NLP Basic

02 – LANGUAGE MODEL

AI VIET NAM
Nguyen Quoc Thai

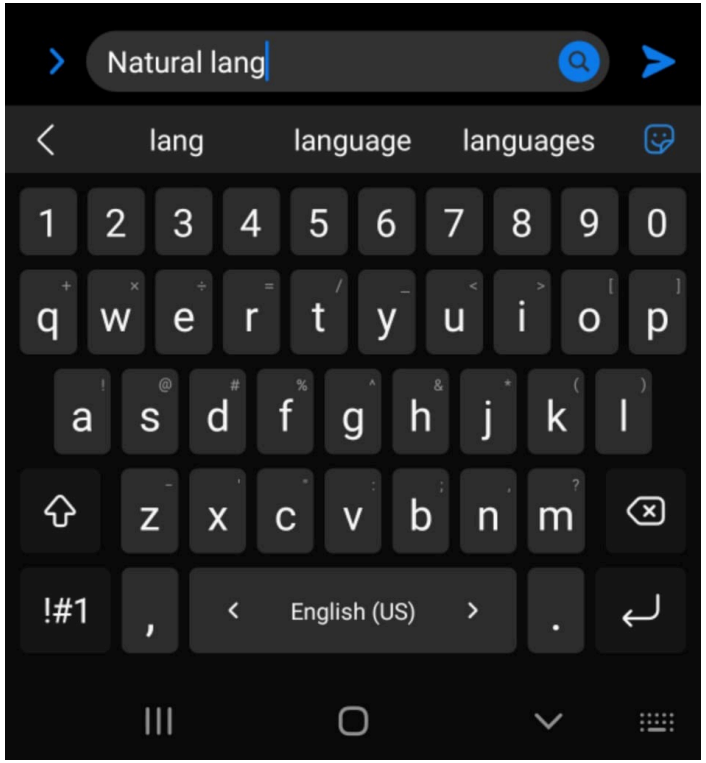
CONTENT

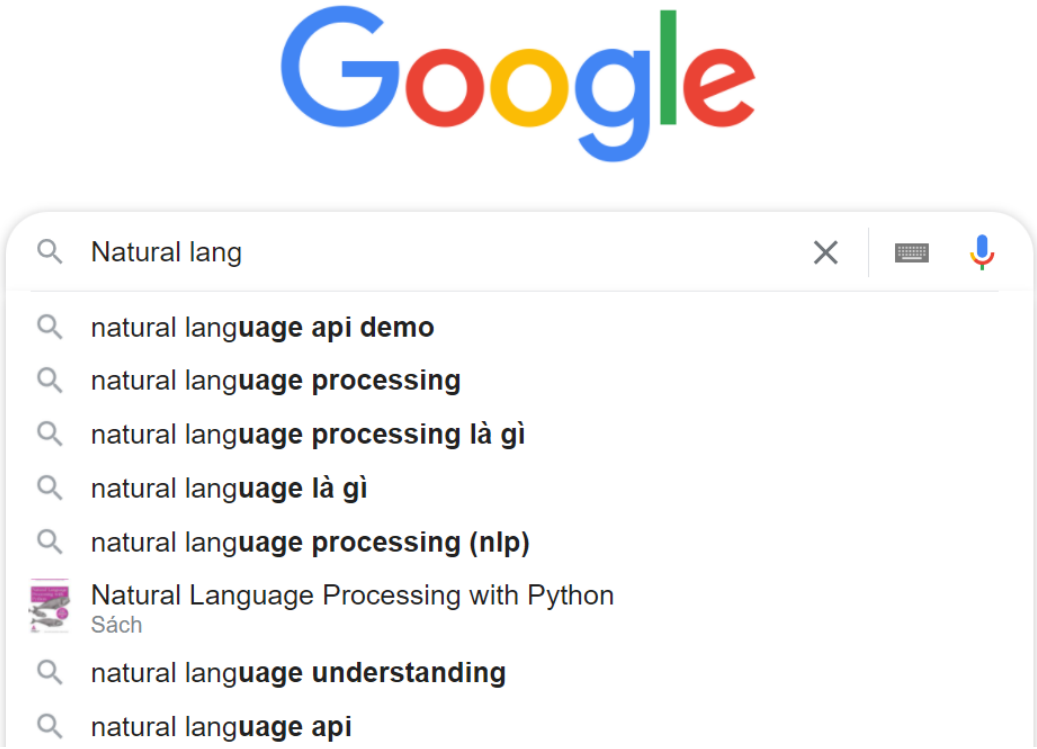
AI VIET NAM
@aivietnam.edu.vn

1 – N-grams Language Model

Language Model

- The task of predicting what word comes next

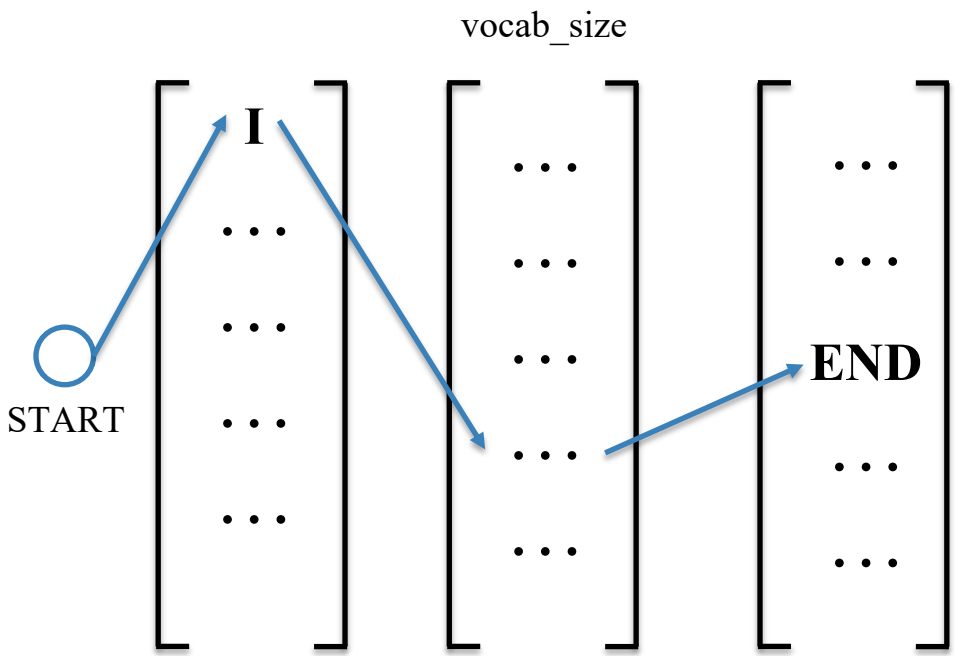


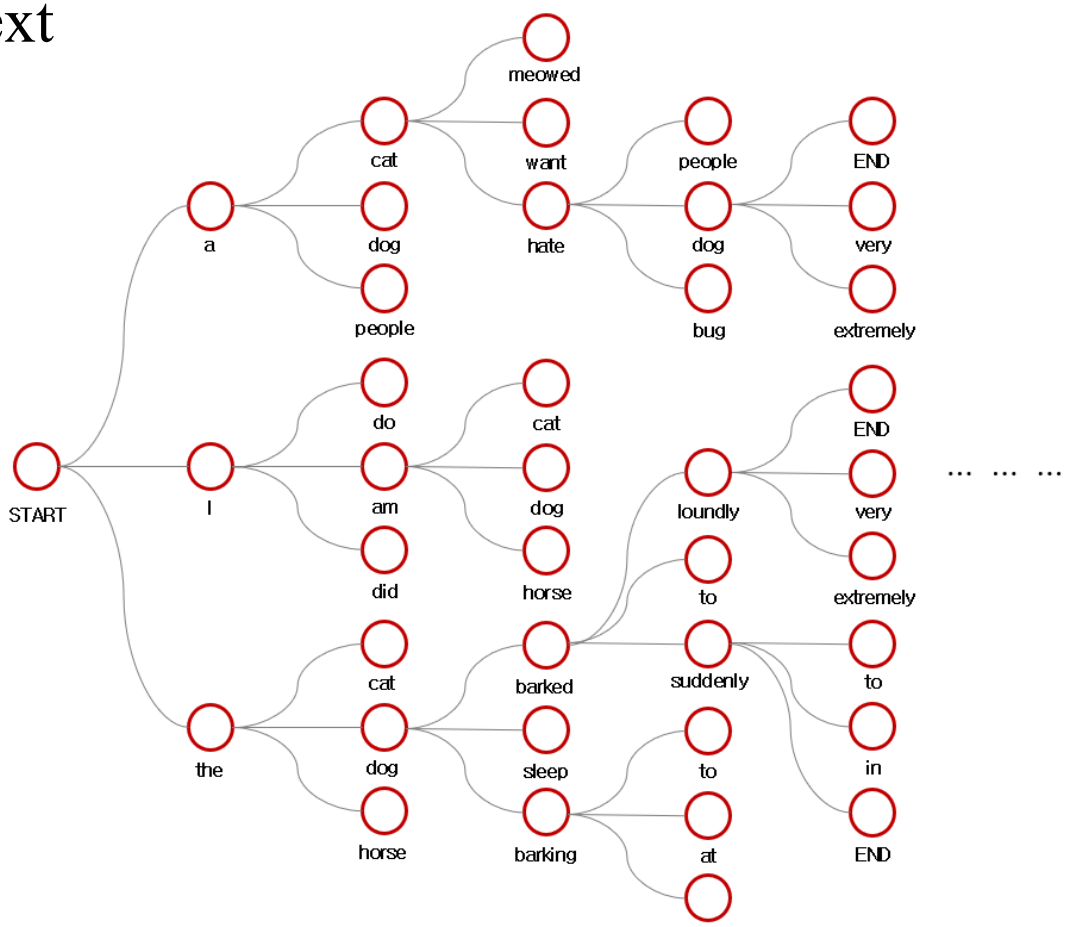


1 – N-grams Language Model

Language Model

- The task of predicting what word comes next



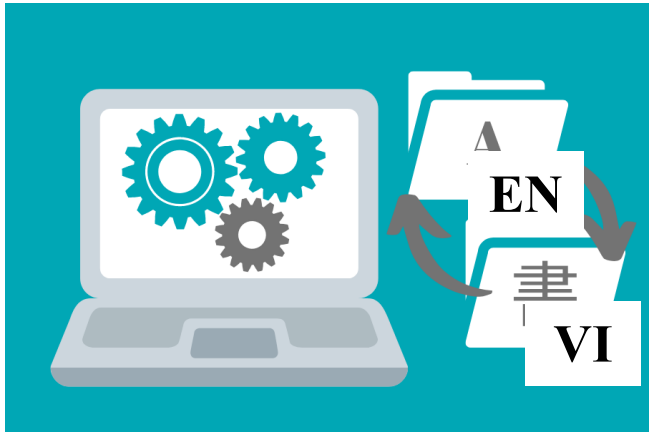


1 – N-grams Language Model

Language Model

- Goal: assign a probability to a sentence

Machine Translation



Input Sentence

"Tôi đi học."

Candidate Sentence

"I go to school."		"I go to work."
P("I go to school.")	>	P("I go to work."

1 – N-grams Language Model

Language Model

- The task of predicting what word comes next

Text Preprocessing → Word-based Tokenization

"I go to school." → "i go to school" → "i,go,to,school"

- Compute the probability of occurrence of the number of words (tokens) in a sentence (sequence)

P("i,go,to,school")

$P(W) = P(w_1, w_2, \ldots, w_n)$

1 – N-grams Language Model

Probabilistic Language Model

- Compute P(W)

$$P(W) = P(w_1, w_2, \ldots, w_n)$$

- Conditional Probability

$$P(B|A) = P(AB)/P(A) \Rightarrow P(AB) = P(A).P(B|A)$$

- The chain rule of probability

$$P(A,B,C,D) = P(A).P(B|A).P(C|AB).P(D|ABC)$$

$$P(W) = P(w_1, w_2, \ldots, w_n) = P(w_1).P(w_2|w_1).P(w_3|w_{1:2})\ldots P(w_n|w_{1:n-1}) = \prod_{k=1}^{n} P(w_k|w_{1:k})$$

1 – N-grams Language Model

Probabilistic Language Model

- Compute P(W)

$$P(W) = P(w_1, w_2, \ldots, w_n) = P(w_1).P(w_2|w_1).P(w_3|w_{1:2})\ldots P(w_n|w_{1:n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_{1:k})$$

- Compute the probability of an upcoming words

"I go to school."

P("i,go,to,school") = P(i).P(go|i).P(to|i,go).P(school|i,go,to)

1 – N-grams Language Model

Probabilistic Language Model

- Compute the probability of an upcoming words

"I go to school."

P("i,go,to,school") = P(i).P(go|i).P(to|i,go).P(school|i,go,to)

- Compute P(w|h)

P(w|h) = P(school|i,go,to)

- w: token as word "school"
- h: history tokens as "i,go,to"

1 – N-grams Language Model

Probabilistic Language Model

- Compute P(w|h)

P(w|h) = P(school|i,go,to)

- w: token as word “school”
- h: history tokens as “i,go,to”

$$P(w|h) = \frac{count(hw)}{count(h)}$$

$$P(school|i,go,to) = \frac{count(i,go,to,school)}{count(i,go,to)}$$

1 – N-grams Language Model

Probabilistic Language Model

- The probability of a word depends only on some previous words

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i | w_{i-N+1:i-1})$$

$$P(w_i | w_{1:i-1}) = P(w_i | w_{i-N+1:i-1})$$

- N-gram model with N = {1, 2,…}

1 – N-grams Language Model

Probabilistic Language Model

- N = 1
- Unigram Model (1-gram)

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i|w_{i-N+1:i-1}) = \prod_{i=1}^{n} P(w_i)$$

P("i,go,to,school") → = P(i).P(go|i).P(to|i,go).P(school|i,go,to)

= P(i).P(go).P(to).P(school)

1 – N-grams Language Model

Probabilistic Language Model

- N = 2
- Bigram Model (2-gram)

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i|w_{i-N+1:i-1}) = \prod_{i=1}^{n} P(w_i|w_{i-1})$$

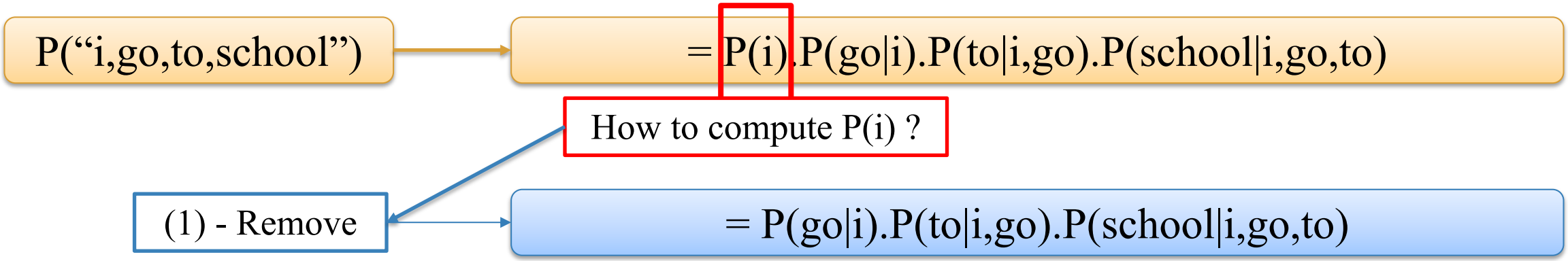
P("i,go,to,school") → = P(i).P(go|i).P(to|i,go).P(school|i,go,to)

How to compute P(i) ?

1 – N-grams Language Model

Probabilistic Language Model

- N = 2
- Bigram Model (2-gram)



1 – N-grams Language Model

Probabilistic Language Model

- N = 2
- Bigram Model (2-gram)

P("i,go,to,school") → = P(i).P(go|i).P(to|i,go).P(school|i,go,to)

How to compute P(i) ?

(1) - Remove

(2) - Padding

Pad = 1

= P(i|<s>).P(go|<s>,i).P(to|<s>,i,go)
.P(school|<s>,i,go,to).P(</s>|<s>,i,go,to,school)

P("<s>,i,go,to,school, </s>")

= P(i|<s>).P(go|i).P(to|go)
.P(school|to).P(</s>|school)

1 – N-grams Language Model

Probabilistic Language Model

- N = 3
- Trigram Model (3-gram)

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i|w_{i-N+1:i-1}) = \prod_{i=1}^{n} P(w_i|w_{i-2:i-1})$$

P("i,go,to,school") → Pad = 2 → P("<s>,<s>,i,go,to,school, </s>,</s>")

1 – N-grams Language Model

Probabilistic Language Model

- N = 3
- Trigram Model (3-gram)

P("i,go,to,school") → Pad = 2 → P("<s>,<s>,i,go,to,school, </s>,</s>")

= P(i|<s>,<s>).P(go|<s>,<s>,i).P(to|<s>,<s>,i,go).P(school|<s>,<s>,i,go,to)
.P(</s>|<s>,<s>,i,go,to,school). P(</s>|<s>,<s>,i,go,to,school,</s>)

= P(i|<s>,<s>).P(go|<s>,i).P(to|i,go).P(school|go,to)
.P(</s>|to,school). P(</s>|school,</s>)

1 – N-grams Language Model

Estimating N-gram Probability

- N-gram Probability:

$$P(w_i|w_{i-N+1:i-1}) = \frac{\text{count}(w_{i-N+1:i-1}, w_i)}{\text{count}(w_{i-N+1:i-1})}$$

- Bigram Probability

$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

1 – N-grams Language Model

Estimating N-gram Probability

- Example:

$$P(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

<s> tôi đang học </s>
<s> tôi đang học lớp nlp </s>
<s> lớp nlp có vẻ hơi vui </s>

P(tôi|<s>) = P(đang|tôi) = P(học|đang) =

P(</s>|học) = P(lớp|học) = P(nlp|lớp) =

- P(tôi, đang, học) =

1 – N-grams Language Model

Estimating N-gram Probability

➢ Example:

$$P(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

<s> tôi đang học </s>
<s> tôi đang học lớp nlp </s>
<s> lớp nlp có vẻ hơi vui </s>

P(tôi|<s>) = 2/3 P(đang|tôi) = 2/2 P(học|đang) = 2/2

P(</s>|học) = 1/2 P(lớp|học) = 1/2 P(nlp|lớp) = 2/2

❖ P(tôi, đang, học) = P(tôi|<s>).P(đang|tôi).P(học|đang).(</s>|học)
= 2/3.2/2.2/2.1/2 = 1/3

1 – N-grams Language Model

Estimating N-gram Probability

➢ "Truyện Kiều – Nguyễn Du":

Trăm năm trong cõi người ta,
Chữ tài chữ mệnh khéo là ghét nhau.
Trải qua một cuộc bể dâu,
Những điều trông thấy mà đau đớn lòng.
Lạ gì bỉ sắc tư phong,
Trời xanh quen thói má hồng đánh ghen.
Cảo thơm lần giở trước đèn,
Phong tình cổ lục còn truyền sử xanh.
Rằng năm Gia Tĩnh triều Minh,
Bốn phương phẳng lặng, hai kinh vững vàng.

1 – N-grams Language Model

Estimating N-gram Probability

➢ "Truyện Kiều – Nguyễn Du":

	<s>	trăm	năm	trong	cõi	người	ta
<s>	0	13	5	23	1	22	1
trăm	0	0	9	0	0	0	0
năm	0	0	4	1	0	1	0
trong	0	0	1	0	3	1	0
cõi	0	0	0	0	0	1	0
người	0	0	0	2	0	2	7
ta	0	0	0	0	0	0	2

1 – N-grams Language Model

Estimating N-gram Probability

➢ "Truyện Kiều – Nguyễn Du":

	<s>	trăm	năm	trong	cõi	người	ta
<s>	0	0.004	0.0015	0.007	0.0003	0.0068	0.0003
trăm	0	0	0.29	0	0	0	0
năm	0	0	0.077	0.019	0	0.019	0
trong	0	0	0.0095	0	0.0285	0.0095	0
cõi	0	0	0	0	0	0.1	0
người	0	0	0	0.009	0	0.009	0.031
ta	0	0	0	0	0	0	0.035

1 – N-grams Language Model

Estimating N-gram Probability

- "Truyện Kiều – Nguyễn Du":

Sentence: "Trăm năm trong cõi người ta"

P(trăm,năm,trong,cõi,người,ta)
= P(trăm|<s>).P(năm|trăm).P(trong|năm).P(cõi|trong)
.P(người|cõi).P(ta|người).P(</s>|ta)
= 0.004 * 0.29 * 0.019 *0.0285*0.1*0.031*0.439 = 8.5e-10

1 – N-grams Language Model

Estimating N-gram Probability

- "Truyện Kiều – Nguyễn Du":

Sentence: "Trăm năm trong cõi người ta"

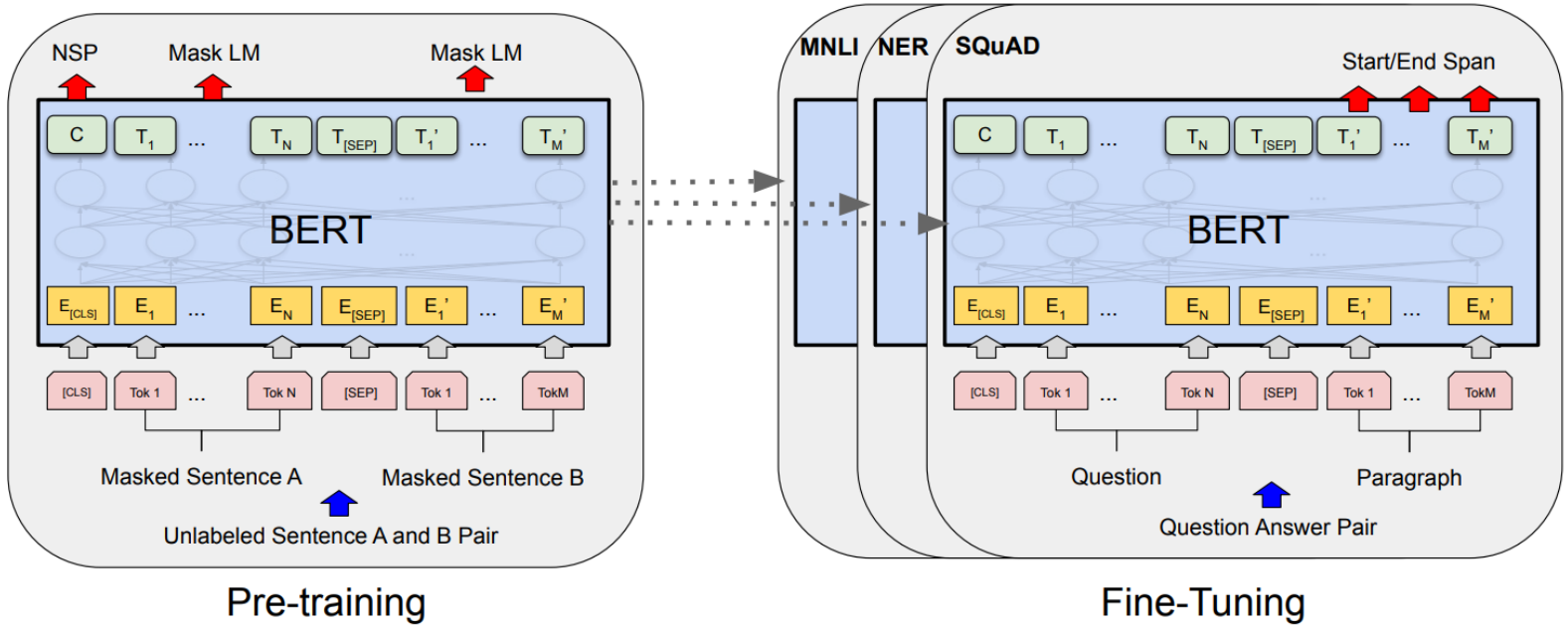
P(trăm,năm,trong,cõi,người,ta)
= P(trăm|<s>).P(năm|trăm).P(trong|năm).P(cõi|trong)
.P(người|cõi).P(ta|người).P(</s>|ta)
= 0.004 * 0.29 * 0.019 *0.0285*0.1*0.031*0.439 = 8.5e-10

Numerical Underflow => Convert to log space

2 – Evaluating Language Model

Extrinsic Evaluation

- Best way to evaluate the performance
- Embedded models into an application (downs task)
- NLP systems end-to-end: very expensive



2 – Evaluating Language Model

Extrinsic Evaluation

Model	Vocabulary Size(K)	Backbone #Params(M)	MNLI-m/mm ACC	SQuAD v2.0 F1/EM
Base models:12 layers,768 hidden size,12 heads				
$\text{BERT}_{\text{base}}$	30	86	84.3/84.7	76.3/73.7
$\text{RoBERTa}_{\text{base}}$	50	86	87.6/-	83.7/80.5
$\text{XLNet}_{\text{base}}$	32	92	86.8/-	-/80.2
$\text{ELECTRA}_{\text{base}}$	30	86	88.8/-	-/80.5
$\text{DeBERTa}_{\text{base}}$	50	100	88.8/88.5	86.2/83.1
$\text{DeBERTaV3}_{\text{base}}$	128	86	**90.6/90.7**	**88.4/85.4**
Small models:6 layers,768 hidden size,12 heads				
$\text{TinyBERT}_{\text{small}}$	30	44	84.5/-	77.7/-
$\text{MiniLMv2}_{\text{small}}$	30	44	87.0/-	81.6/
$\text{BERT}_{\text{small}}$	30	44	81.8/-	73.2/-
$\text{DeBERTaV3}_{\text{small}}$	128	44	**88.2/87.9**	**82.9/80.4**
XSmall models:12 layers,384 hidden size,6 heads				
$\text{MiniLMv2}_{\text{xsmall}}$	30	22	86.9/-	82.3/
$\text{DeBERTaV3}_{\text{xsmall}}$	128	22	**88.1/88.3**	**84.8/82.0**

2 – Evaluating Language Model

Intrinsic Evaluation: Perplexity

- Perplexity (PP) is the inverse probability of the test set, normalized by the number of words
- Lower perplexity => better model
- Test sample: $W = \{w_1, w_2, \ldots, w_n\}$

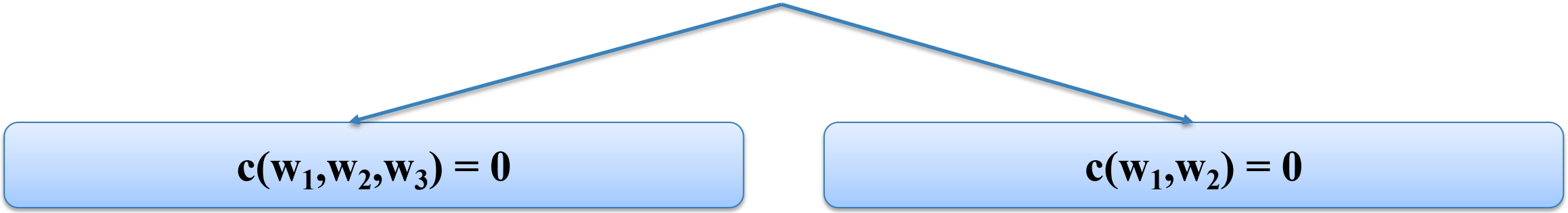
$$PP(W) = P(w_1 w_2 \ldots w_n)^{-\frac{1}{n}}$$

$$PP(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i | w_{1:i-1})}}$$

- Bigram Models: $PP(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i | w_{i-1})}}$

3 – Generalization and Zeros

$$P(w_3|w_1w_2) = \frac{c(w_1, w_2, w_3)}{c(w_1, w_2)}$$



3 – Generalization and Zeros

$$P(w_3|w_1w_2) = \frac{c(w_1, w_2, w_3)}{c(w_1, w_2)}$$

$c(w_1,w_2,w_3) = 0$

$c(w_1,w_2) = 0$

Smoothing

$$P_{Add-1}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$$

Backoff

$P(w_i|w_{i-2:i-1}) \Rightarrow P(w_i|w_{i-1}) \Rightarrow P(w_i)$

Interpolation

$$\hat{P}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P(w_n|w_{n-2}w_{n-1}) + \lambda_2 P(w_n|w_{n-1}) + \lambda_3 P(w_n)$$

$$\sum_i \lambda_i = 1$$

3 – Generalization and Zeros

Add-1 Smoothing – "Truyện Kiều"

	<s>	trăm	năm	trong	cõi	người	ta
<s>	0	13	5	23	1	22	1
trăm	0	0	9	0	0	0	0
năm	0	0	4	1	0	1	0
trong	0	0	1	0	3	1	0
cõi	0	0	0	0	0	1	0
người	0	0	0	2	0	2	7
ta	0	0	0	0	0	0	2

3 – Generalization and Zeros

Add-1 Smoothing – "Truyện Kiều"

	<s>	trăm	năm	trong	cõi	người	ta
<s>	1	14	6	24	2	23	2
trăm	1	1	10	1	1	1	1
năm	1	1	5	2	1	2	1
trong	1	1	2	1	4	2	1
cõi	1	1	1	1	1	2	1
người	1	1	1	3	1	3	8
ta	1	1	1	1	1	1	3

3 – Generalization and Zeros

Add-1 Smoothing – "Truyện Kiều"

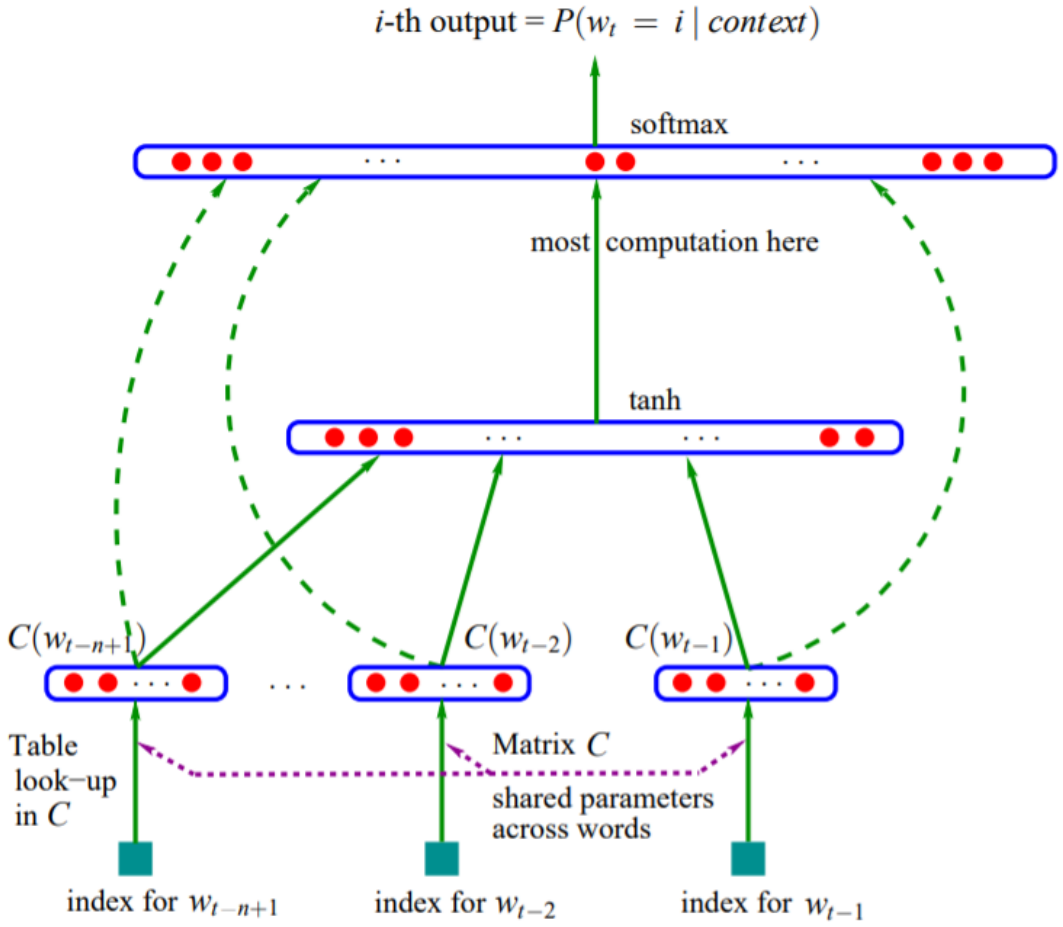
	<s>	trăm	năm	trong	cõi	người	ta
<s>	0.00015	0.0021	0.0009	0.0036	0.0003	0.0035	0.003
trăm	0.0004	0.0004	0.0041	0.0004	0.0004	0.0004	0.0004
năm	0.0004	0.0004	0.002	0.0008	0.0004	0.0008	0.0004
trong	0.0004	0.0004	0.0008	0.0004	0.0015	0.0008	0.0004
cõi	0.0004	0.0004	0.0004	0.0004	0.0004	0.0008	0.0004
người	0.0004	0.0004	0.0004	0.0011	0.0004	0.0011	0.003
ta	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0012

4 – Neural Language Model

Source: [Paper]- A Neural Probabilistic Language Model

“trăm năm trong cõi người ta”

Source	Target
trăm	năm
…	…
trăm năm	trong
…	…
trăm năm trong	cõi
…	…
trăm năm trong cõi người	ta



AI VIET NAM

@aivietnam.edu.vn

Thanks!

Any questions?