 1-gram = unigram

$W_{i+1} \rightarrow W_{i+2}$ 2-gram = bigram

$W_{i+1}\ W_{i+2} \rightarrow W_{i+3}$ 3-gram = trigram

$W_{i+1} \ldots\ W_{i+n-2}\ W_{i+n-1} \rightarrow W_{i+n}$ n-gram



Frequentist Probability

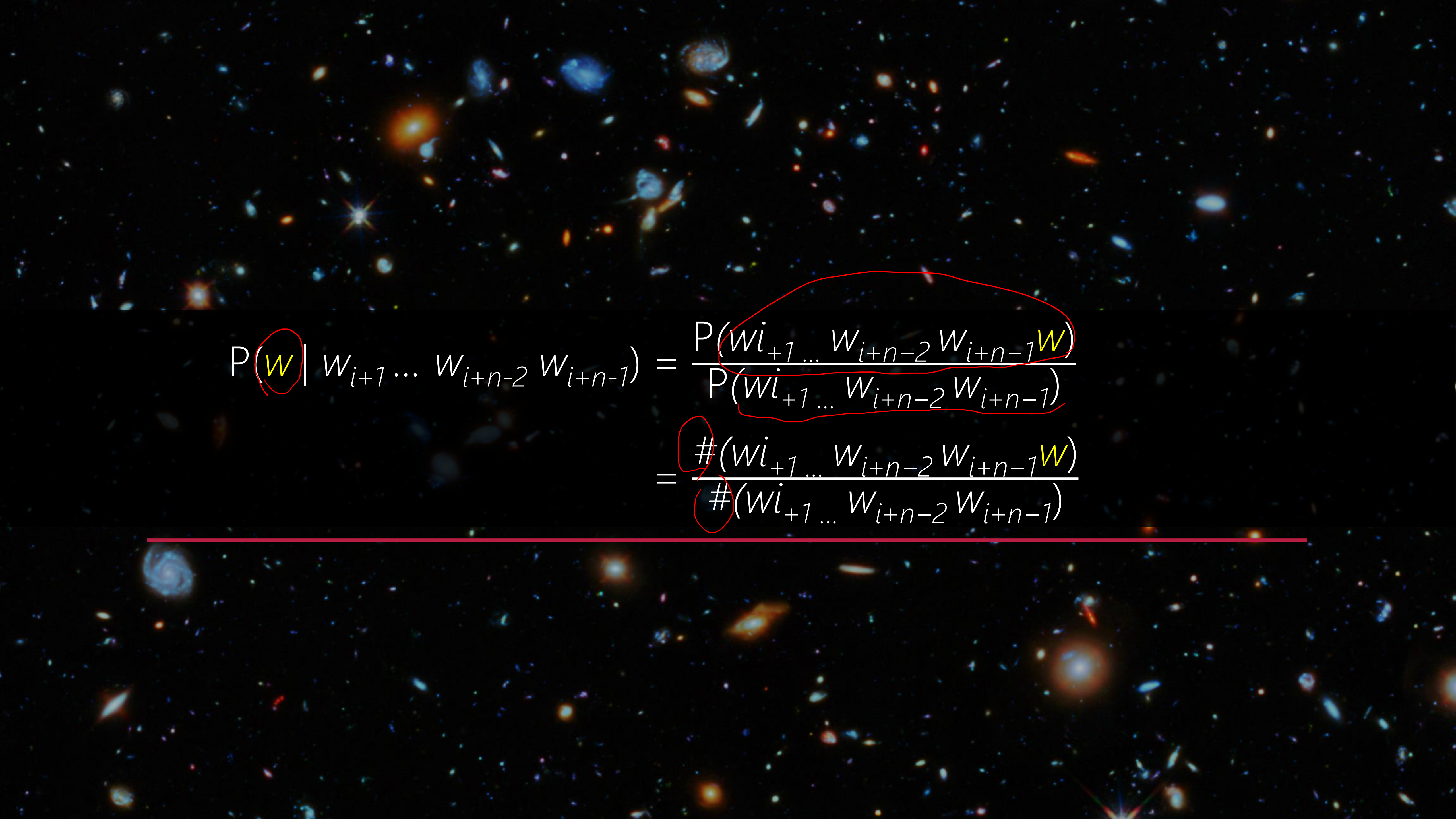
as opposed to Bayesian Probability

Frequentist probability or frequentism is an interpretation of probability that defines an event's probability as the limit of its relative frequency in many trials - Wikipedia

n-Gram Language Modeling

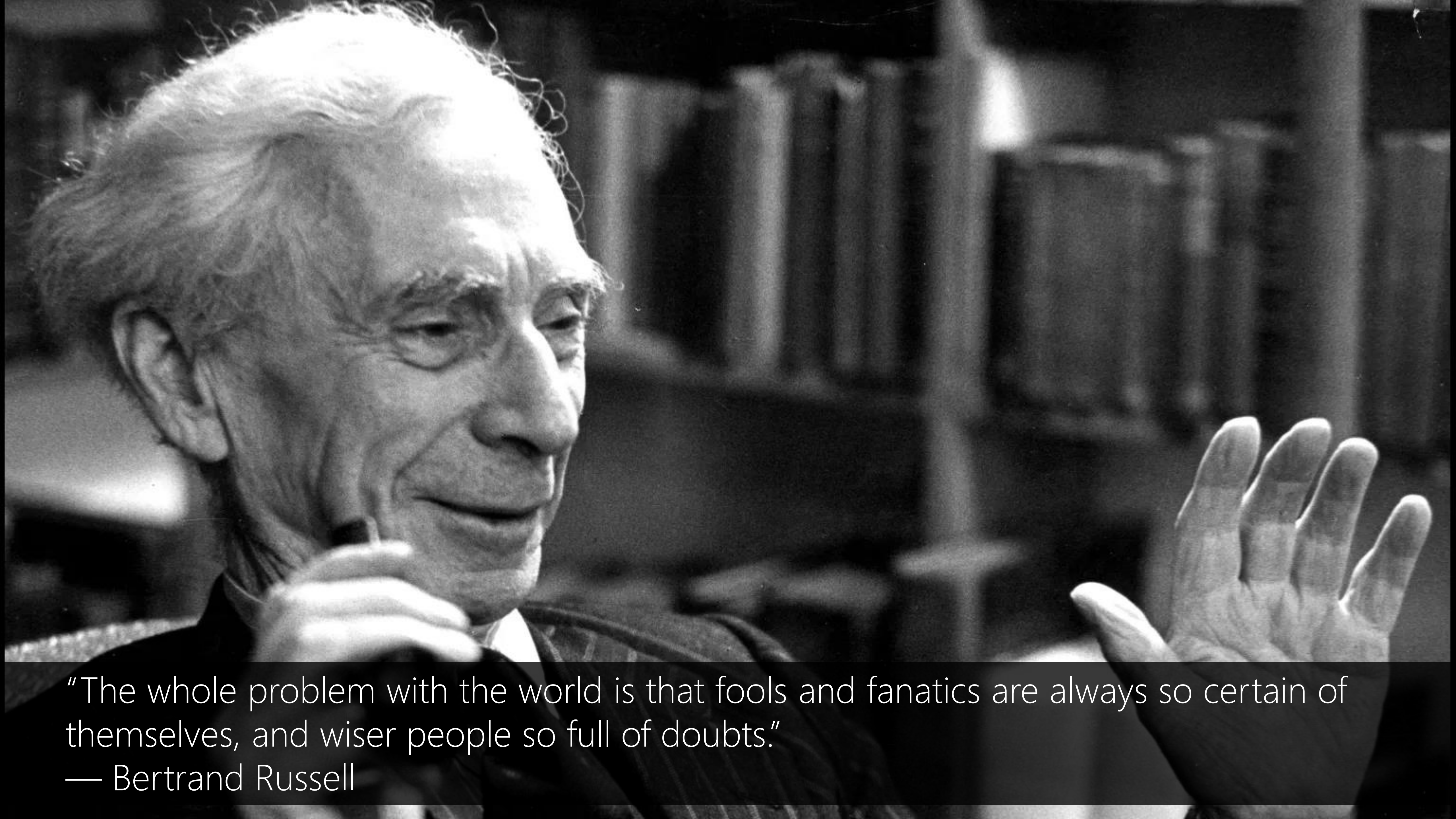
Recent Past of Size *n-1* → Future of Size 1 → Most Frequent Future Given the Past

$$\overset{n-1}{\underline{w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}}} \rightarrow \boxed{w_{i+n}} = \text{Max } P(w \mid \underline{w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}}) \text{ in all } w \in V$$

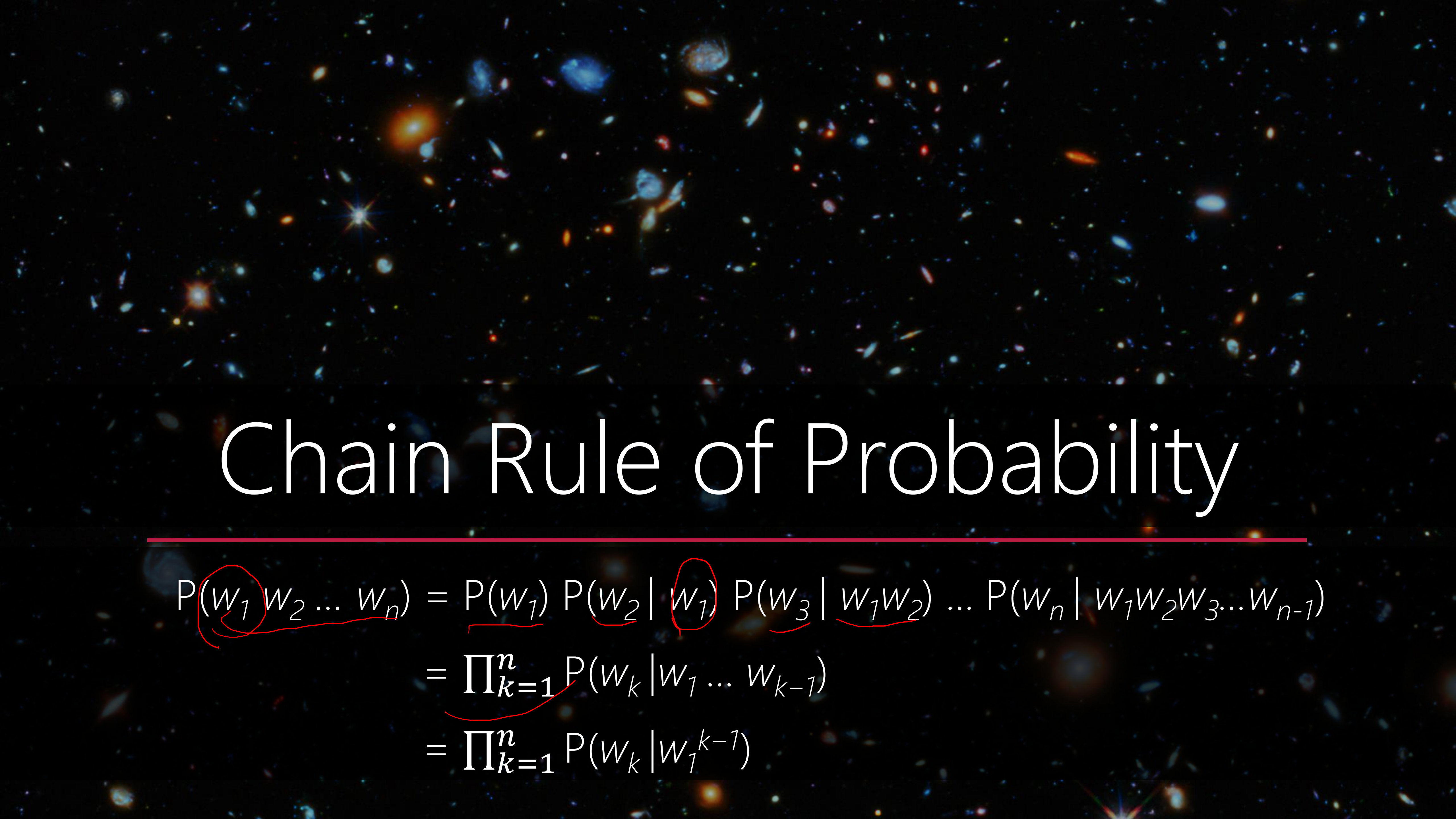


$$P(w \mid w_{i+1} \ldots w_{i+n\text{-}2}\, w_{i+n\text{-}1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

$$= \frac{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$



“The whole problem with the world is that fools and fanatics are always so certain of themselves, and wiser people so full of doubts.”
— Bertrand Russell

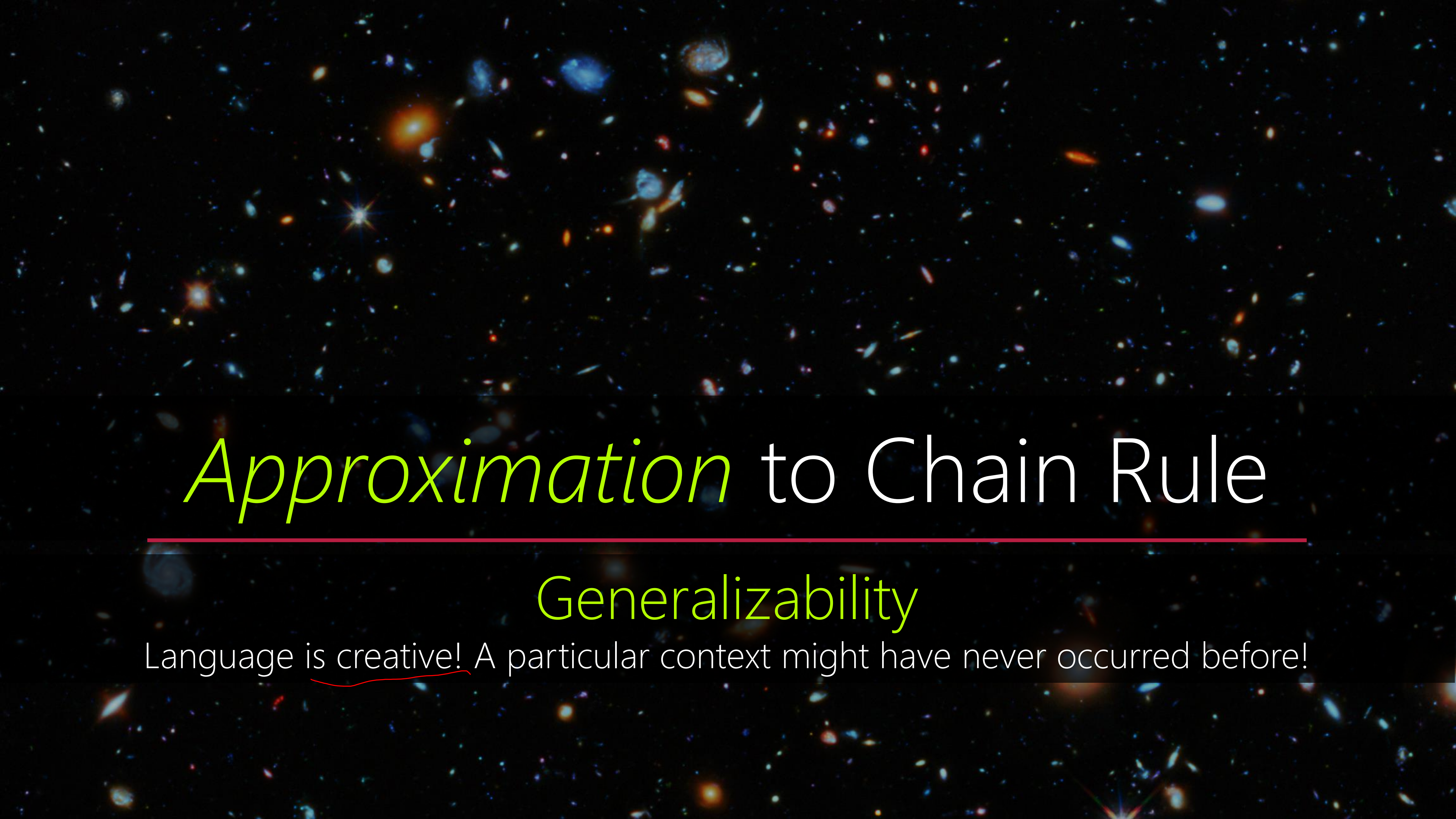


Chain Rule of Probability

$$P(w_1 w_2 \ldots w_n) = P(w_1)\, P(w_2 | w_1)\, P(w_3 | w_1 w_2) \ldots P(w_n | w_1 w_2 w_3 \ldots w_{n-1})$$

$$= \prod_{k=1}^{n} P(w_k | w_1 \ldots w_{k-1})$$

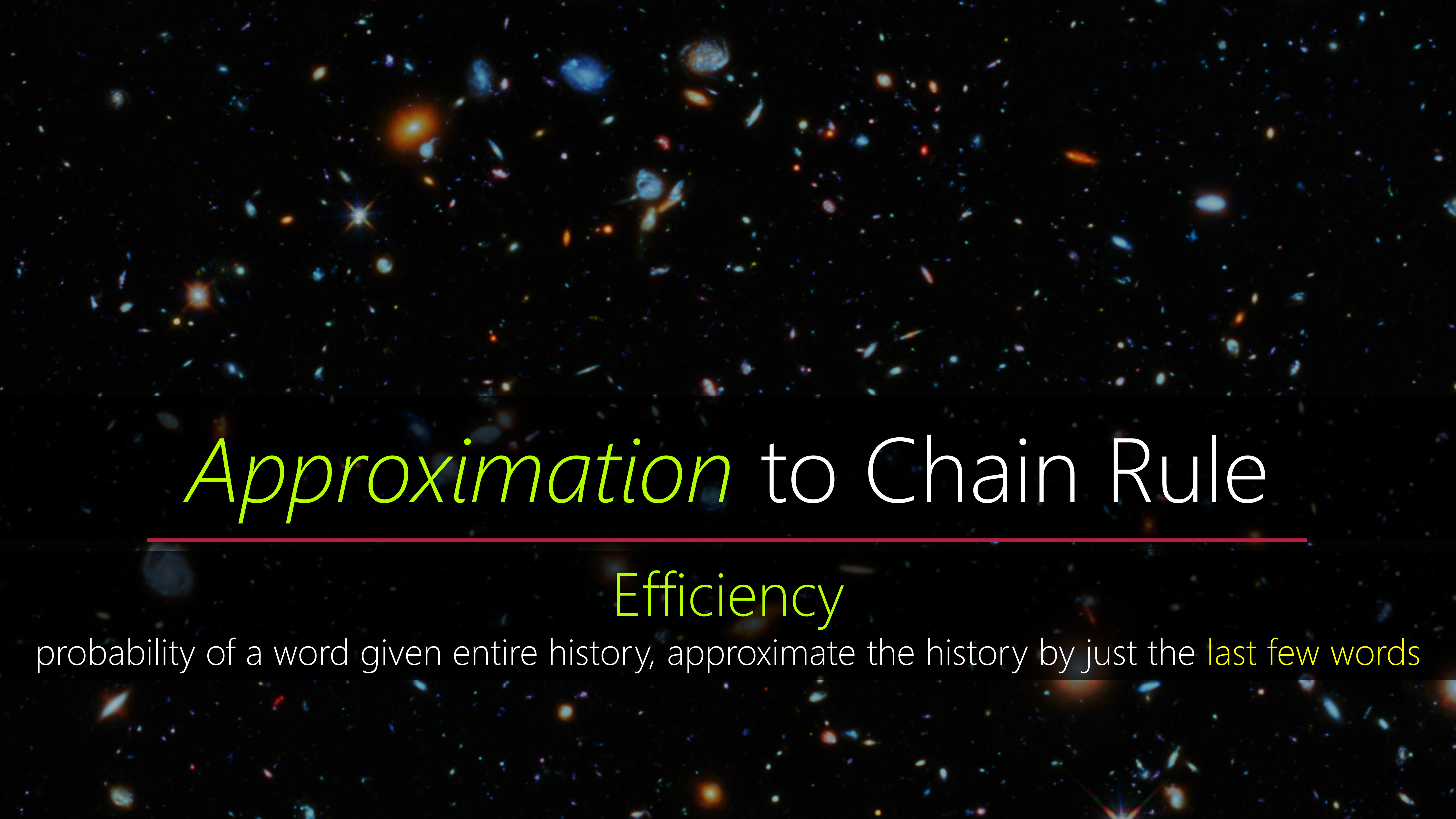
$$= \prod_{k=1}^{n} P(w_k | w_1^{k-1})$$



Approximation to Chain Rule

Generalizability

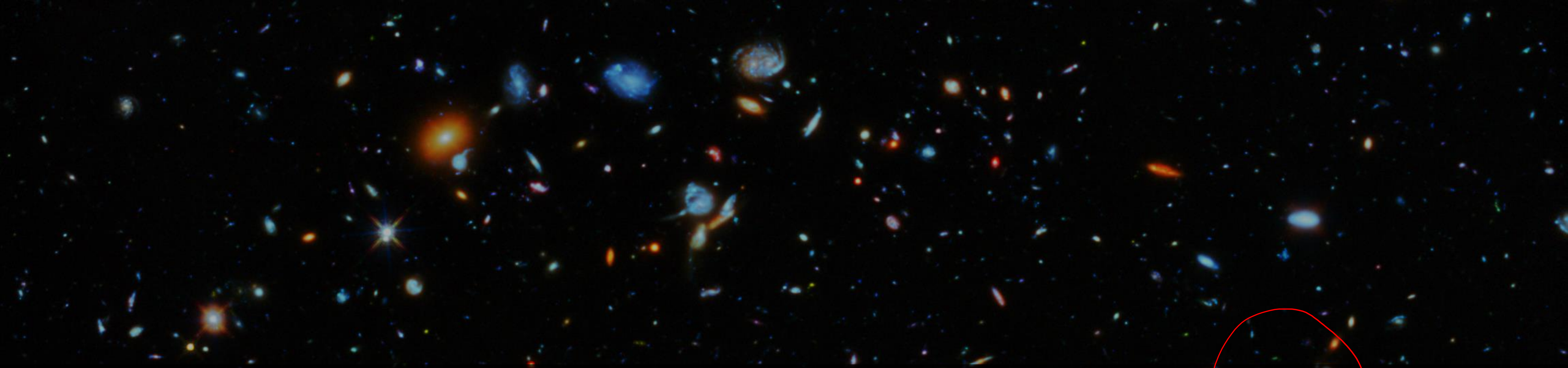
Language is creative! A particular context might have never occurred before!



Approximation to Chain Rule

Efficiency

probability of a word given entire history, approximate the history by just the last few words



Unigram Approx.

Bag-of-Word (BoW). Why?

$$P(w_1\ w_2 \ldots\ w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|w_1w_2w_3 \ldots w_{n-1})$$

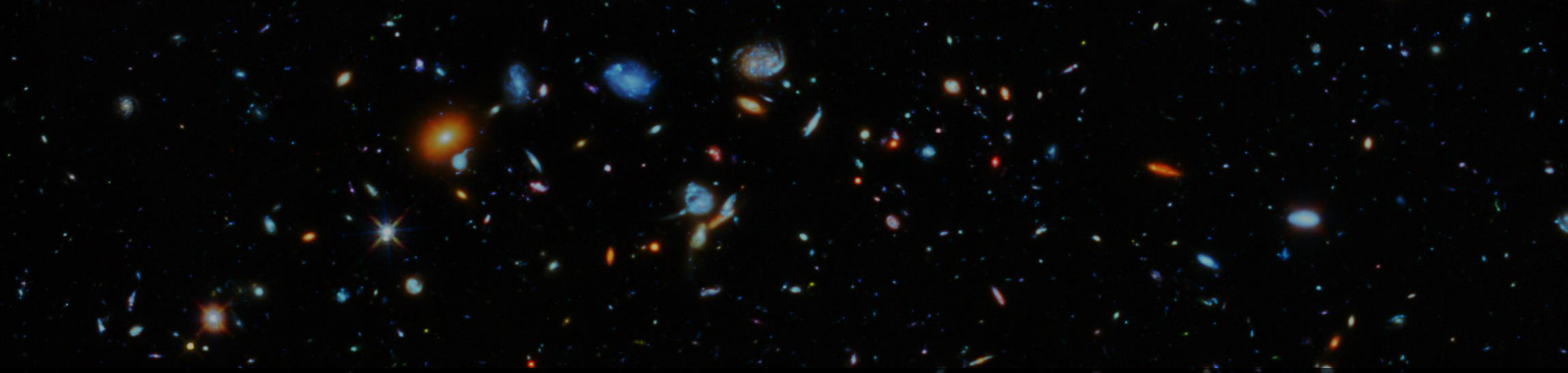
$$= P(w_1)P(w_2|\quad)P(w_3|\qquad) \ldots P(w_n|\qquad\qquad)$$

$$= P(w_1)P(w_2)P(w_3) \ldots P(w_n)$$

$\frac{\#w}{|token|}$

$\sim N(1,2)$

$\sim N(0,1)$



Bigram Approx.

Markovian: probability of a variable depends only on the previous variable

$$P(\underline{w_1\ w_2 \ldots\ w_n}) = P(w_1)P(w_2|w_1)P(w_3|\cancel{w_1}w_2) \ldots P(w_n|w_1w_2w_3\ldots w_{n-1})$$

$$= \underline{P(w_1)P(w_2|w_1)}P(w_3|\quad w_2) \ldots P(w_n|\qquad\qquad w_{n-1})$$

Trigram Approx.

$$\begin{aligned} P(w_1\ w_2 \ldots\ w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|w_1w_2w_3\ldots w_{n-1}) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \ldots P(w_n|\qquad w_{n-2}\, w_{n-1}) \end{aligned}$$

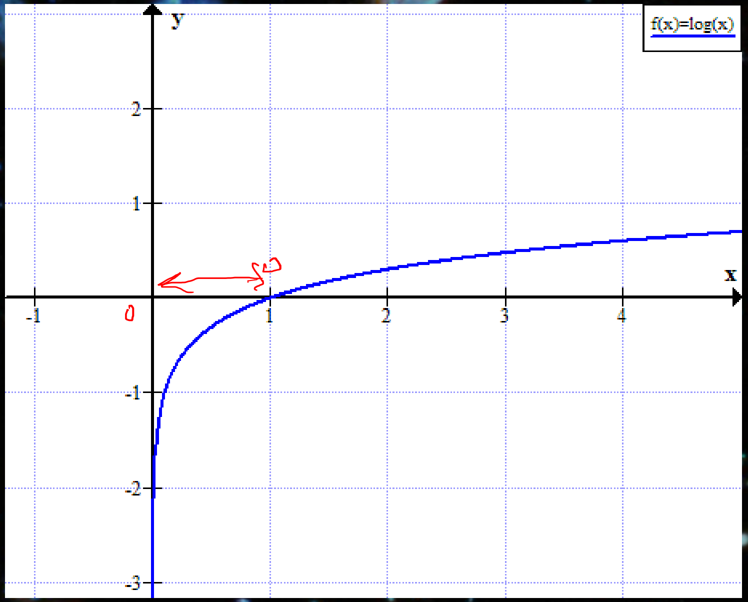
Approx. n-gram Language Modeling

Corpus: Brown University

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', 'Friday', 'an', 'investigation', 'of', ..., '.'],
['The', 'jury', 'further', 'said', 'in', 'term-end', 'presentments', 'that', 'the', 'City', ... 'conducted', '.'],
['The', 'September-October', 'term', 'jury', 'had', 'been', 'charged', 'by', 'Fulton', 'S..., 'Allen', 'Jr.', '.'],
['``', 'Only', 'a', 'relative', 'handful', 'of', 'such', 'reports', 'was', 'received', "''", ',, '... 'city', "''", '.'],
['The', 'jury', 'said', 'it', 'did', 'find', 'that', 'many', 'of', "Georgia's", 'registration', ... 'ambiguous', "''", '.']

Make it worse!	Gives chance to new combination
P([Mr.][and][Mrs.])	P([Mr.][and][I])
0.00045851027827207127	0.0
1.4208331509791766e-05	1.7517121039469З e-06
9.0782284239431O8e-08	6.42293631575421 4e-08

Unig



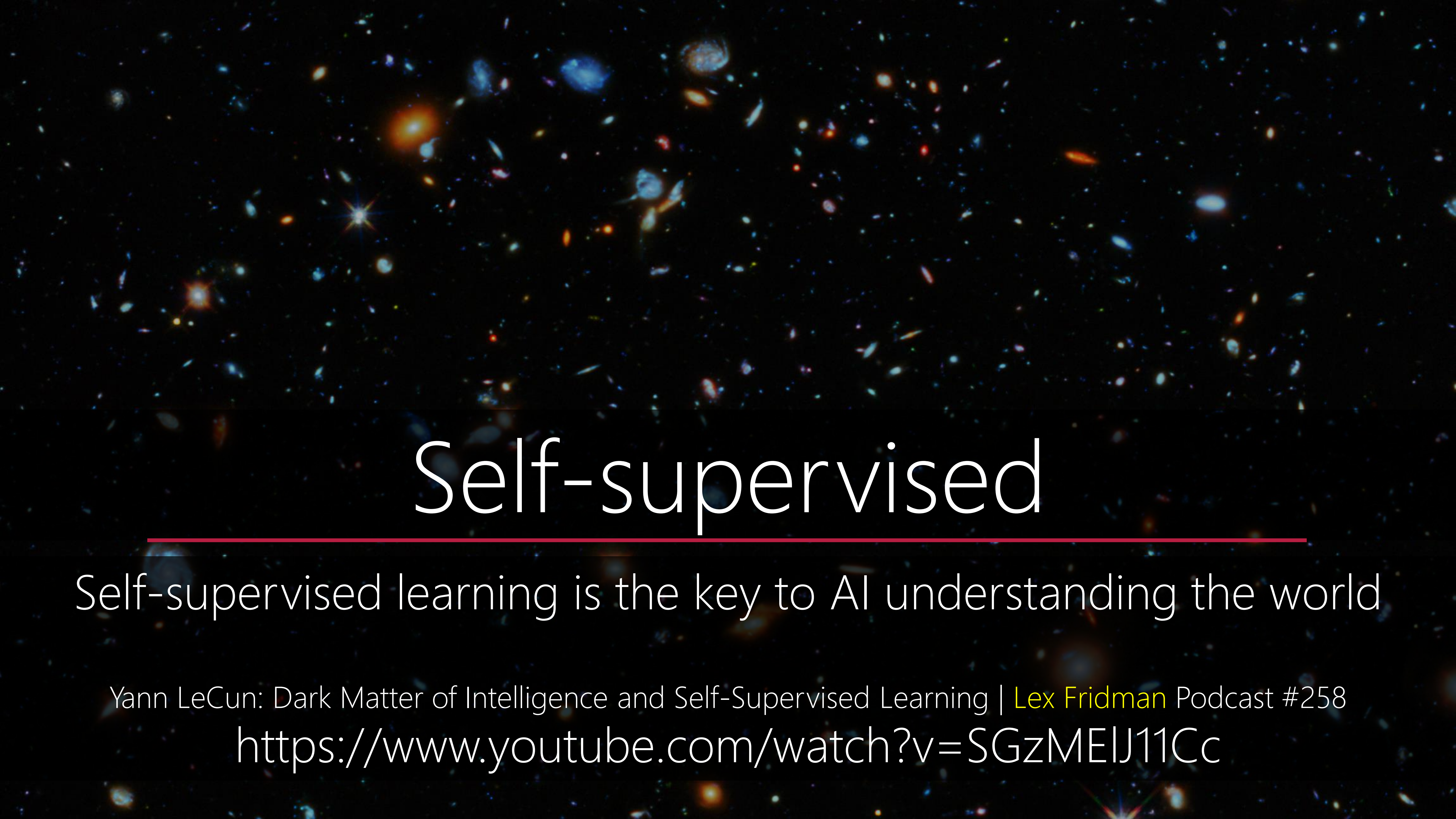
Log of Probabilities

$$P(x_1) \times P(x_2) \times \ldots \times P(x_n) \propto \underline{\log P(x_1)} + \log P(x_2) + \ldots + \log P(x_n)$$

left and right sides have same order!

$[0, 1] \longrightarrow [-\infty, 0]$

Product ⟶ Sum



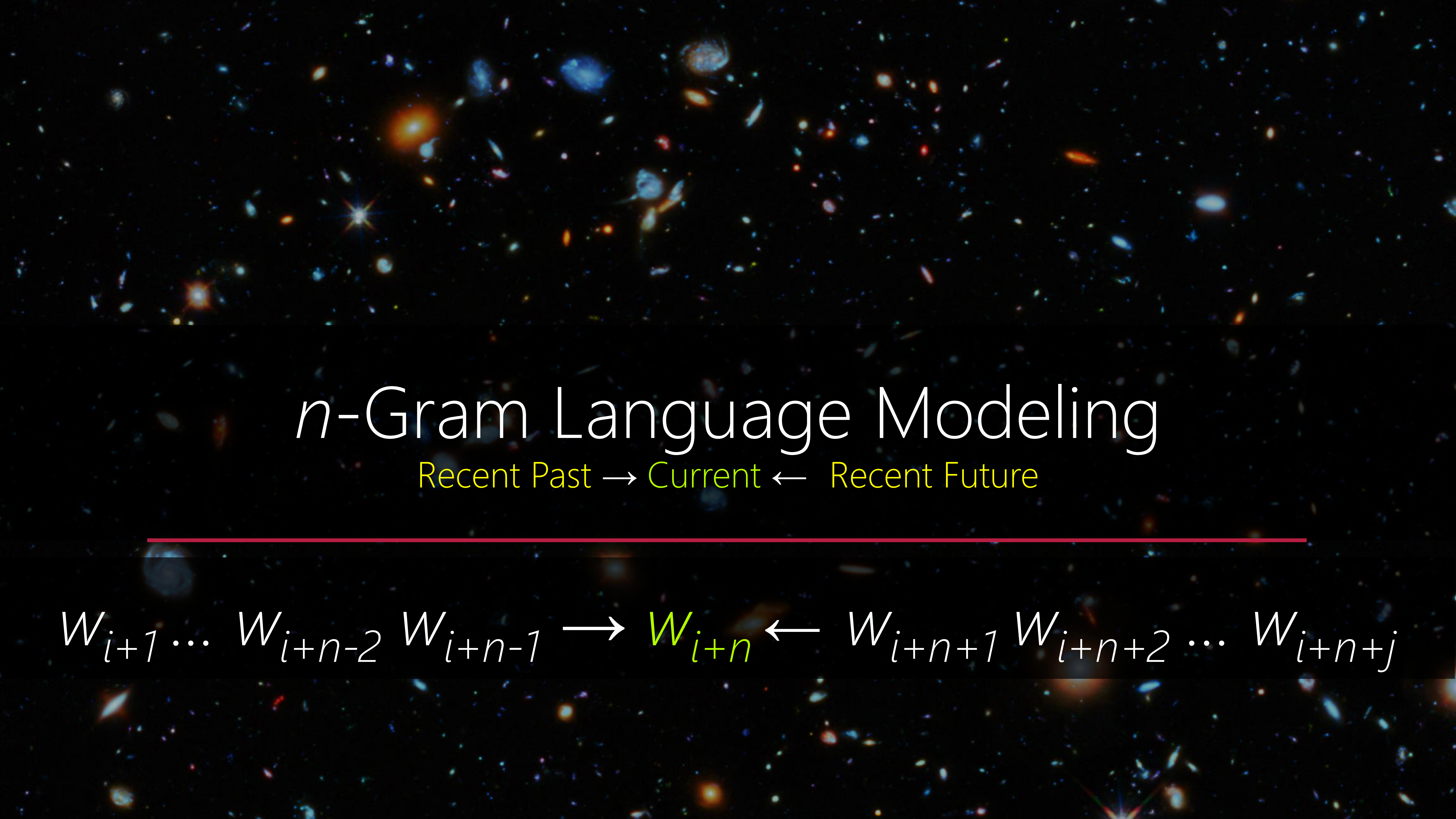
Self-supervised

Self-supervised learning is the key to AI understanding the world

Yann LeCun: Dark Matter of Intelligence and Self-Supervised Learning | Lex Fridman Podcast #258

https://www.youtube.com/watch?v=SGzMElJ11Cc

Tenet, Christopher Nolan, 2020
Budget $200 m
Box office $363 million



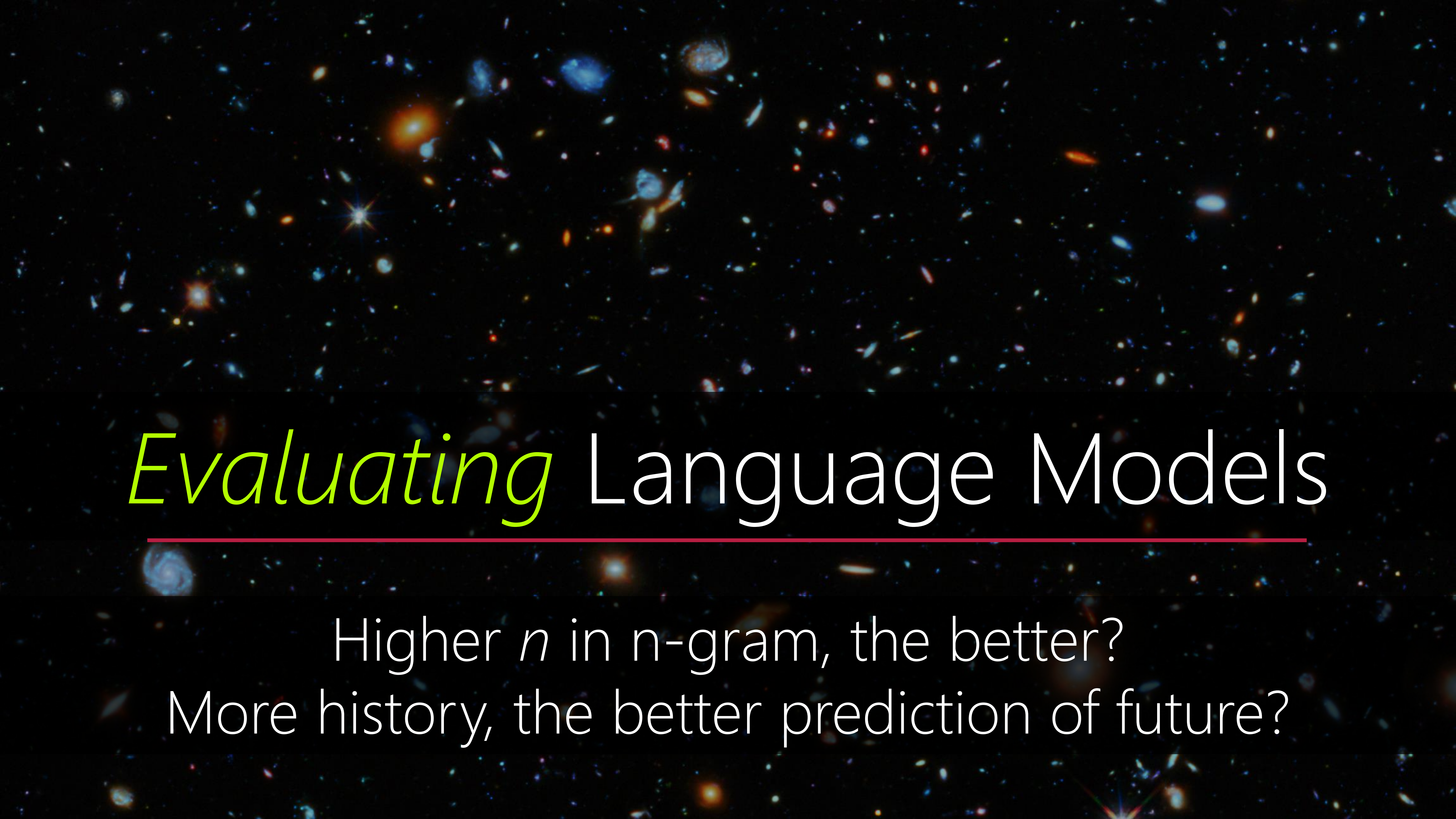
n-Gram Language Modeling

Recent Past → Current ← Recent Future

$$w_{i+1} \cdots w_{i+n-2}\, w_{i+n-1} \longrightarrow w_{i+n} \longleftarrow w_{i+n+1}\, w_{i+n+2} \cdots w_{i+n+j}$$

Evaluating Language Models

Following, Christopher Nolan (1998)
Budget $6,000
Box office $48,482

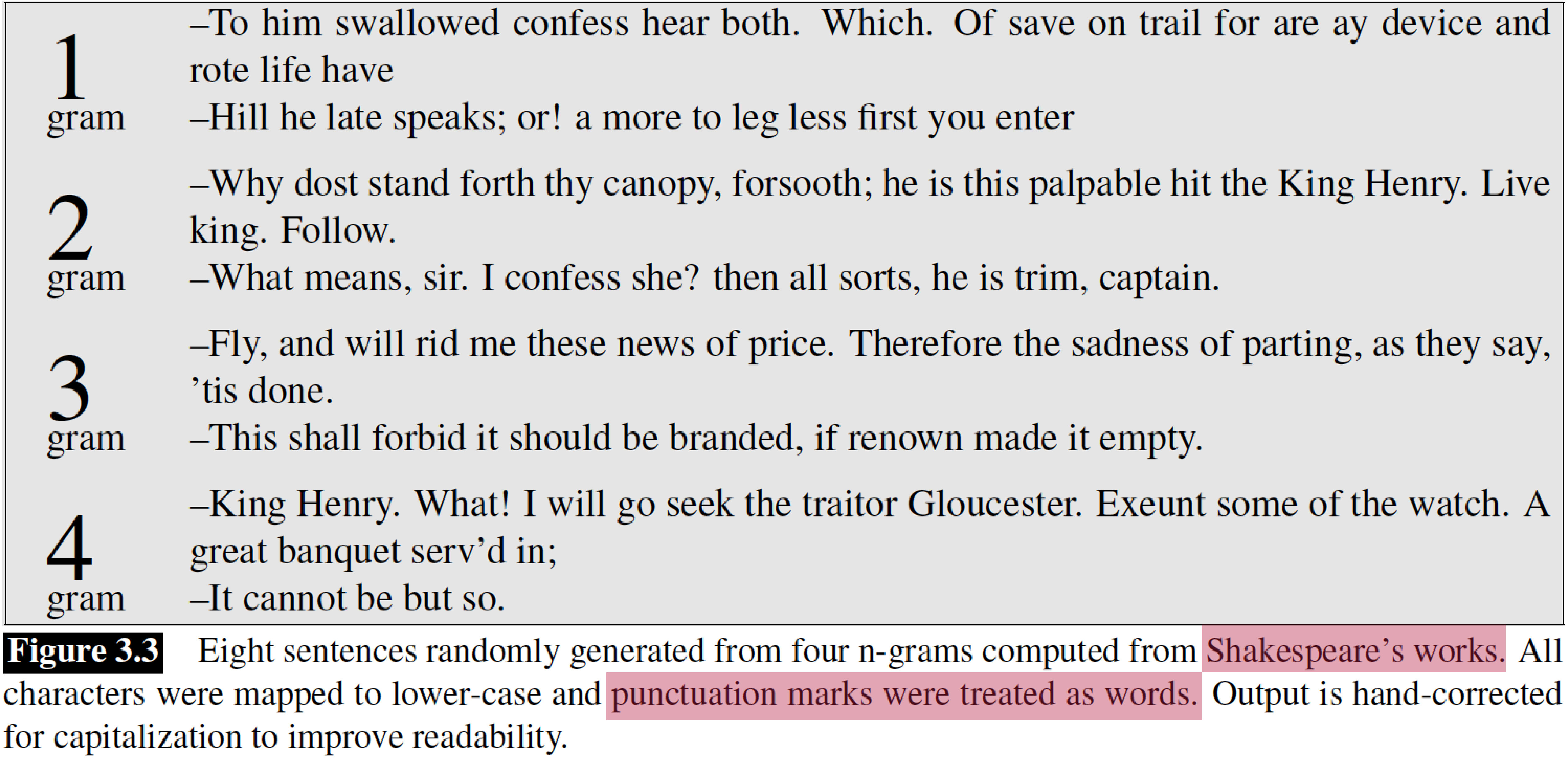


Evaluating Language Models

Higher *n* in n-gram, the better?
More history, the better prediction of future?

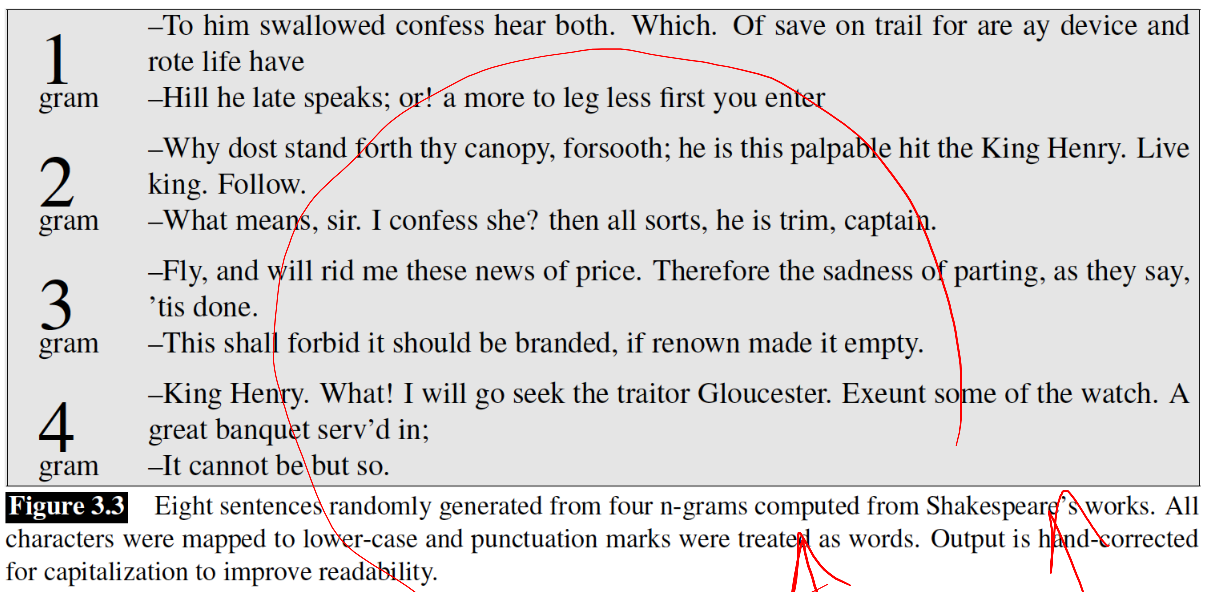
Evaluating Language Models

Qualitative → Let’s Communicate → Generate



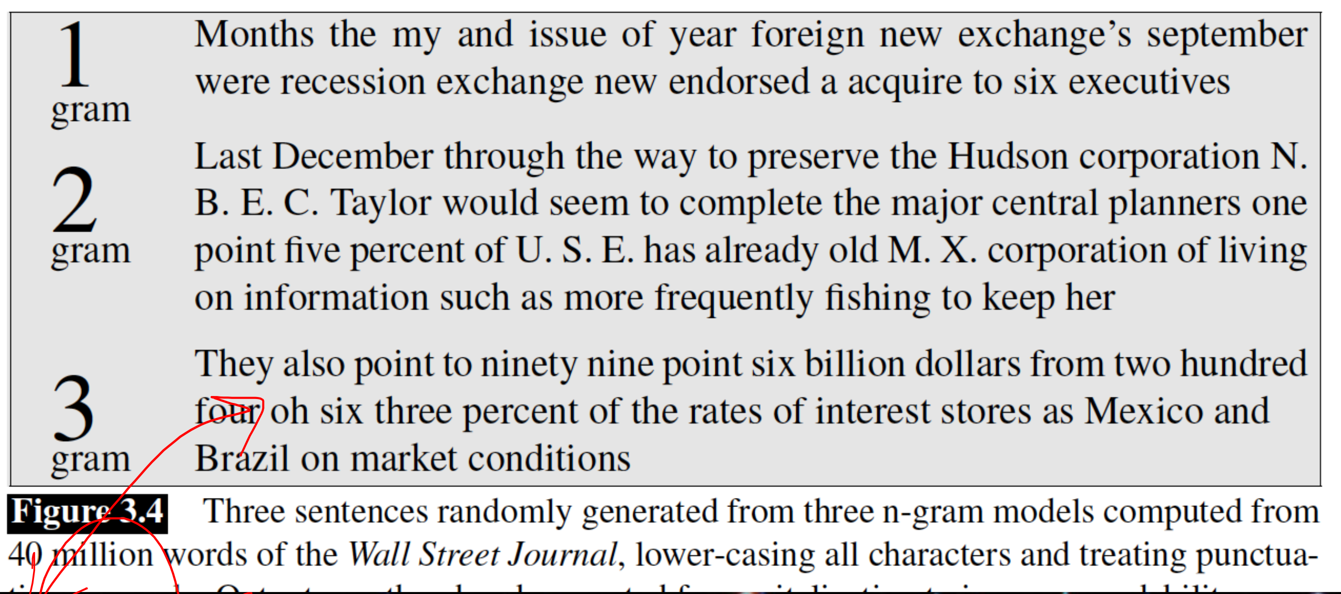
1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess 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.

Figure 3.3 Eight sentences randomly generated from four n-grams computed from Shakespeare’s works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.



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.

Figure 3.3 Eight sentences randomly generated from four n-grams computed from Shakespeare’s works. All characters were mapped to lower-case and punctuation marks were treated as words. Output is hand-corrected for capitalization to improve readability.



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

Figure 3.4 Three sentences randomly generated from three n-gram models computed from 40 million words of the *Wall Street Journal*, lower-casing all characters and treating punctua-

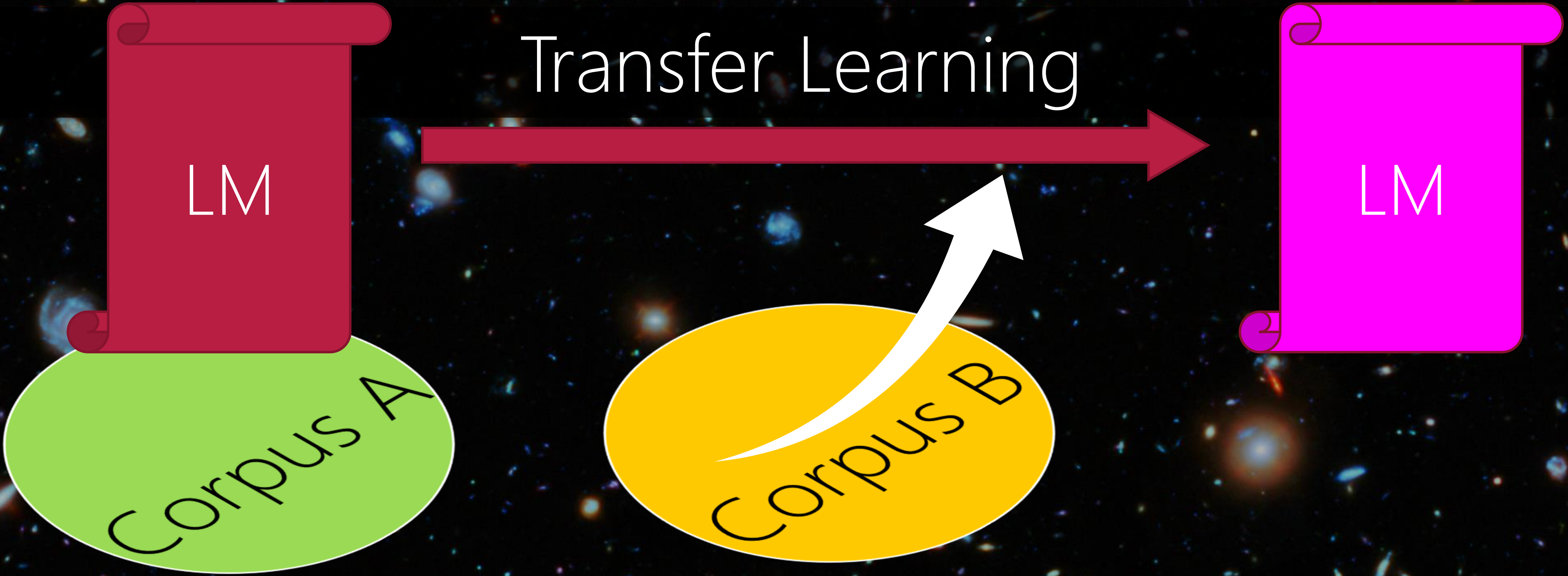
$w_1 w_2 w_3$

Cross Evaluating Language Models

Biased toward the corpus! Dialect, Genre, ...

Better LM is the one that can generalize!

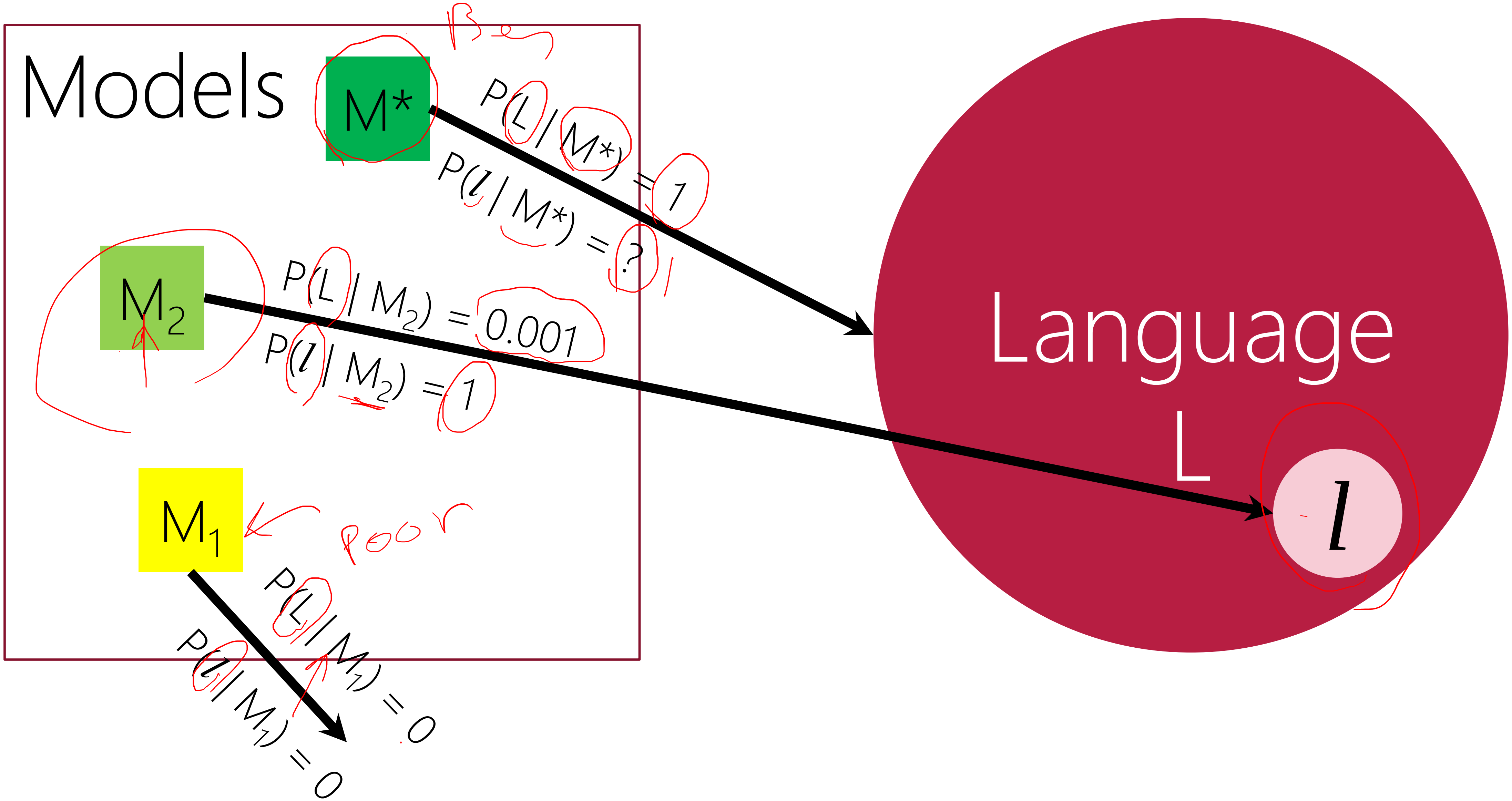
Pre-trained Language Models



$$x_{t+1} = f\left((x_t) x_{t-1}\right)$$

Evaluating Language Models

Quantitative → Likelihood



Models

M^*

Ben

$P(L \mid M^*) = 1$

$P(l \mid M^*) = ?$ 1

M_2

$P(L \mid M_2) = 0.001$

$P(l \mid M_2) = 1$

M_1

poor

$P(L \mid M_1) = 0$

$P(l \mid M_1) = 0$

Language

L

l

Find M*

Golden Model

Assumptions:

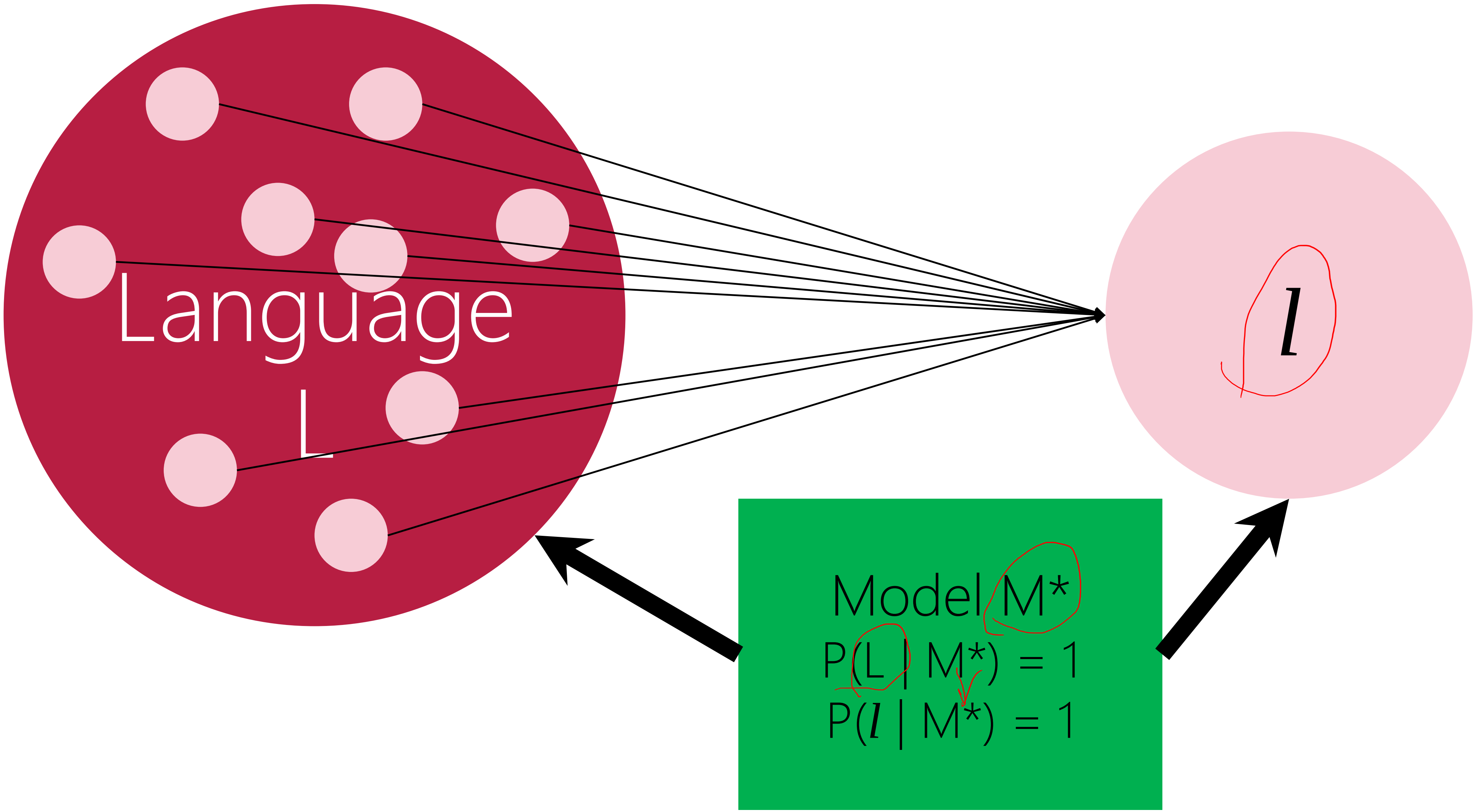
- The language space is available. ✗
- Search the model space to find M*, assuming it exists!

Find M*

Golden Model

Relax Assumptions:

- The language space is available. → Random Subsets
- Search the model space to find M*, assuming it exists!



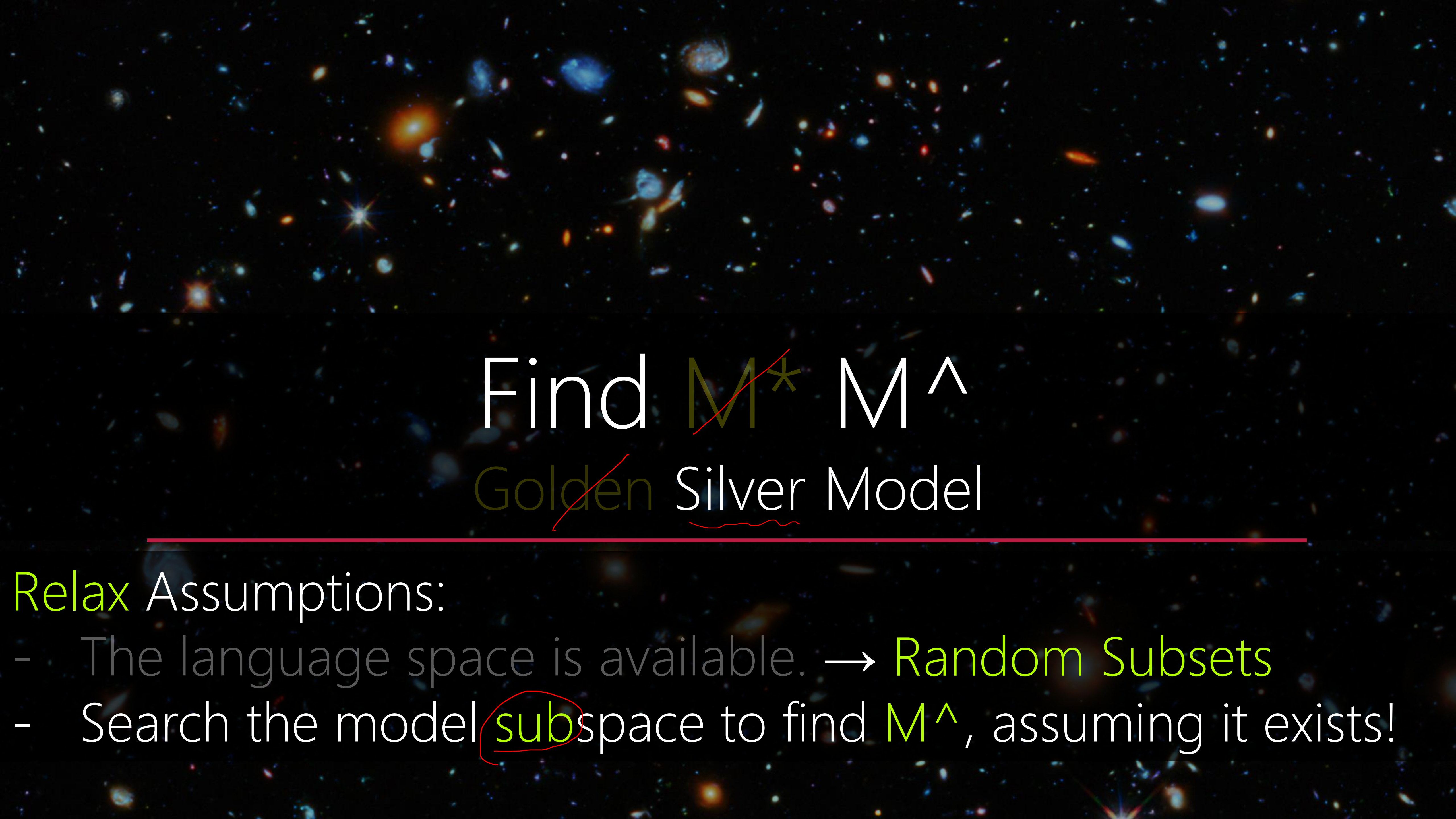
Language L

l

Model M*

$P(L \mid M^*) = 1$

$P(l \mid M^*) = 1$

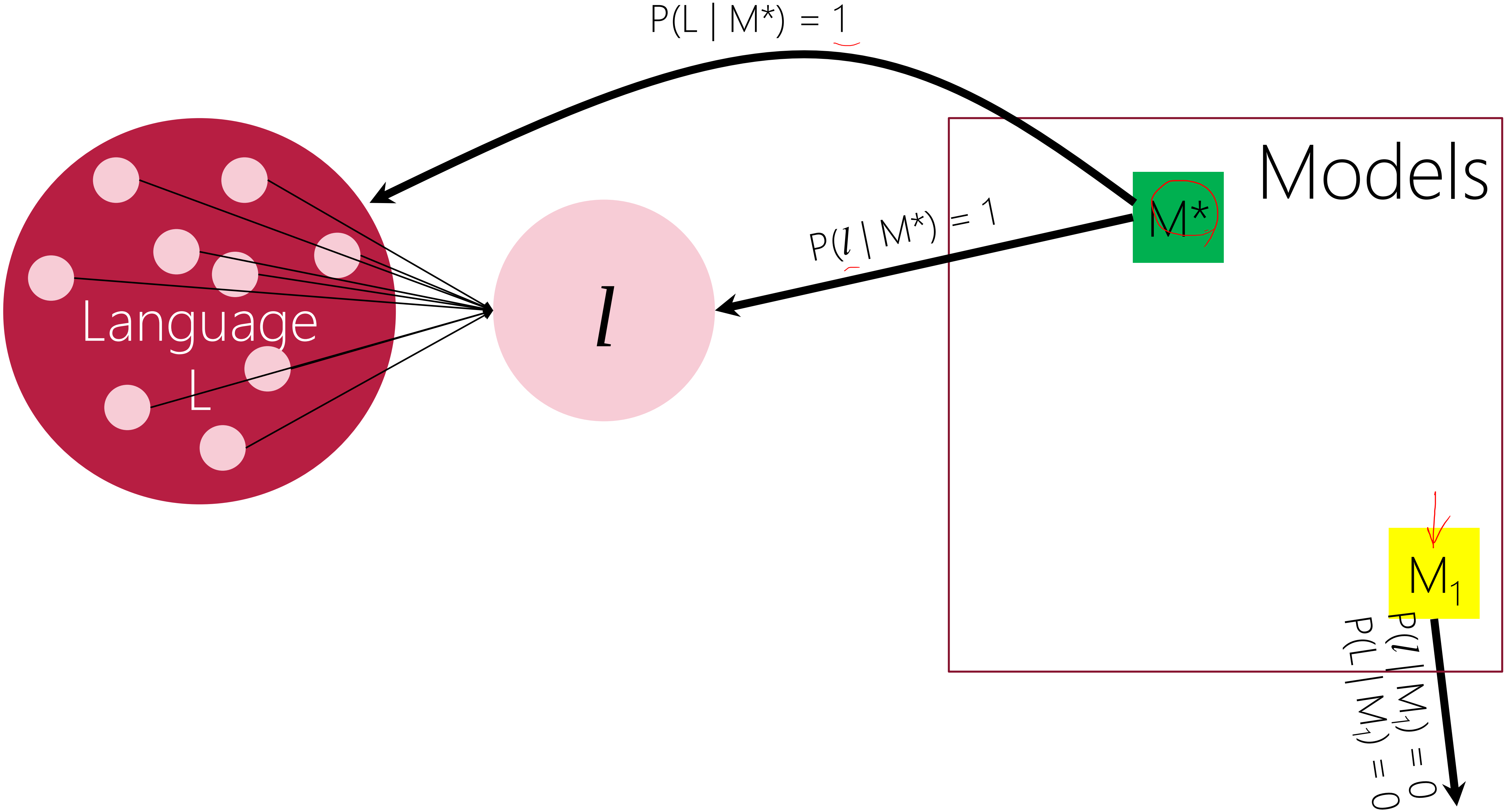


Find ~~M*~~ M^

~~Golden~~ Silver Model

Relax Assumptions:

- The language space is available. → Random Subsets
- Search the model subspace to find M^, assuming it exists!



$P(L \mid M^*) = 1$

Models

M^*

$P(l \mid M^*) = 1$

l

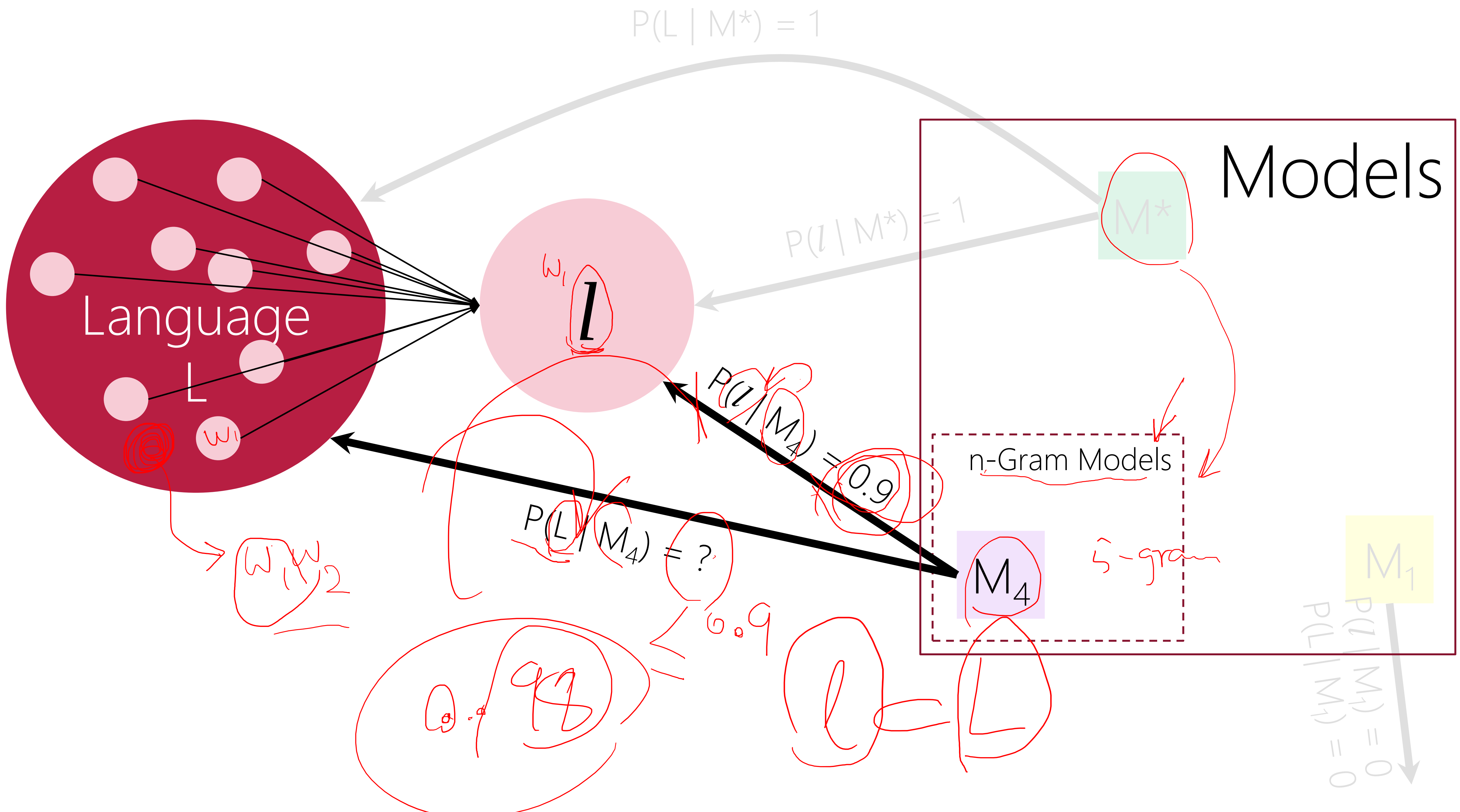
Language

L

M_1

$P(l \mid M_1) = 0$

$P(L \mid M_1) = 0$



$P(L \mid M^*) = 1$

$P(l \mid M^*) = 1$

Models

M^*

Language L

l

$P(l \mid M_4) = 0.9$

$P(L \mid M_4) = ?$

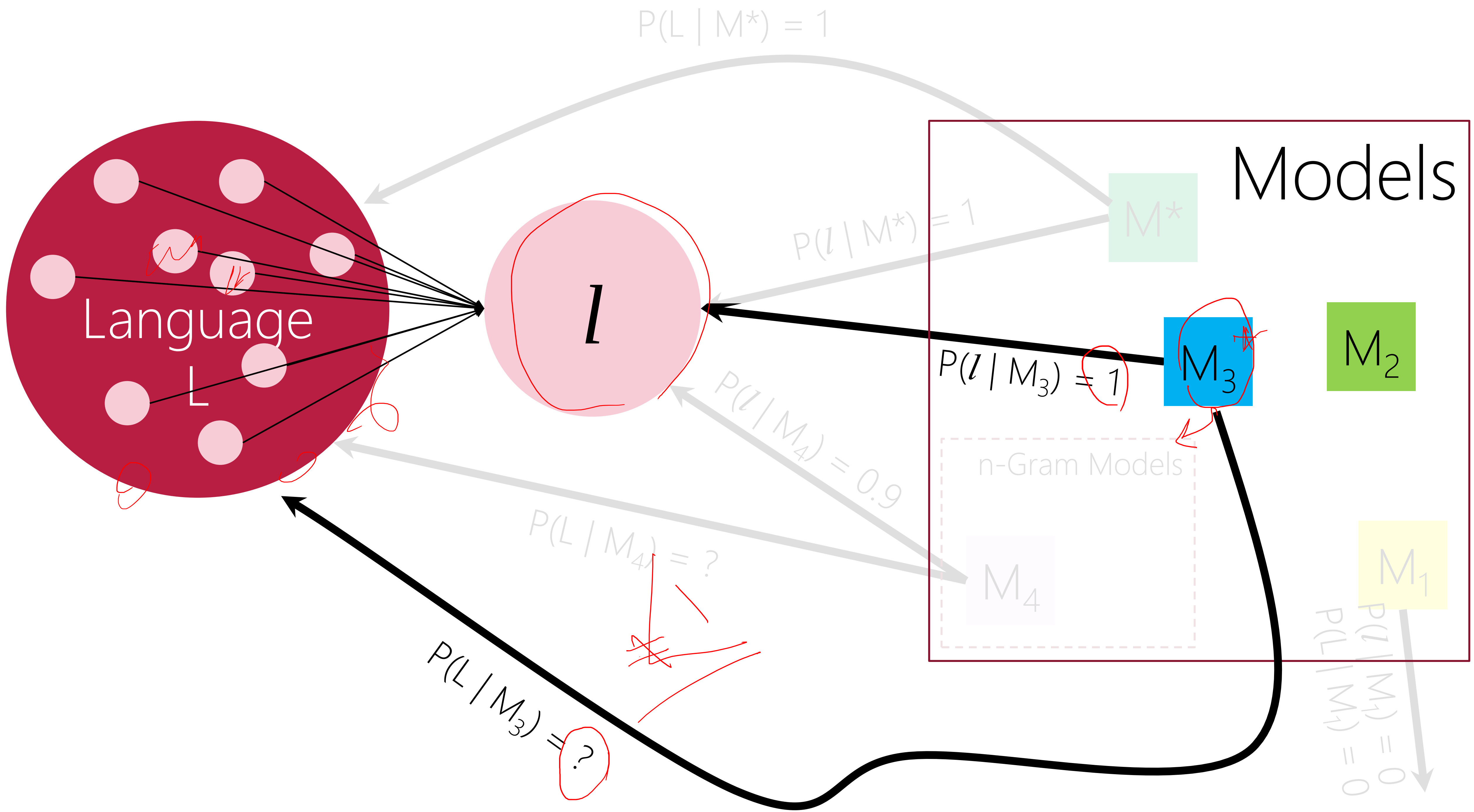
n-Gram Models

M_4

M_1

$P(l \mid M_1) = 0$

$P(L \mid M_1) = 0$



Models

M^*

M_3

M_2

n-Gram Models

M_4

M_1

Language L

l

$P(L \mid M^*) = 1$

$P(l \mid M^*) = 1$

$P(l \mid M_3) = 1$

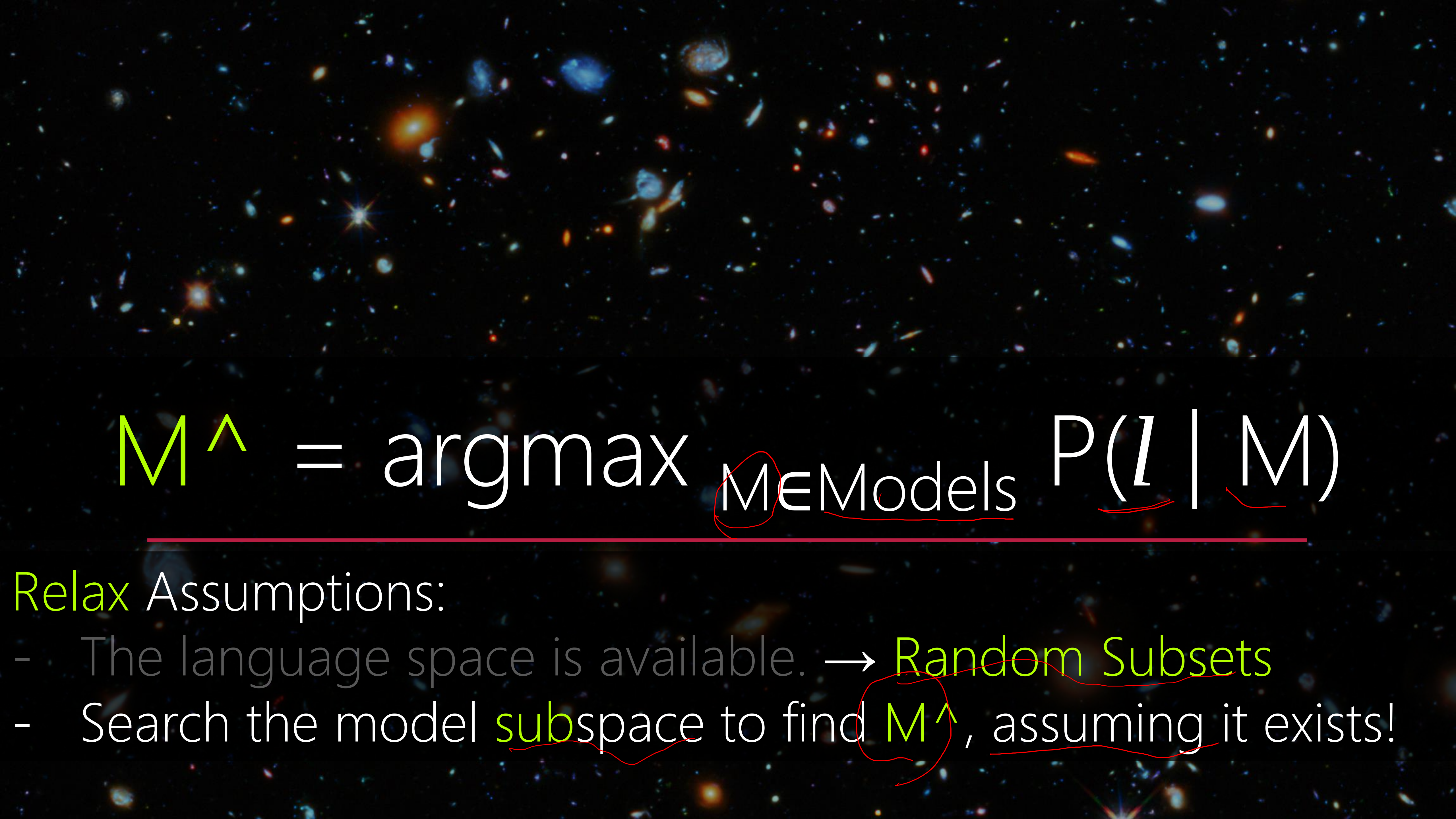
$P(l \mid M_4) = 0.9$

$P(L \mid M_4) = ?$

$P(L \mid M_3) = ?$

$P(l \mid M_1) = 0$

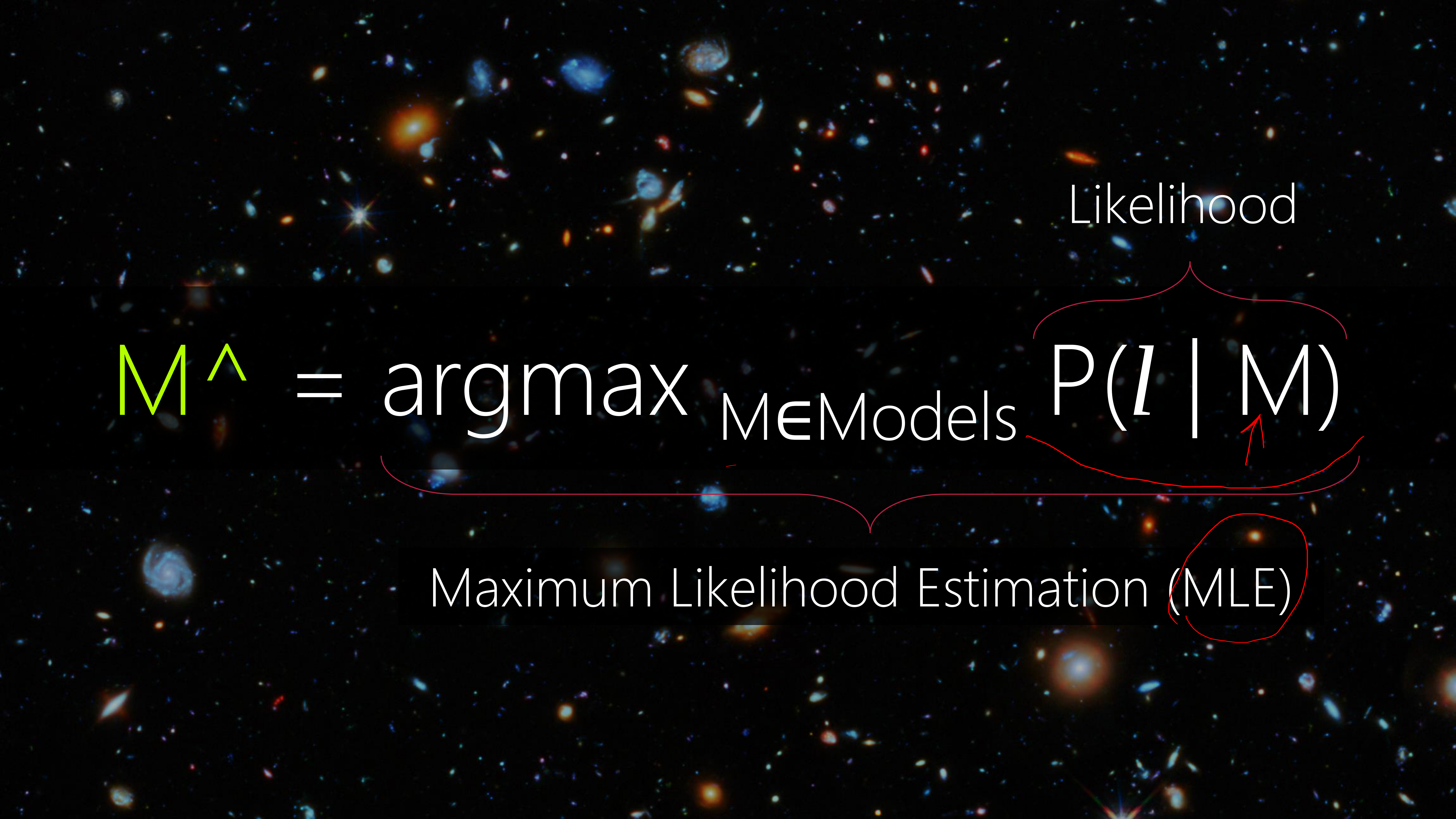
$P(L \mid M_1) = 0$



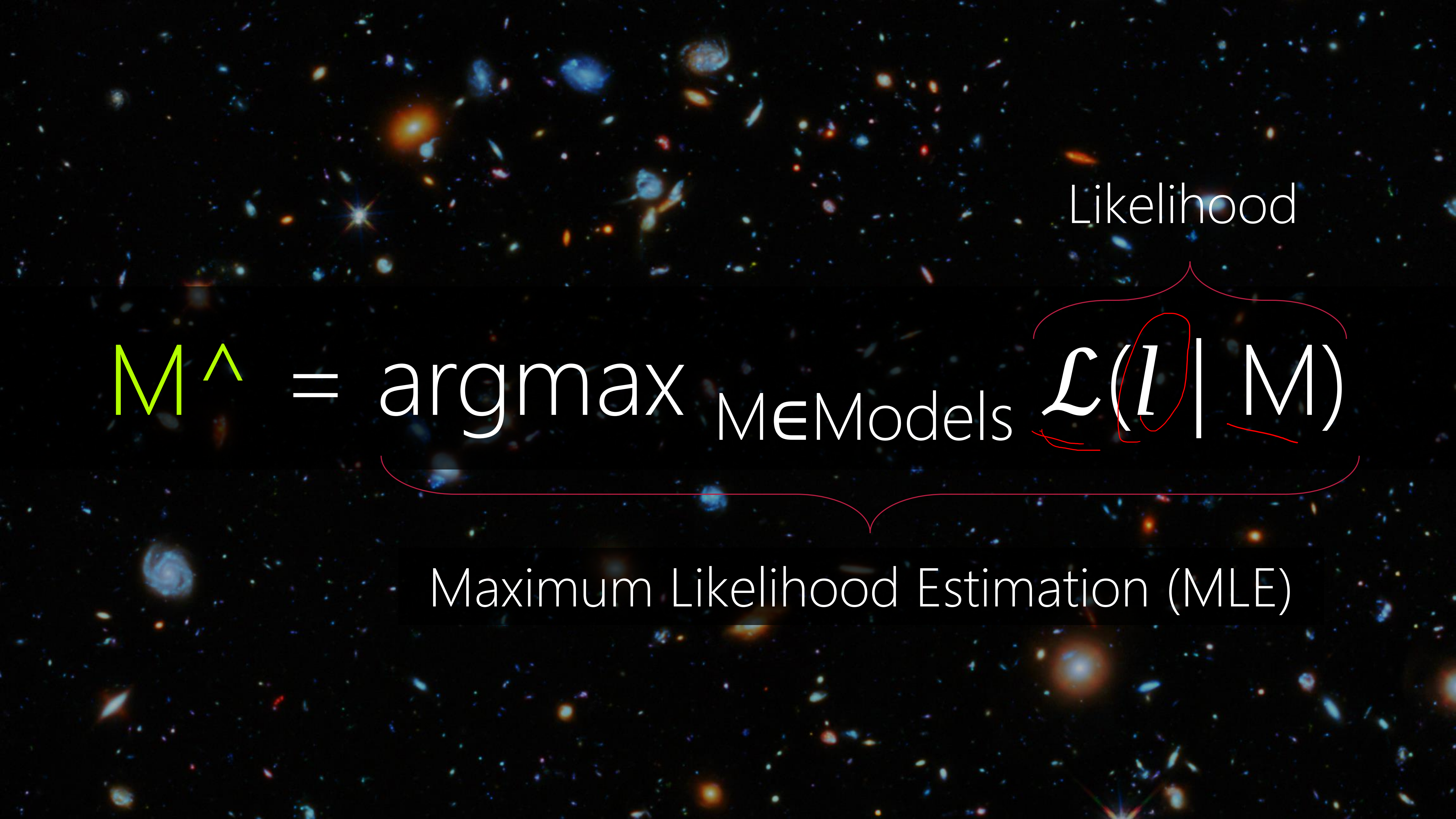
$$M^{\wedge} = \operatorname{argmax}_{M \in \text{Models}} P(l \mid M)$$

Relax Assumptions:

- The language space is available. → Random Subsets
- Search the model subspace to find M^, assuming it exists!



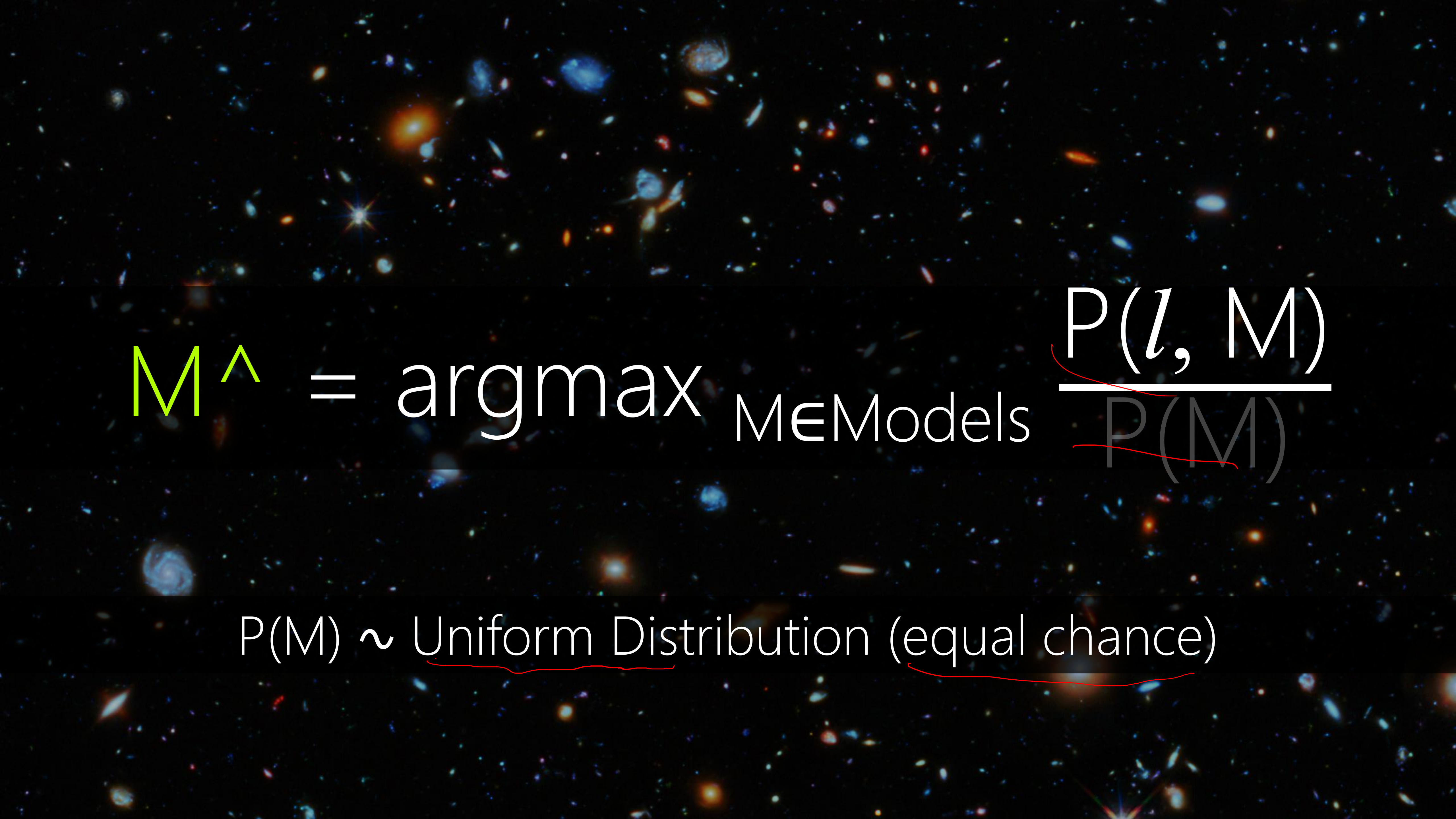
$$M\,\hat{} = \underbrace{\operatorname{argmax}_{M \in \mathrm{Models}} \overbrace{P(l \mid M)}^{\text{Likelihood}}}_{\text{Maximum Likelihood Estimation (MLE)}}$$



$$M^{\wedge} = \underbrace{\operatorname{argmax}_{M\in\text{Models}} \overbrace{\underline{\mathcal{L}}(l \mid \underline{M})}^{\text{Likelihood}}}_{\text{Maximum Likelihood Estimation (MLE)}}$$

$P(l|M)$

$$\hat{M} = \operatorname{argmax}_{M \in \text{Models}} \frac{P(l, M)}{P(M)}$$



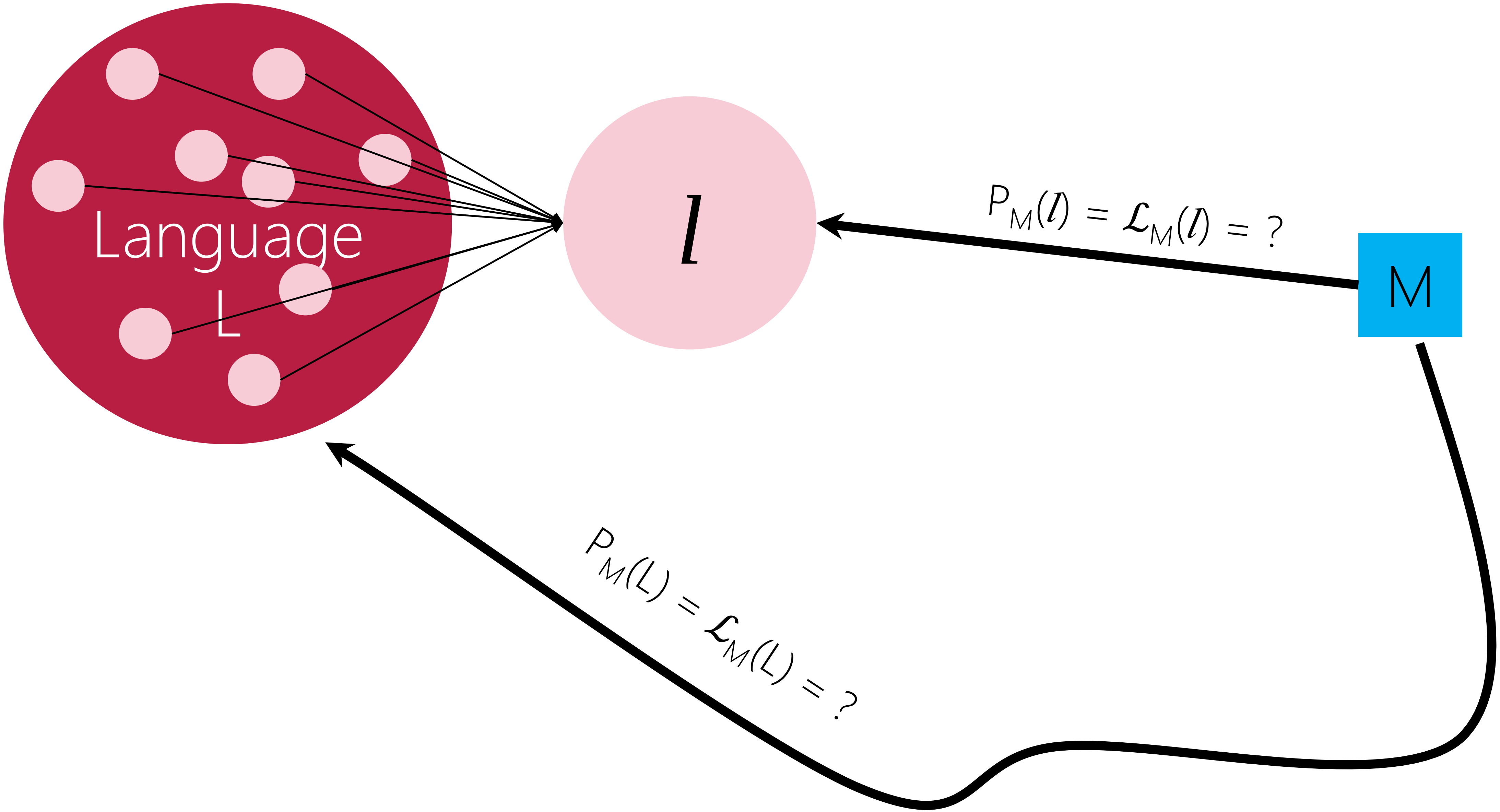
$$M\hat{} = \operatorname{argmax}_{M \in \mathrm{Models}} \frac{P(l, M)}{\cancel{P(M)}}$$

P(M) ~ Uniform Distribution (equal chance)

$$M\,\hat{} = \text{argmax}_{M \in \text{Models}} P(l, M)$$

$$\boxed{P(l, M)} = \underline{P_{\boxed{M}}(l)} = \underline{\mathcal{L}_{M}(l)}$$

$$M\,\hat{} = \underline{\text{argmax}_{M \in \text{Models}} \mathcal{L}_{M}(l)}$$


Language
L
l
$P_M(l) = \mathcal{L}_M(l) = ?$
M
$P_M(L) = \mathcal{L}_M(L) = ?$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

$$\text{M1} = |\text{token}|\text{-gram model} = 22\text{-gram model} \longrightarrow P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$$

$$P(\text{The c} \ldots) = \frac{\#(l)}{\#l}$$

We found our silver model!

M^ = M1

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

$$\text{M1} = |\text{token}|\text{-gram model} = \text{22-gram model} \longrightarrow P_{\text{22-gram}}(l) = \mathcal{L}_{\text{22-gram}}(l) = 1$$

$$\text{M2} = \underline{|\text{vocab}|}\text{-gram model} = \boxed{16}\text{-gram model} \longrightarrow P_{\underline{\text{16-gram}}}(l) = \underline{\mathcal{L}}_{\text{16-gram}}(l) = \boxed{?}$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$

M2 = |vocab|-gram model = $\underline{16}$-gram model → $P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$

P([The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.])

(the)

= P(the)(

P(course | the)((the cou))

= //

=

Chain Rule of Probability

$$P(w_1\, w_2 \ldots w_n) = P(w_1)\, P(w_2 \mid w_1)\, P(w_3 \mid w_1 w_2) \ldots P(w_n \mid w_1 w_2 w_3 \ldots w_{n\text{-}1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1 \ldots w_{k-1})$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $\mathrm{P}_{\text{22-gram}}(l) = \mathcal{L}_{\text{22-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $\mathrm{P}_{\text{16-gram}}(l) = \mathcal{L}_{\text{16-gram}}(l) = ?$

P([The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]) =
P([The])P([course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13] .... | [The])

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$

$$P(\text{[The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]}) =$$

$$P(\text{[The]})P(\text{[course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13] ....} \mid \text{[The]})$$

$$P(\text{[The]})P(\text{[course]} \mid \text{[The]})P(\text{[COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are] ....} \mid \text{[The][course]})$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$

$$P(\text{[The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]}) =$$

$$P(\text{[The]})P(\text{[course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13] ....} \mid \text{[The]})$$

$$P(\text{[The]})P(\text{[course]} \mid \text{[The]})P(\text{[COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are] ....} \mid \text{[The][course]})$$

$$P(\text{[The]})P(\text{[course]} \mid \text{[The]})P(\text{[}\underline{\text{COMP8730}}\text{]} \mid \text{[The][course]})P(\text{[is][about][}\underline{\text{nlp][.] ... There][are] ....}} \mid \underline{\text{[The][course][COMP8730}}\text{]})$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{\text{22-gram}}(l) = \mathcal{L}_{\text{22-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{\text{16-gram}}(l) = \mathcal{L}_{\text{16-gram}}(l) = ?$

$$P(\text{[The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]}) =$$

$$P(\text{[The]})P(\text{[course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13] ....}\,|\,\text{[The]})$$

$$P(\text{[The]})P(\text{[course]}\,|\,\text{[The]})P(\text{[COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are] ....}\,|\,\text{[The][course]})$$

$$P(\text{[The]})P(\text{[course]}\,|\,\text{[The]})P(\text{[COMP8730]}\,|\,\text{[The][course]})P(\text{[is][about][nlp][.] ... There][are] ....}\,|\,\text{[The][course] [COMP8730]})$$

$$P(\text{[The]})P(\text{[course]}\,|\,\text{[The]})P(\text{[COMP8730]}\,|\,\text{[The][course]})\ldots P(\text{[name]}\,|\,\text{[The][course] [COMP8730] .... [’s]})\ldots P(\text{[.]}\,|\,\text{[The] ... [class]})$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

$$M1 = |\text{token}|\text{-gram model} = 22\text{-gram model} \longrightarrow P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$$

$$M2 = |\text{vocab}|\text{-gram model} = 16\text{-gram model} \longrightarrow P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$$

$$P([\text{ The }]) \ldots P([\text{ are }] \mid \underbrace{[\text{ The }][\text{ course }]\ [\text{ COMP8730 }] \ldots [\text{ There }]}_{\text{Only the last 15 words}}) \ldots P([\text{ . }] \mid \underbrace{\cancel{[\text{ The }][\text{ course }][\text{ COMP8730 }]} \ldots}_{\text{cannot be considered in 16-gram model}} \underbrace{[\text{ . }]\ [\text{ The }] \ldots [\text{ class }]}_{\text{Only the last 15 words}})$$

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$

M3 = 3-gram model = → $P_{3\text{-gram}}(l) = \mathcal{L}_{3\text{-gram}}(l) = ?$

$$P([\text{ The }])P([\text{ course }] \mid [\text{ The }])P([\text{ COMP8730 }] \mid [\text{ The }][\text{course}]) \ldots P([\text{ are }] \mid \cancel{\ldots}\, [\text{ There }][\text{ is }]) \ldots P([\text{ . }] \mid \ldots [\text{ the }][\text{ class }])$$

cannot be considered in 3-gram model

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{\text{22-gram}}(l) = \mathcal{L}_{\text{22-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{\text{16-gram}}(l) = \mathcal{L}_{\text{16-gram}}(l) = ?$

M3 = 3-gram model = → $P_{\text{3-gram}}(l) = \mathcal{L}_{\text{3-gram}}(l) = ?$

M4 = 2-gram model = → $P_{\text{2-gram}}(l) = \mathcal{L}_{\text{2-gram}}(l) = ?$

$$P(\text{[The]})\,P(\text{[course]} \mid \text{[The]})\,P(\text{[COMP8730]} \mid \text{[The][course]}) \ldots P(\text{[are]} \mid \text{... [is]}) \ldots P(\text{[.]} \mid \text{... [class]})$$

cannot be considered in 2-gram model

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l) = \mathcal{L}_{22\text{-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model → $P_{16\text{-gram}}(l) = \mathcal{L}_{16\text{-gram}}(l) = ?$

M3 = 3-gram model = → $P_{3\text{-gram}}(l) = \mathcal{L}_{3\text{-gram}}(l) = ?$

M4 = 2-gram model = → $P_{2\text{-gram}}(l) = \mathcal{L}_{2\text{-gram}}(l) = ?$

M5 = 1-gram model = → $P_{1\text{-gram}}(l) = \mathcal{L}_{1\text{-gram}}(l) = ?$

$P(_{\text{[The]}})P(_{\text{[course]}})P(_{\text{[COMP8730]}}) \ldots P(_{\text{[are]}}) \ldots P(_{\text{[.]}})$ No History!

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

$\underline{wwwwww} = P(w)\,P(w \mid w)\,P(w \mid ww) \cdots \neq P(w)P(w) \cdots P(w)$

M1 = |token|-gram model = 22-gram model ⟶ $P_{\text{22-gram}}(l) = \mathcal{L}_{\text{22-gram}}(l) = 1$

M2 = |vocab|-gram model = 16-gram model ⟶ $P_{\text{16-gram}}(l) = \mathcal{L}_{\text{16-gram}}(l) = ?$

M3 = 3-gram model = ⟶ $P_{\text{3-gram}}(l) = \mathcal{L}_{\text{3-gram}}(l) = ?$

M4 = 2-gram model = ⟶ $P_{\text{2-gram}}(l) = \mathcal{L}_{\text{2-gram}}(l) = ?$ $(P(w))^n$

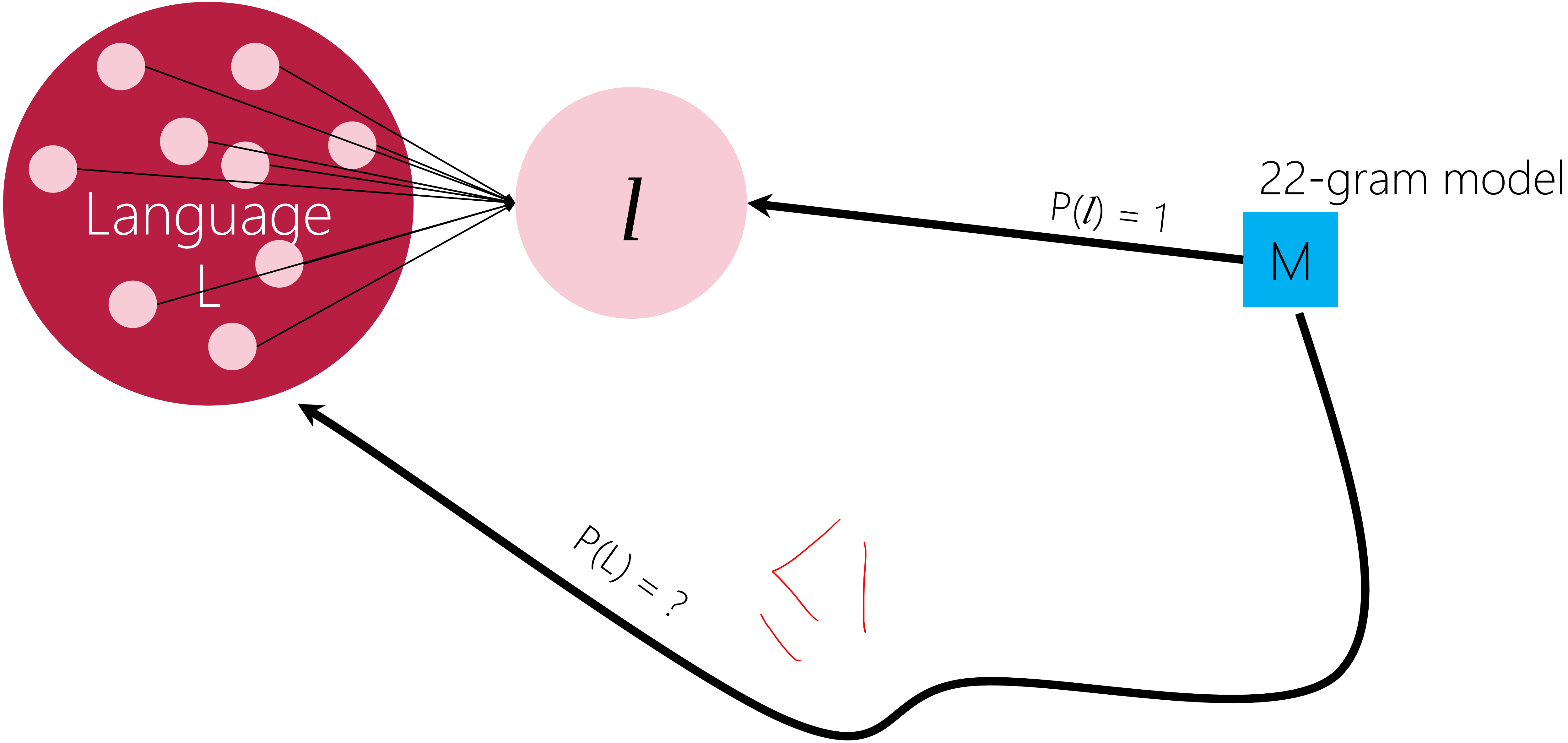
M5 = 1-gram model = ⟶ $P_{\text{1-gram}}(l) = \mathcal{L}_{\text{1-gram}}(l) = ?$

$P(w) = 1$

$\frac{|ww}{\#w}$

$P(w \mid w) = \frac{\#ww}{\#w} = \frac{4}{5}$ $\#1$

Do you think $\mathcal{L}_{\text{M-\{2..5\}}}(l) \geq \mathcal{L}_{\text{22-gram}}(l)$?



Language L

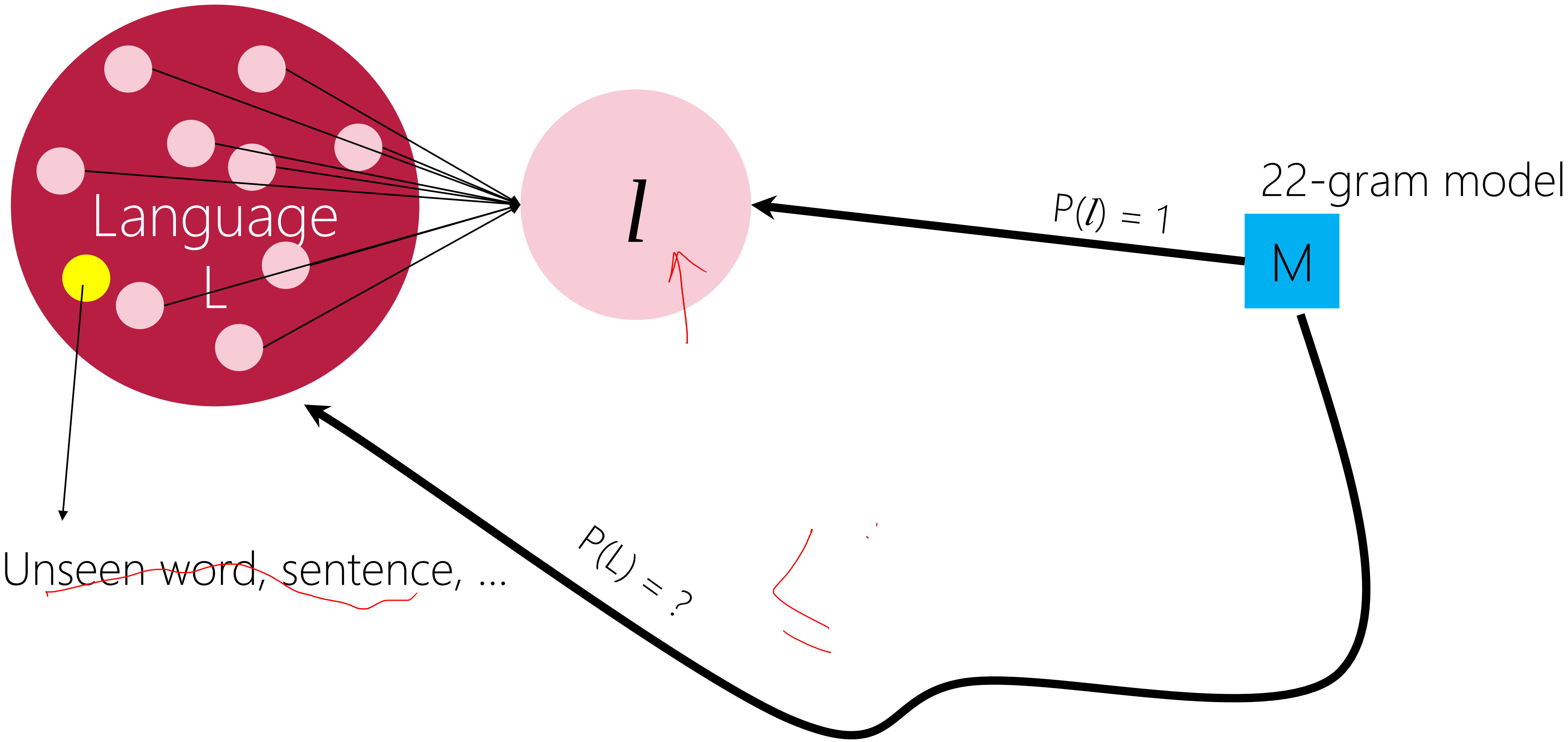
l

$P(l) = 1$

22-gram model

M

$P(L) = ?$



Language L

l

22-gram model

M

$P(l) = 1$

$P(L) = ?$

Unseen word, sentence, ...

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

l': [Hossein][is][the][name][.]

p(hossein) x p(is | ~~ho~~) p(th | hon is)

M1 = |token|-gram model = 22-gram model → $P_{22\text{-gram}}(l') = \mathcal{L}_{22\text{-gram}}(l')$ = ?

M2 = |vocab|-gram model = 16-gram model → $P_{16\text{-gram}}(l') = \mathcal{L}_{16\text{-gram}}(l')$ = ?

M3 = 3-gram model = → $P_{3\text{-gram}}(l) = \mathcal{L}_{3\text{-gram}}(l')$ = ?

M4 = 2-gram model = → $P_{2\text{-gram}}(l) = \mathcal{L}_{2\text{-gram}}(l')$ = ?

M5 = 1-gram model = → $P_{1\text{-gram}}(l) = \mathcal{L}_{1\text{-gram}}(l')$ = ?

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor][’s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

l': [Hossein][is][the][name][.]

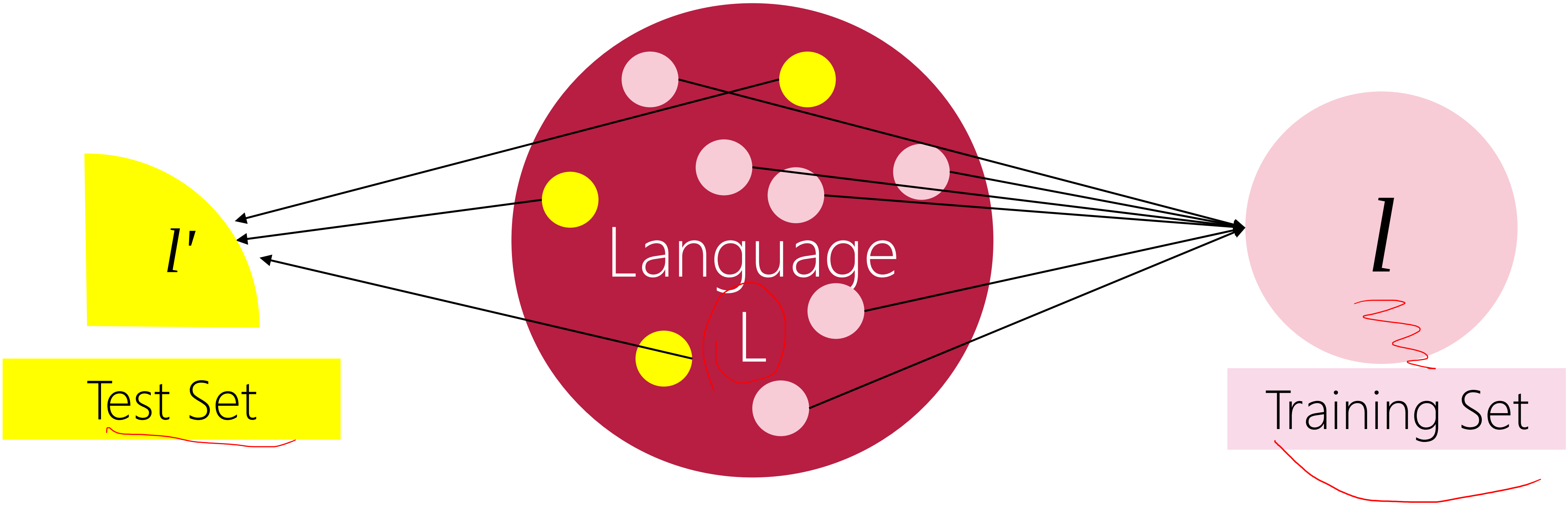
M1 = |token|-gram model = 22-gram model $\longrightarrow P_{\text{22-gram}}(l') = \mathcal{L}_{\text{22-gram}}(l') = 0$

M2 = |vocab|-gram model = 16-gram model $\longrightarrow P_{\text{16-gram}}(l') = \mathcal{L}_{\text{16-gram}}(l') = 0$

M3 = 3-gram model = $\longrightarrow P_{\text{3-gram}}(l) = \mathcal{L}_{\text{3-gram}}(l') = 0$

M4 = 2-gram model = $\longrightarrow P_{\text{2-gram}}(l) = \mathcal{L}_{\text{2-gram}}(l') = 0$

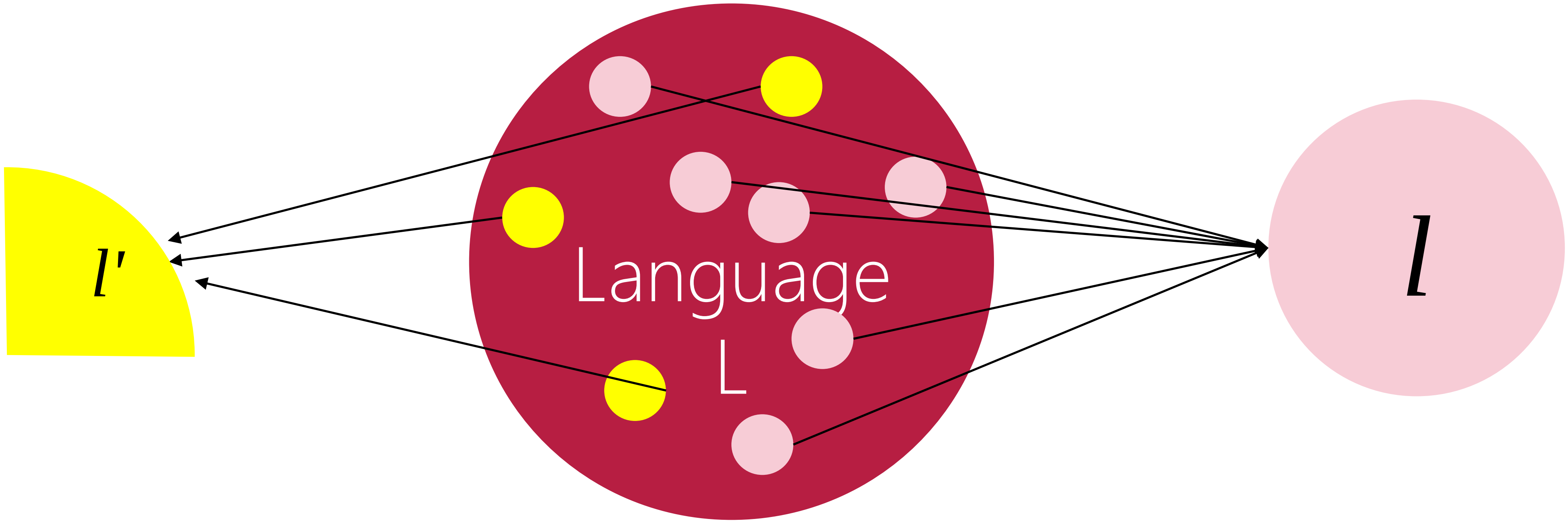
M5 = 1-gram model = $\longrightarrow P_{\text{1-gram}}(l) = \mathcal{L}_{\text{1-gram}}(l') =$ Nonzero!



M is trained on l and does not know anything about l'

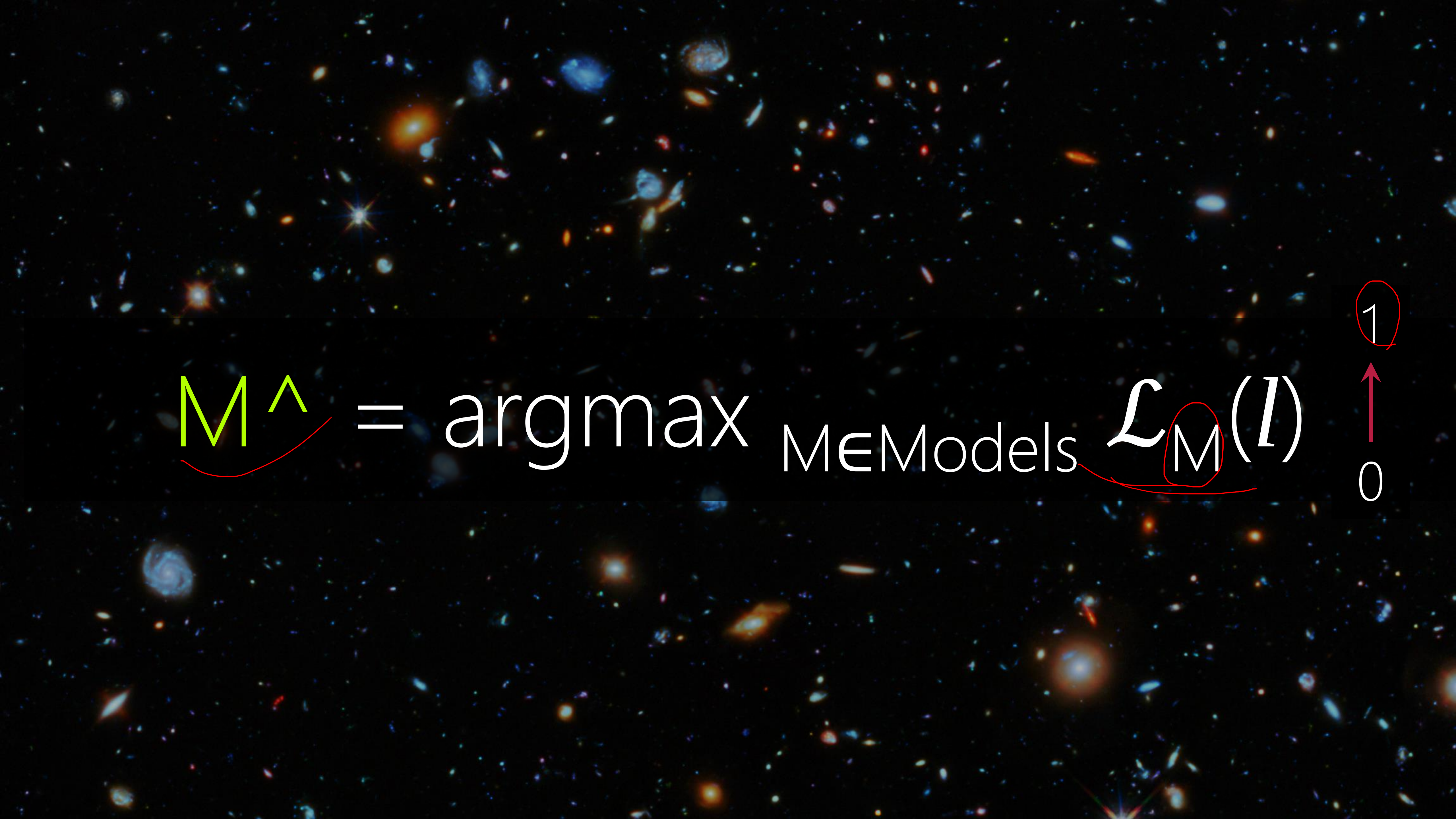
	$P_M(l)=\mathcal{L}_M(l)$	$P_M(l')=\mathcal{L}_M(l')$
M10	High	High
M20	High	Low
M30	Low	High
M40	Low	Low

he~~v~~er rarely

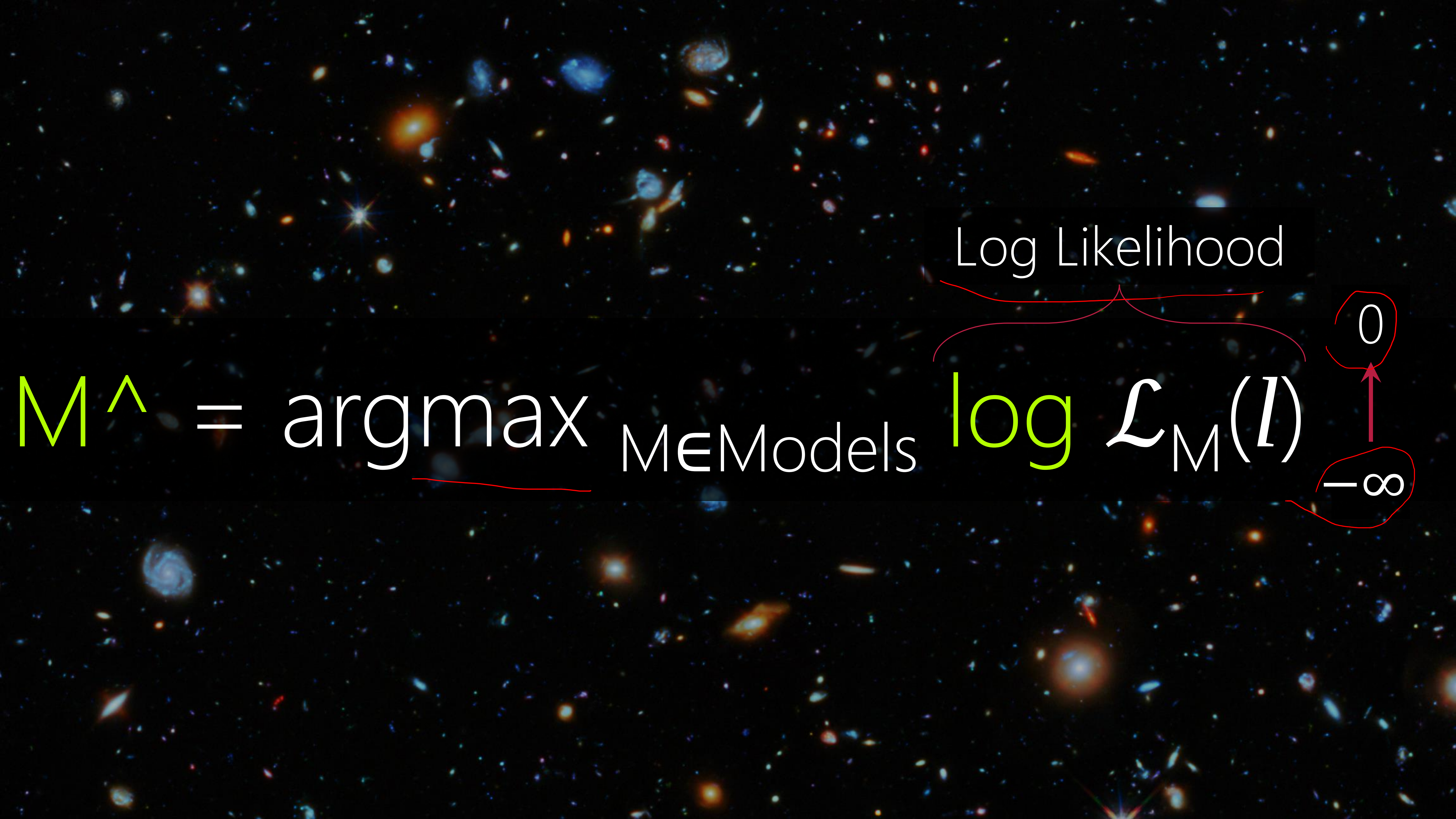


M is trained on l and does not know anything about l'

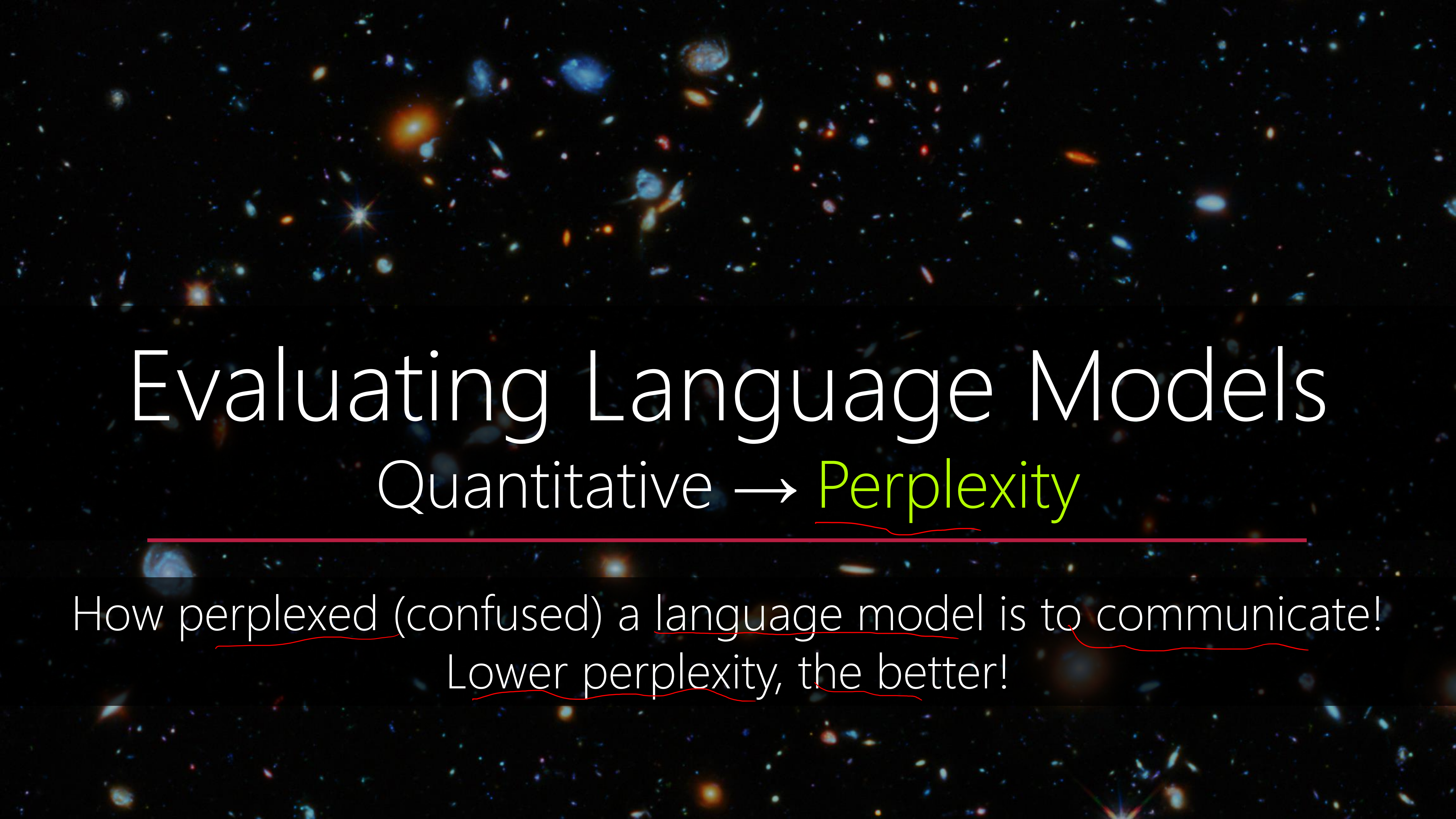
	$P_M(l)=\mathcal{L}_M(l)$	$P_M(l')=\mathcal{L}_M(l')$	
M10	High	High	
M20	High	Low	overfitting
M30	Low	High	?
M40	Low	Low	underfitting



$$\underline{M^{\wedge}} = \text{argmax}_{M\in\text{Models}}\ \underline{\mathcal{L}_{\boxed{M}}}(I)\quad \underset{0}{\overset{\boxed{1}}{\uparrow}}$$



$$M\,\hat{}\; = \underline{\text{argmax}}_{\,M\in\text{Models}}\ \overbrace{\log \mathcal{L}_{M}(l)}^{\text{Log Likelihood}}\quad \underset{-\infty}{\overset{0}{\uparrow}}$$



Evaluating Language Models

Quantitative → Perplexity

How perplexed (confused) a language model is to communicate!

Lower perplexity, the better!

Perplexity

$$x^{-\frac{1}{n}} = \frac{1}{x^{1/n}} = \frac{1}{\sqrt[n]{x}}$$

$$x^{-n} = \frac{1}{x^{n}} = \sqrt[n]{\frac{1}{x}}$$

$$PP_{M}(l') = \mathcal{L}_{M}(l')^{\frac{-1}{|l'|}} = \sqrt[|l'|]{\frac{1}{\mathcal{L}(l')}}$$

M is trained on *l* and test on *l′*

Higher $\mathcal{L}_{M}(l')$, lower perplexity, the better!

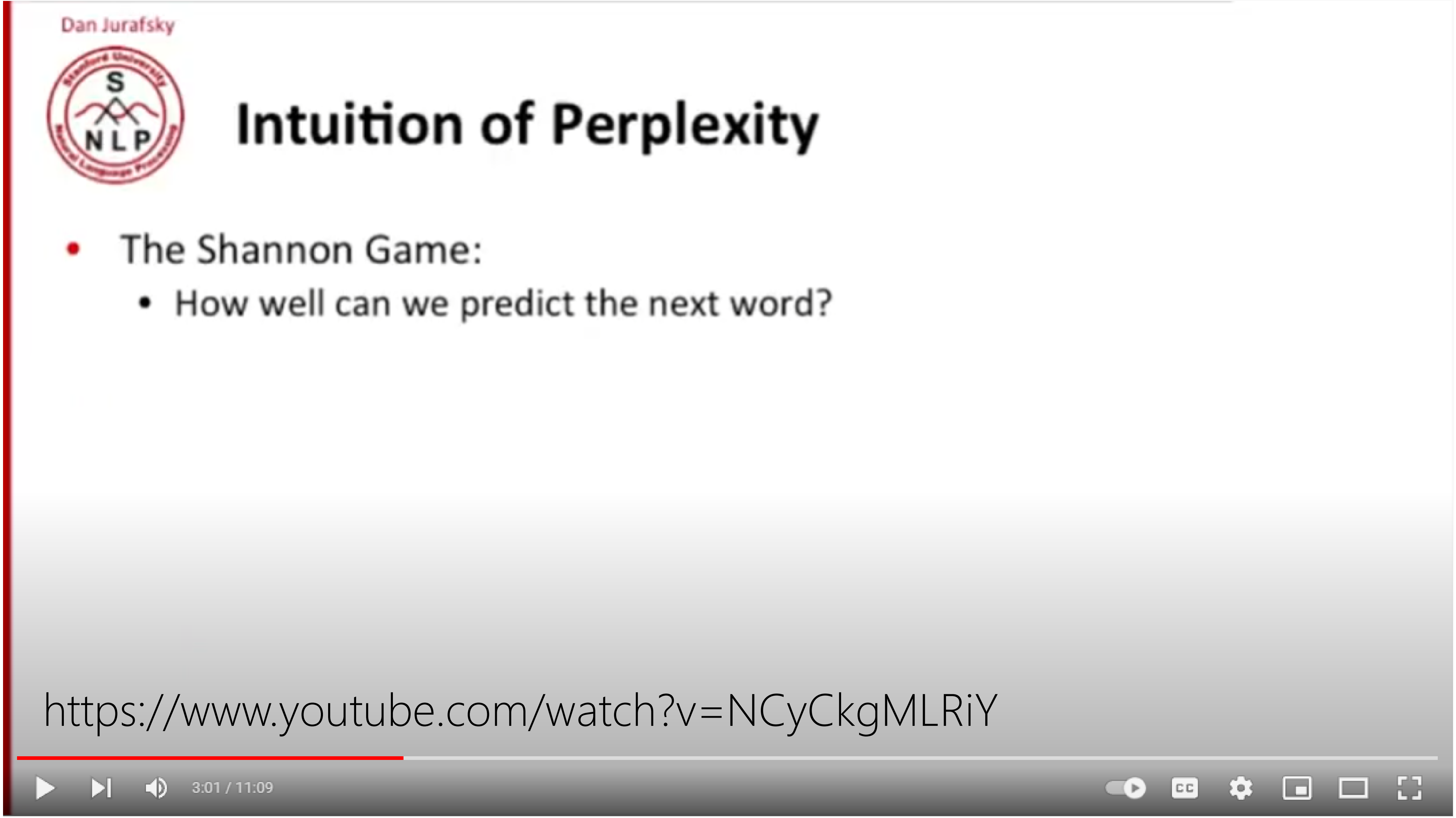
Perplexity

Unigram approx.: $\sqrt[|l'|]{\frac{1}{\mathcal{L}_{\mathrm{M}}(l')}} = \sqrt[|l'|]{\frac{1}{\prod_{k=1}^{|l'|} \mathrm{P}(w_k)}}$; $w_i \in l'$

$= \sqrt[|l'|]{\frac{1}{\mathrm{P}(w_k)^{|l'|}}}$; if uniform distribution over words

$= \frac{1}{\mathrm{P}(w_k)} = \frac{1}{\frac{1}{|V|}} = |\mathrm{V}|$ the size of vocabs

If LM wants to select a word, it is perplexed in the factor of |V|



Dan Jurafsky

Intuition of Perplexity

- The Shannon Game:
  - How well can we predict the next word?

https://www.youtube.com/watch?v=NCyCkgMLRiY

Lecture 14 — Evaluation and Perplexity — [NLP || Dan Jurafsky || Stanford University]

Perplexity

Unigram approx.: $\sqrt[|l'|]{\frac{1}{\mathcal{L}_{\mathrm{M}}(l')}} = \sqrt[|l'|]{\frac{1}{\prod_{k=1}^{|l'|} \mathrm{P}(w_k)}}$; $w_i \in l'$

model

Bigram approx.: $\sqrt[|l'|]{\frac{1}{\mathcal{L}_{\mathrm{M}}(l')}} = \sqrt[|l'|]{\frac{1}{\prod_{k=1}^{|l'|} \mathrm{P}(w_i|w_{i-1})}}$; $w_{i-1}w_i \in l'$

Trigram approx.: $\sqrt[|l'|]{\frac{1}{\mathcal{L}_{\mathrm{M}}(l')}} = \sqrt[|l'|]{\frac{1}{\prod_{k=1}^{|l'|} \mathrm{P}(w_i|w_{i-2}w_{i-1})}}$; $w_{i-2}w_{i-1}w_i \in l'$

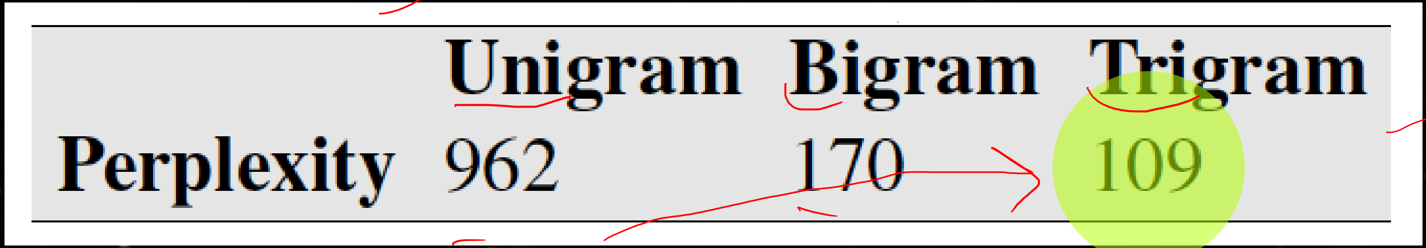
Perplexity

l: Wall Street Journal

Size: 38 million words

Vocab (Types): 19,979

l': 1.5 million words



	Unigram	Bigram	Trigram
Perplexity	962	170	109

P(1) P(2)

22/e

130

P(1,2)

Is 4-Gram better?

Zeros!

Likelihood for a Language Model

l: [The][course][COMP8730][is][about][nlp][.][The][instructor]['s][name][is][Hossein][.][There][are][13][students][in][the][class][.]

l': [Hossein][is][the][name] [of] [instructor] [.]

$$\text{M1} = |\text{token}|\text{-gram model} = \text{22-gram model} \longrightarrow \text{P}_{\text{22-gram}}(l') = \mathcal{L}_{\text{22-gram}}(l') = 0$$

$$\text{M2} = |\text{vocab}|\text{-gram model} = \text{16-gram model} \longrightarrow \text{P}_{\text{16-gram}}(l') = \mathcal{L}_{\text{16-gram}}(l') = 0$$

$$\text{M3} = \text{3-gram model} = \longrightarrow \text{P}_{\text{3-gram}}(l) = \mathcal{L}_{\text{3-gram}}(l') = 0$$

$$\text{M4} = \text{2-gram model} = \longrightarrow \text{P}_{\text{2-gram}}(l) = \mathcal{L}_{\text{2-gram}}(l') = 0$$

$$\text{M5} = \text{1-gram model} = \longrightarrow \text{P}_{\text{1-gram}}(l) = \mathcal{L}_{\text{1-gram}}(l') = \boxed{0}!\ \text{Why?}$$

$p(of) = 0$

Zeros!

Not ~~all unigrams are available in training~~ set! E.g., [of]

Not all bigrams are available in training set! E.g., [Hossein][is]

Not all trigrams are available in training set! …

Zeros!

1) Vocabulary + <UNK>
2) Train Vocabulary + Learn Unseen Tokens (Subwords)
3) Smoothing

BPE

Zeros! <UNK>

learn the stat of unseen tokens

1) Pick a dictionary D
2) From $w \in l$ such that $w \notin D$, (oov) replace it with <UNK>
3) Train model
4) At test, from $w \in l'$, if $w \notin l$ (unseen), replace it with <UNK>

Zeros! <UNK>

learn the stat of unseen tokens

Hloss => 1

D: [The][course][is][about][instructor][name][There][are][13][students][in][class]

l: ['The][course][<UNK>][is][about][<UNK>][<UNK>][The][instructor][<UNK>][name][is][<UNK>][<UNK>][There][are][13][students][in][the][class][.]

l': [<UNK>][is][the][name][<UNK>][<UNK>][<UNK>][<UNK>]

M1 = 22-gram model → $P(l') = \mathcal{L}_{22\text{-gram}}(l') = 0$

M2 = 16-gram model → $P(l') = \mathcal{L}_{16\text{-gram}}(l') = 0$

M3 = 3-gram model = $P(l') = \mathcal{L}_{3\text{-gram}}(l') = \underline{0}$

M4 = 2-gram model = $P(l') = \mathcal{L}_{2\text{-gram}}(l') = \underline{0}$

M5 = 1-gram model = $\underline{P(l') = \mathcal{L}_{1\text{-gram}}(l') =}$ Nonzero! (Why?)

Zeros! <UNK>

learn the stat of unseen tokens

D: [The][is]

l: ['The][<UNK>][<UNK>][is][<UNK>][<UNK>][<UNK>][The][<UNK>][<UNK>][<UNK>][is][<UNK>][<UNK>][<UNK>][<UNK>]...[<UNK>][the][<UNK>][<UNK>]

l': [<UNK>][is][the][<UNK>][<UNK>][<UNK>][<UNK>][<UNK>]

M1 = 22-gram model → $P(l') = \mathcal{L}_{22\text{-gram}}(l') = 0$

M2 = 16-gram model → $P(l') = \mathcal{L}_{16\text{-gram}}(l') = 0$

M3 = 3-gram model = $P(l') = \mathcal{L}_{3\text{-gram}}(l') = 0$

M4 = 2-gram model = $P(l') = \mathcal{L}_{2\text{-gram}}(l')$ = Nonzero!

M5 = 1-gram model = $P(l') = \mathcal{L}_{1\text{-gram}}(l')$ = Nonzero!

Zeros! <UNK>

learn the stat of unseen tokens

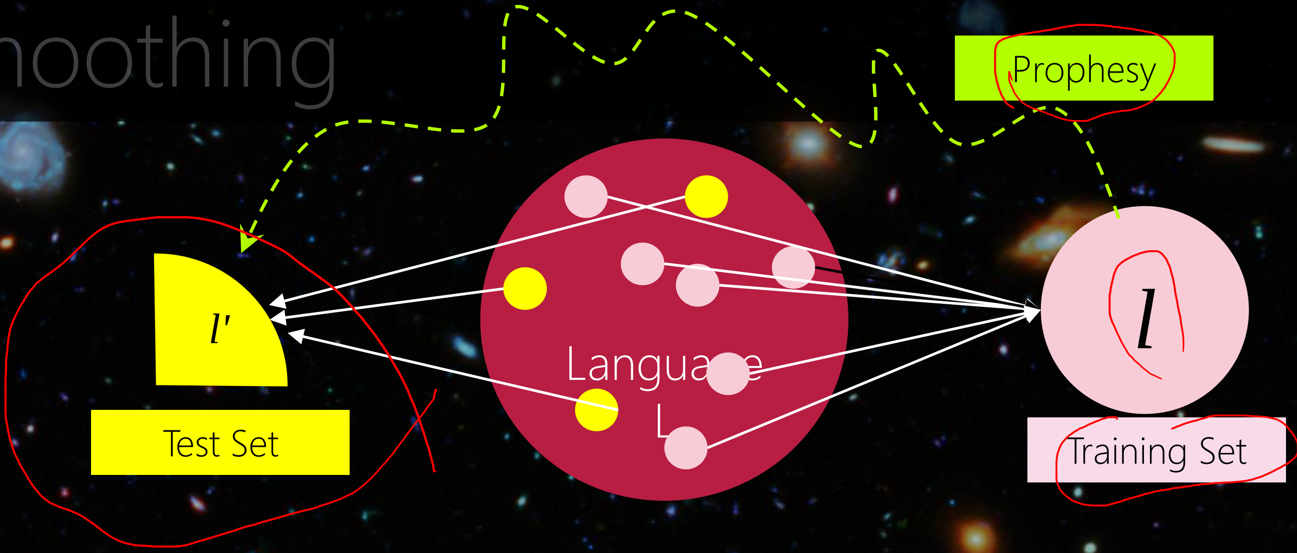
All the model should generate is stream of <UNK>s!

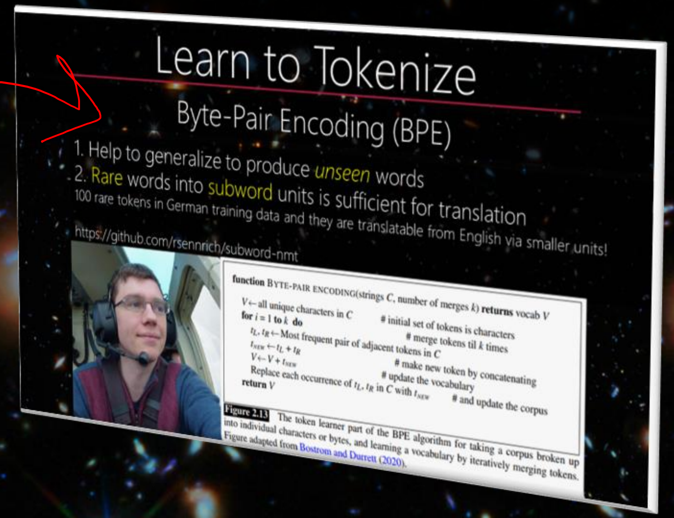
Pick a S~~m~~all Dictionary

Gives higher probability (lower perplexity) at test

Zeros!

1) Vocabulary + <UNK>
2) Train Vocabulary + Learn Unseen Tokens (Subwords)
3) Smoothing





Learn to Tokenize

Byte-Pair Encoding (BPE)

1. Help to generalize to produce *unseen* words
2. Rare words into subword units is sufficient for translation

100 rare tokens in German training data and they are translatable from English via smaller units!

https://github.com/rsennrich/subword-nmt

```
function BYTE-PAIR ENCODING(strings C, number of merges k) returns vocab V
  V ← all unique characters in C            # initial set of tokens is characters
  for i = 1 to k do                         # merge tokens til k times
    t_L, t_R ← Most frequent pair of adjacent tokens in C
    t_NEW ← t_L + t_R                       # make new token by concatenating
    V ← V + t_NEW                           # update the vocabulary
    Replace each occurrence of t_L, t_R in C with t_NEW    # and update the corpus
  return V
```

Figure 2.13 The token learner part of the BPE algorithm for taking a corpus broken up into individual characters or bytes, and learning a vocabulary by iteratively merging tokens. Figure adapted from Bostrom and Durrett (2020).

Zeros!

1) Vocabulary + <UNK>
2) Train Vocabulary + Learn Unseen Tokens (Subwords)
3) Smoothing
   A. Add-1 (Laplace) or Add-k, → k={1,2,...}
   B. Backoff
   C. Interpolation
   D. ...

Zeros! Add-k

Add *k* unit to all counts, so zero entries become *k*

1 = apple

Add-1 is called Laplace

Unigram model: $$P(w_i) = \frac{\#w_i + k}{|\text{tokens}| + k \times |\text{vocabs}|}$$

(the man) 1

(man the) 1

Bigram model: $$P(w_i|w_{i-1}) = \frac{\#(w_{i-1}w_i) + k}{\#(w_{i-1}) + k \times ?}$$

Trigram model: $$P(w_i|w_{i-2}w_{i-1}) = \frac{\#(w_{i-2}w_{i-1}w_i) + k}{\#(w_{i-2}w_{i-1}) + k \times ?}$$

Zeros! Backoff

if n-Gram have not seen, try (n-1)-Gram

Trigram model: $P(w_i|w_{i-2}w_{i-1}) = \frac{\#(w_{i-2}w_{i-1}w_i)}{\#(w_{i-2}w_{i-1})} = 0$

⬇

Bigram model: $P(w_i|w_{i-1}) = \frac{\#(w_{i-1}w_i)}{\#(w_{i-1})} = 0$

⬇

Unigram model: $P(w_i) = \frac{\#w_i}{|tokens|}$

Zeros! Interpolation

P(n-Gram) is linear interpolation of all (n-i)-Grams: i={1,2,..., n-1}.

4-g

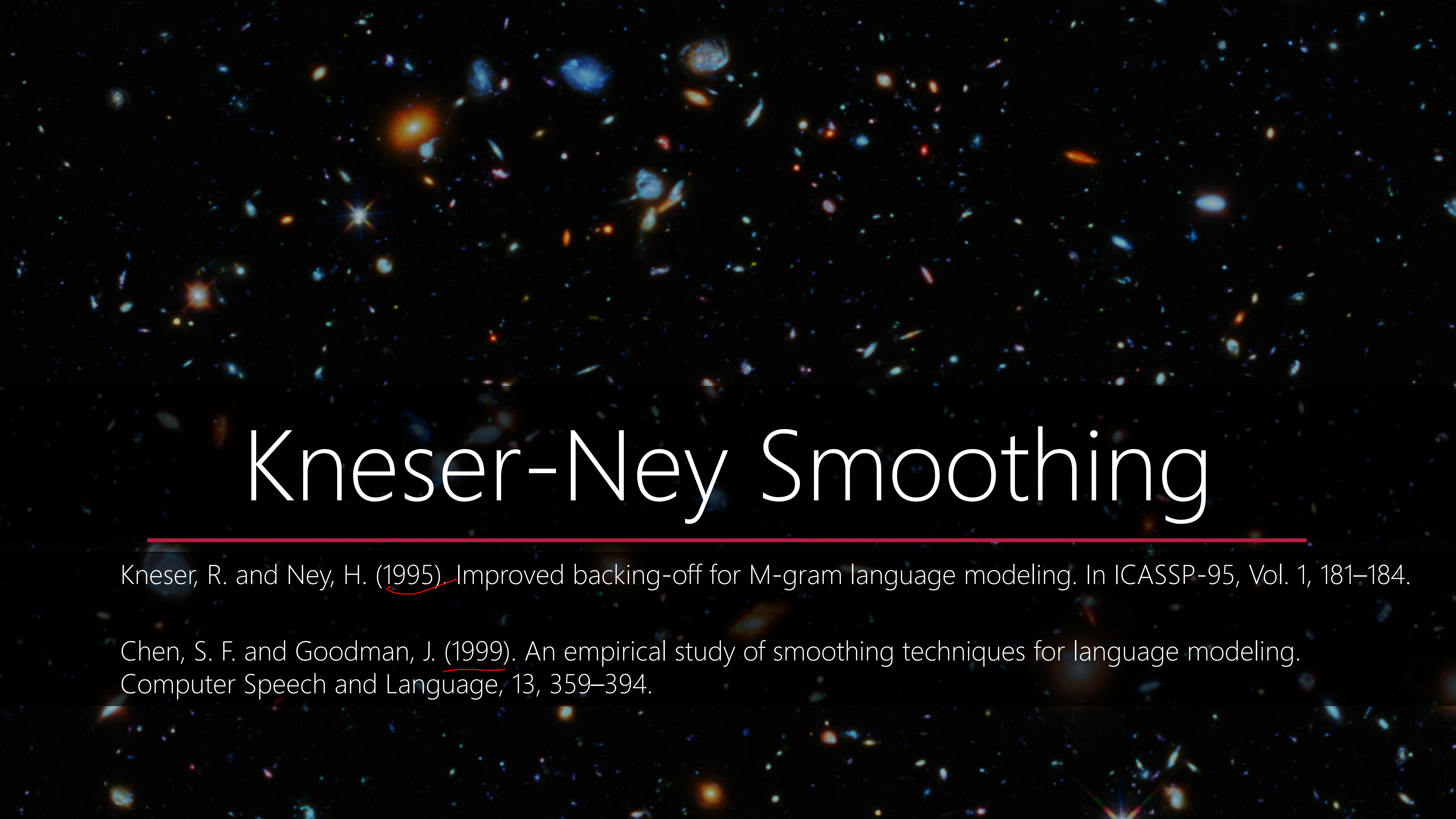
λ_4

Trigram model: $P(w_i|w_{i-2}w_{i-1}) = \lambda_1 \frac{\#(w_{i-2}w_{i-1}w_i)}{\#(w_{i-2}w_{i-1})} +$

Bigram model: $P(w_i|w_{i-1}) = \lambda_2 \frac{\#(w_{i-1}w_i)}{\#(w_{i-1})} +$

Unigram model: $P(w_i) = \lambda_3 \frac{\#w_i}{|\text{tokens}|}$

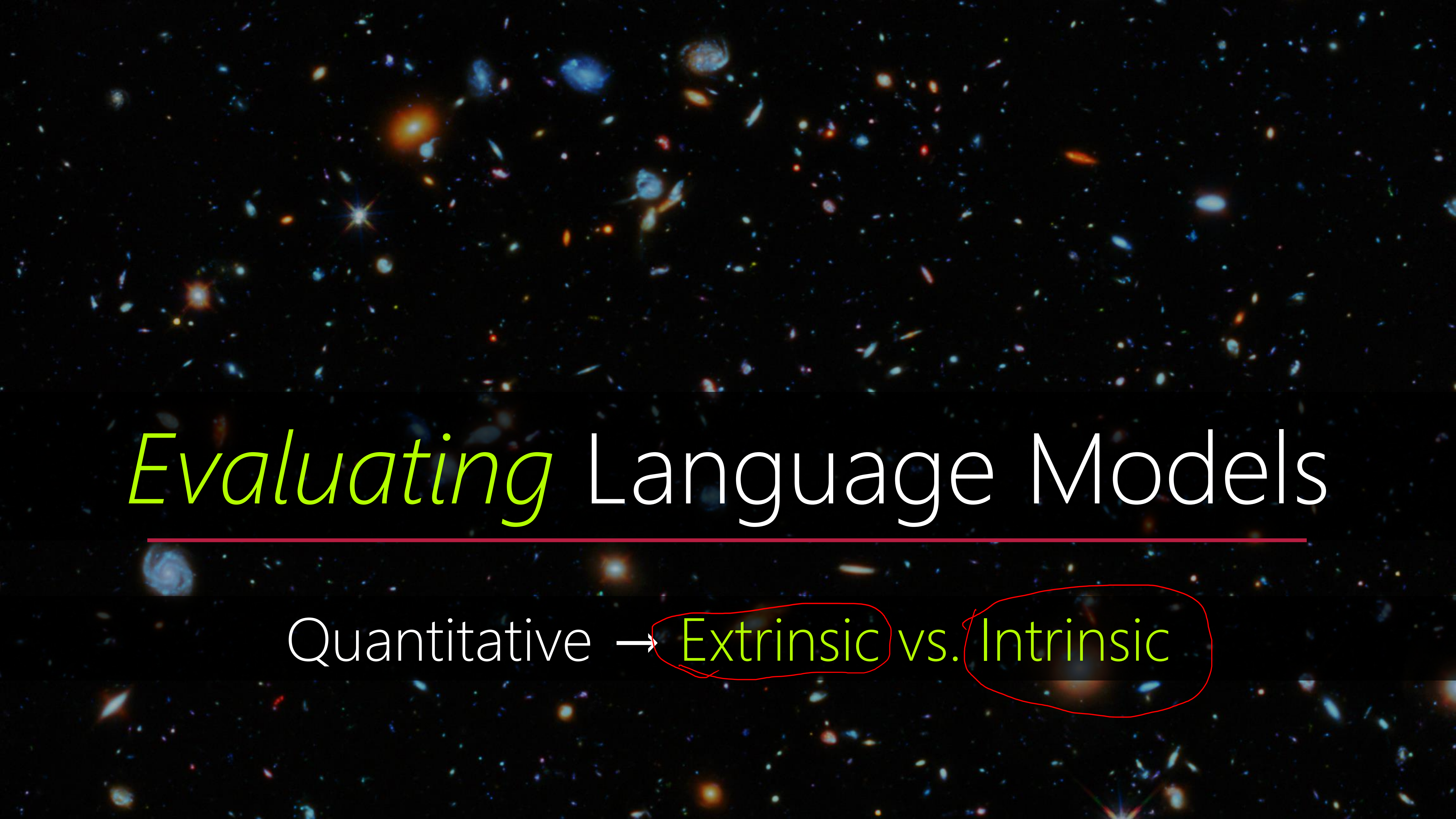
$$\sum \lambda_i = 1$$



Kneser-Ney Smoothing

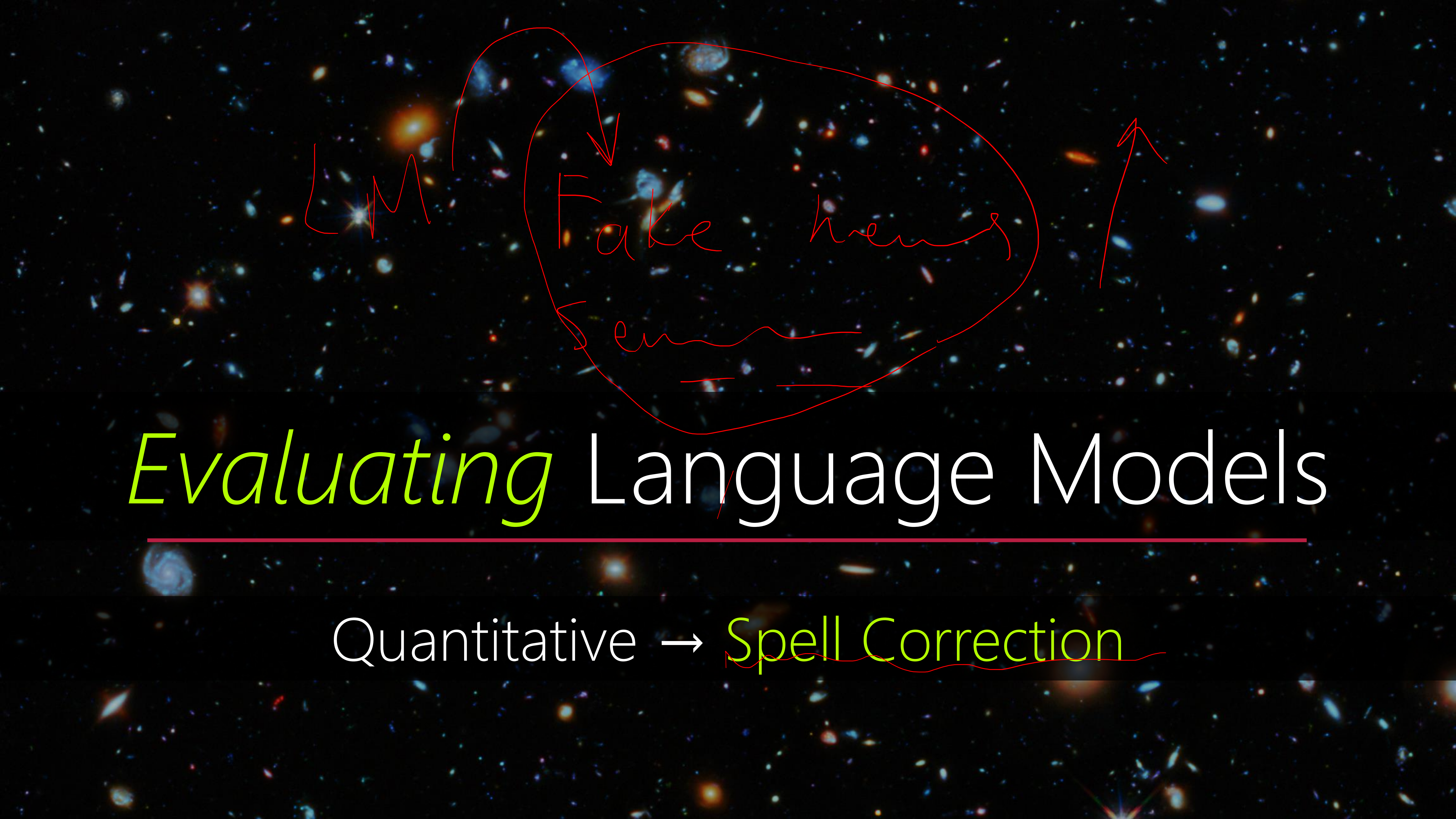
Kneser, R. and Ney, H. (1995). Improved backing-off for M-gram language modeling. In ICASSP-95, Vol. 1, 181–184.

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. Computer Speech and Language, 13, 359–394.



Evaluating Language Models

Quantitative → Extrinsic vs. Intrinsic



LM

Fake news

Sen

Evaluating Language Models

Quantitative → Spell Correction

$$P(w \mid w_{i+1} \dots w_{i+n-2}\, w_{i+n-1}) = \frac{P(wi_{+1} \dots w_{i+n-2}\, w_{i+n-1} w)}{P(wi_{+1} \dots w_{i+n-2}\, w_{i+n-1})}$$

$$= \frac{\#(wi_{+1} \dots w_{i+n-2}\, w_{i+n-1} w)}{\#(wi_{+1} \dots w_{i+n-2}\, w_{i+n-1})}$$

w ∈ voc

top-10

0.6

$$W_{i+1} \cdots W_{i+n-2}\, W_{i+n-1} \longrightarrow \overset{w}{\cancel{W_{i+n}}}$$

More helpful a language model in finding correct spells, the better!

MED

$$P(w \mid w_{i+1} \ldots w_{i+n-2}\, w_{i+n-1}) = \frac{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{P(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

$$= \frac{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1} w)}{\#(wi_{+1} \ldots w_{i+n-2}\, w_{i+n-1})}$$

2-g

3-

$$W_{i+1} \cdots W_{i+n-2}\, W_{i+n-1} \longrightarrow W_{i+n}$$

More helpful a language model in finding correct spells, the better!

Is this judgment correct?