Blank Language Models

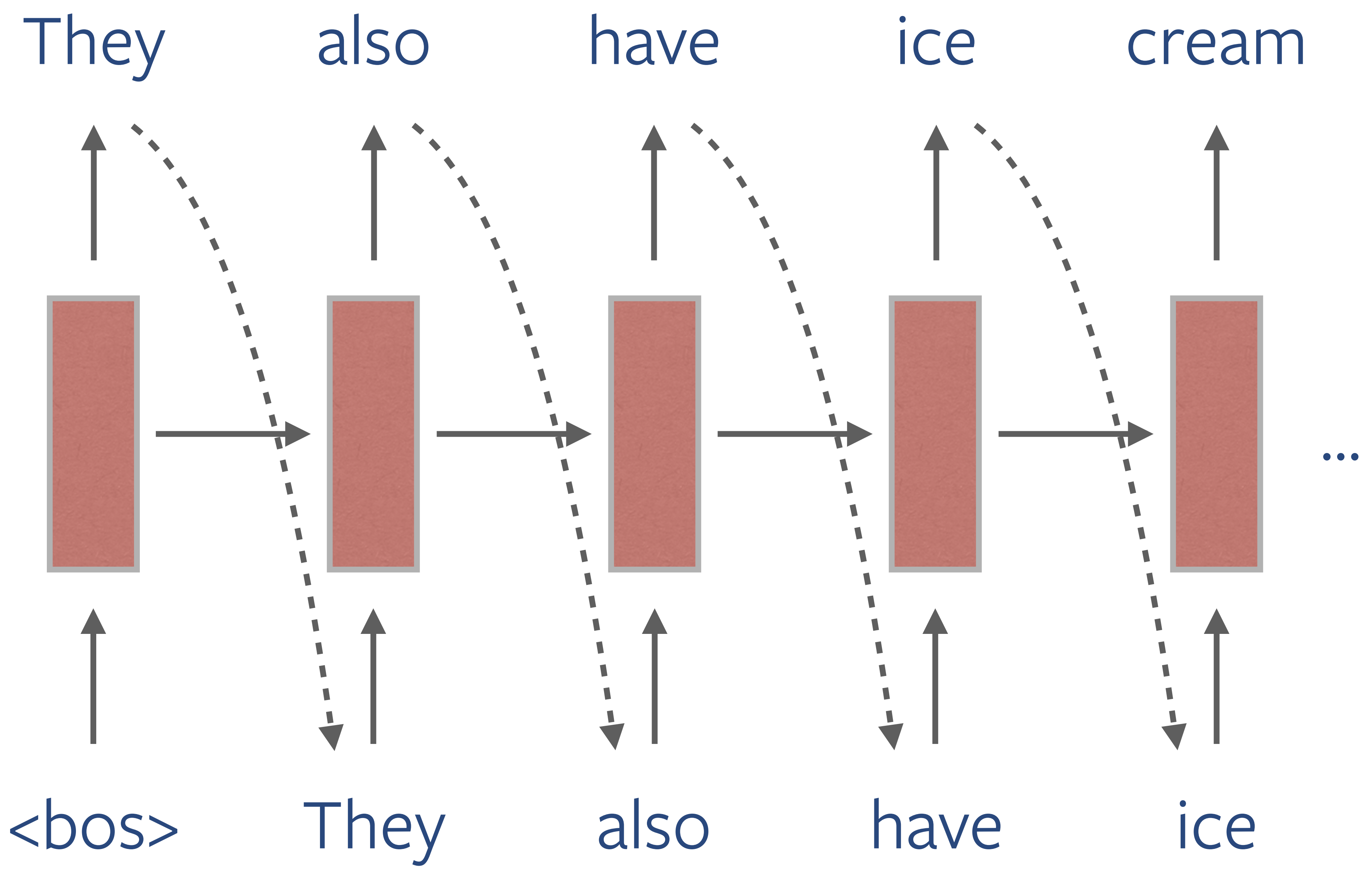
Tianxiao Shen* Victor Quach* Regina Barzilay Tommi Jaakkola (*: equal contribution)
tianxiao@mit.edu

Left-to-Right Language Model

✔ Generate from scratch

✘ Start with partially specified text

- text editing
- template filling
- text restoration
- …



Blank Language Model (BLM)

Input: They also have _____ which _____ .

Output: They also have ice cream which is really good .

- ✓ Fine-grained control over generation location
- ✓ Respect preceding and following context
- ✓ Variable number of missing tokens

Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
    1. select a “___”
    2. predict a word w
    3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ______ which ______ .

Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

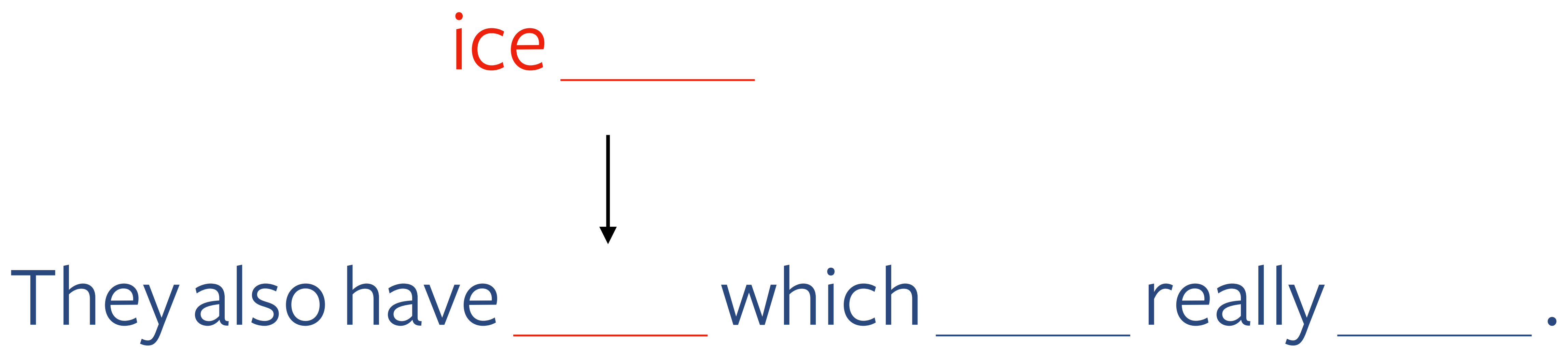
Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

They also have ____ which ____ really ____ .

Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"



Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

They also have ice ___ which ___ really ___ .

Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

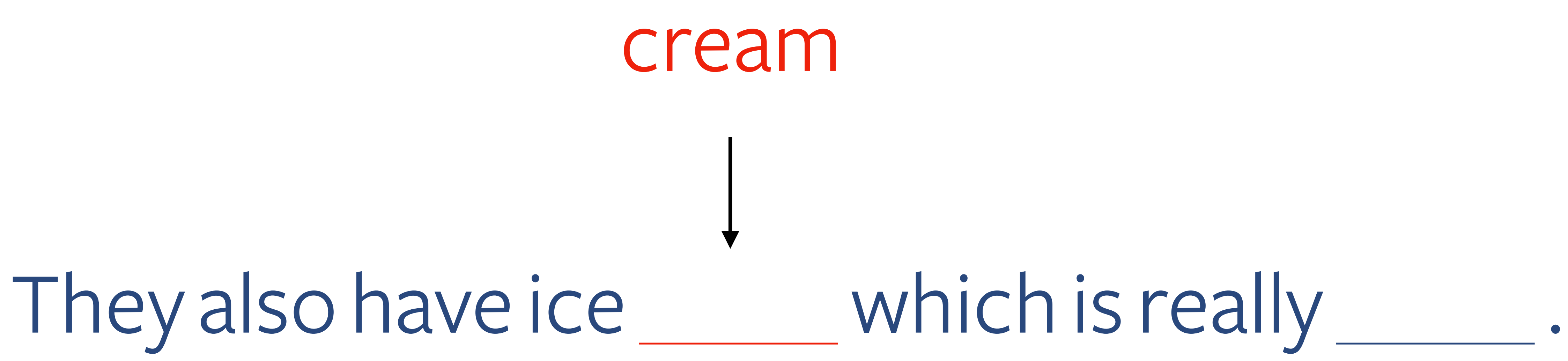
Blank Language Model — Overview

- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

They also have ice ______ which is really ______ .

Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”



Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

good

They also have ice cream which is really ___ .

Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ice cream which is really good .

Blank Language Model — Overview

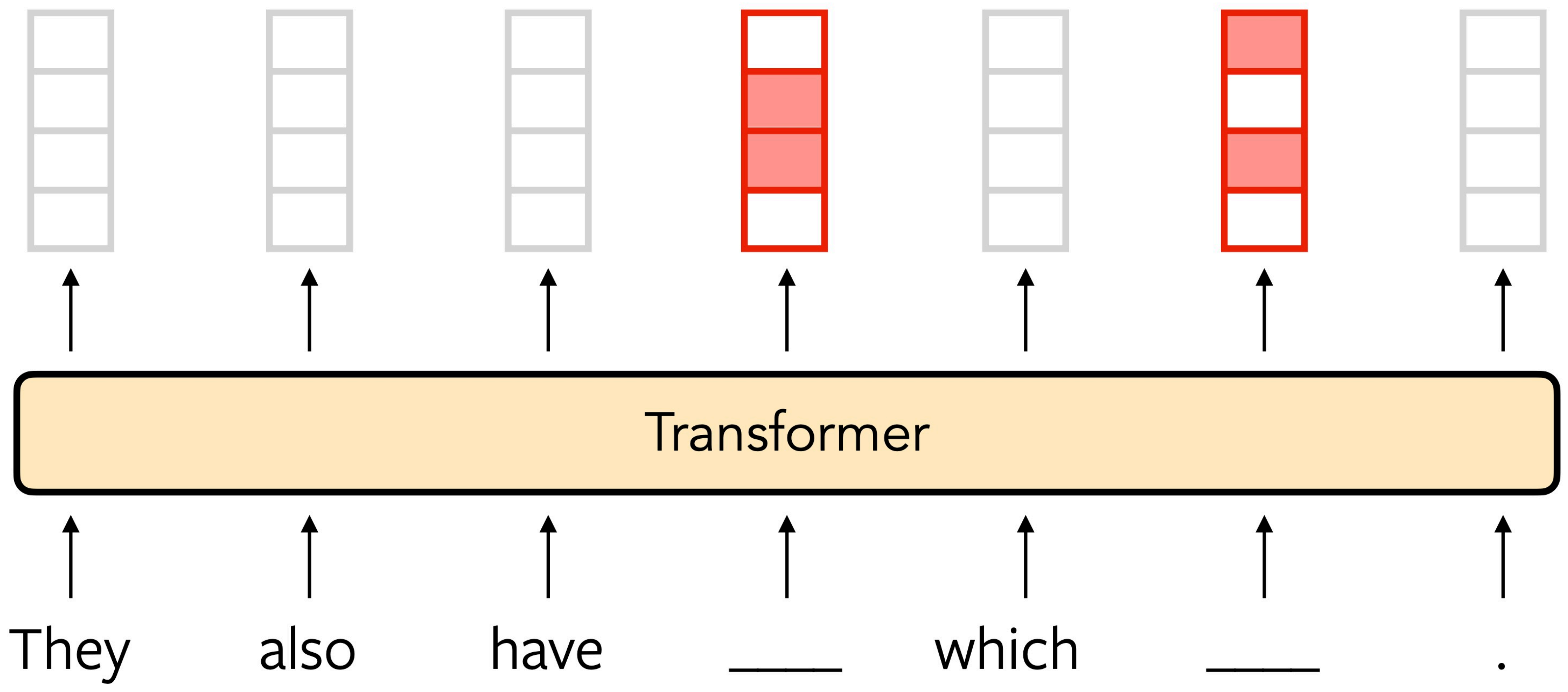
- Dynamic canvas where "___" controls where tokens can be placed
- At each step,
  1. select a "___"
  2. predict a word w
  3. replace that blank with "w", "___ w", "w ___", or "___ w ___"
- Stop when there is no "___"

Grammar

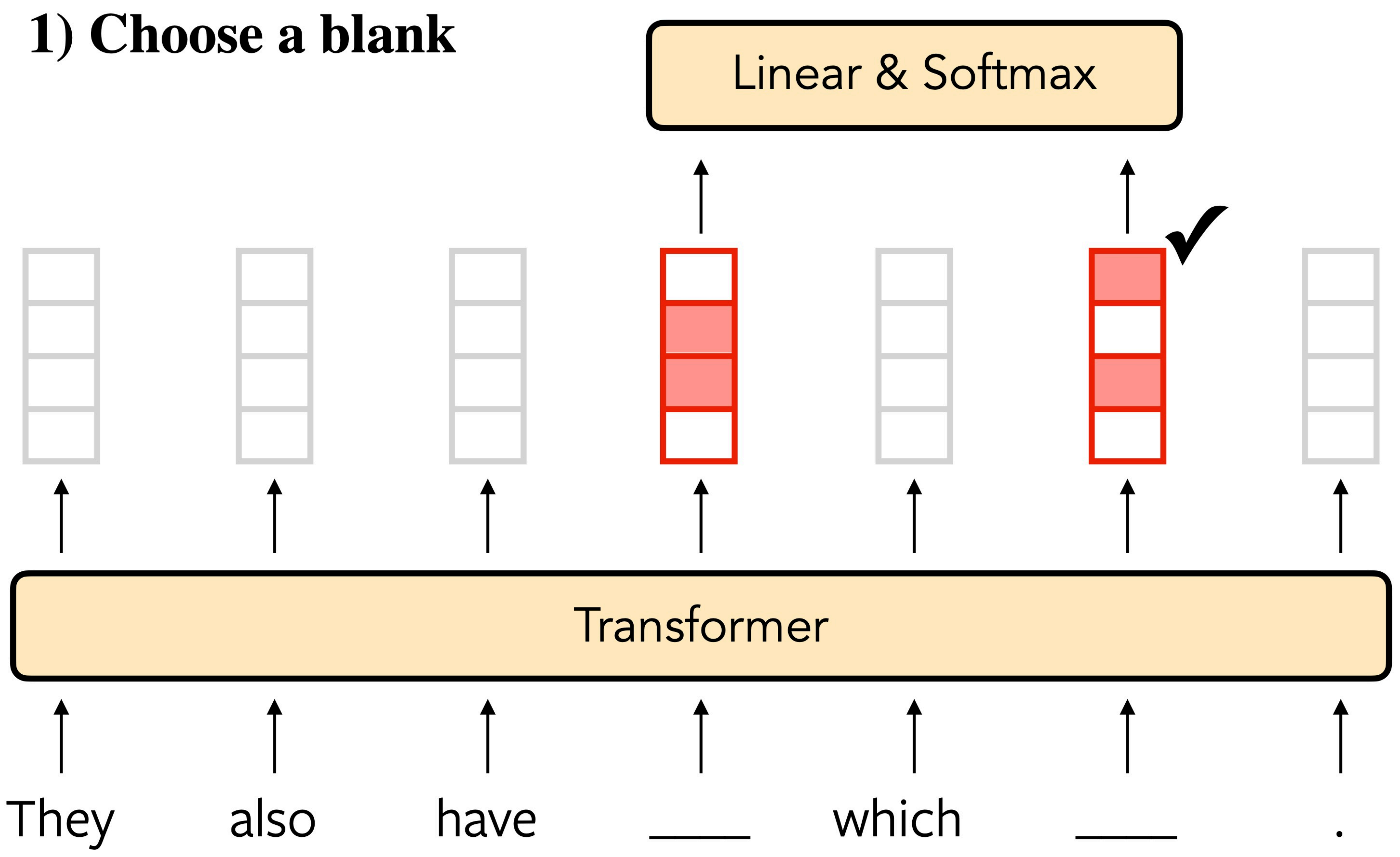
- Nonterminal: ___
- Terminals: $w \in V$
- Production rules: ___ → ___? w ___?

  (dist. depends on model and context)

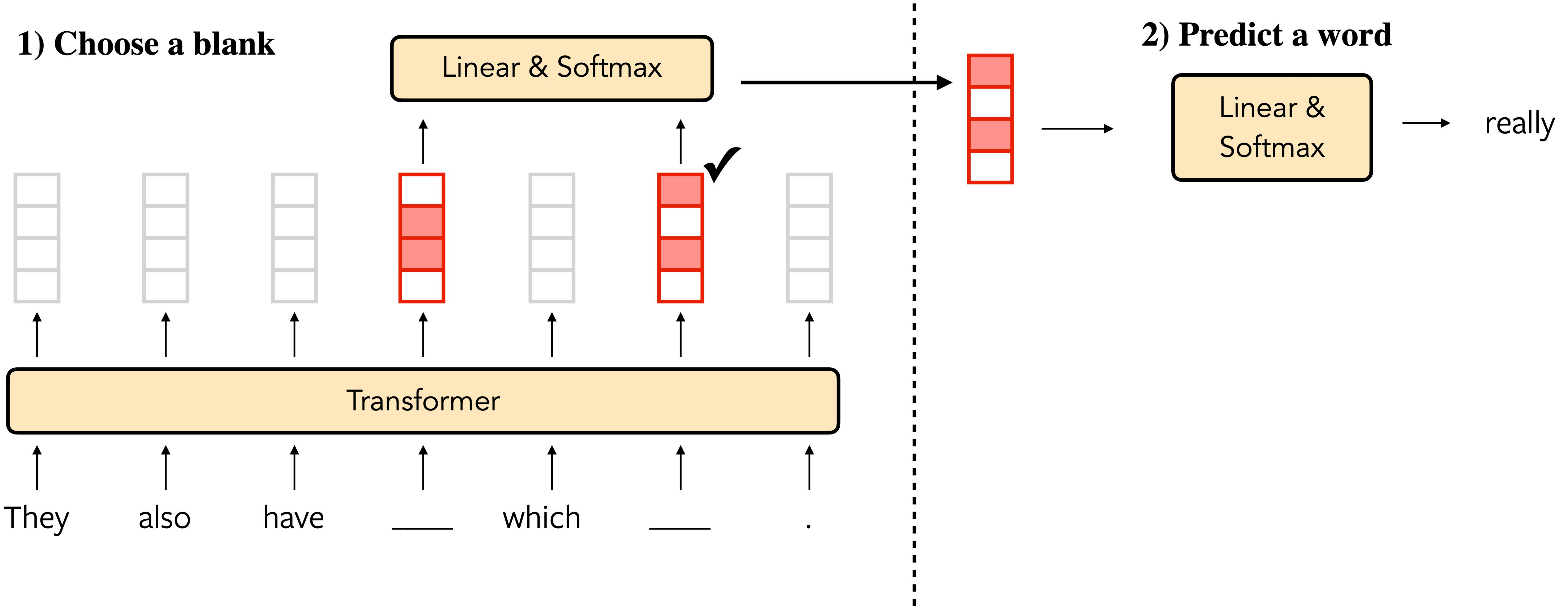
Blank Language Model — Architecture



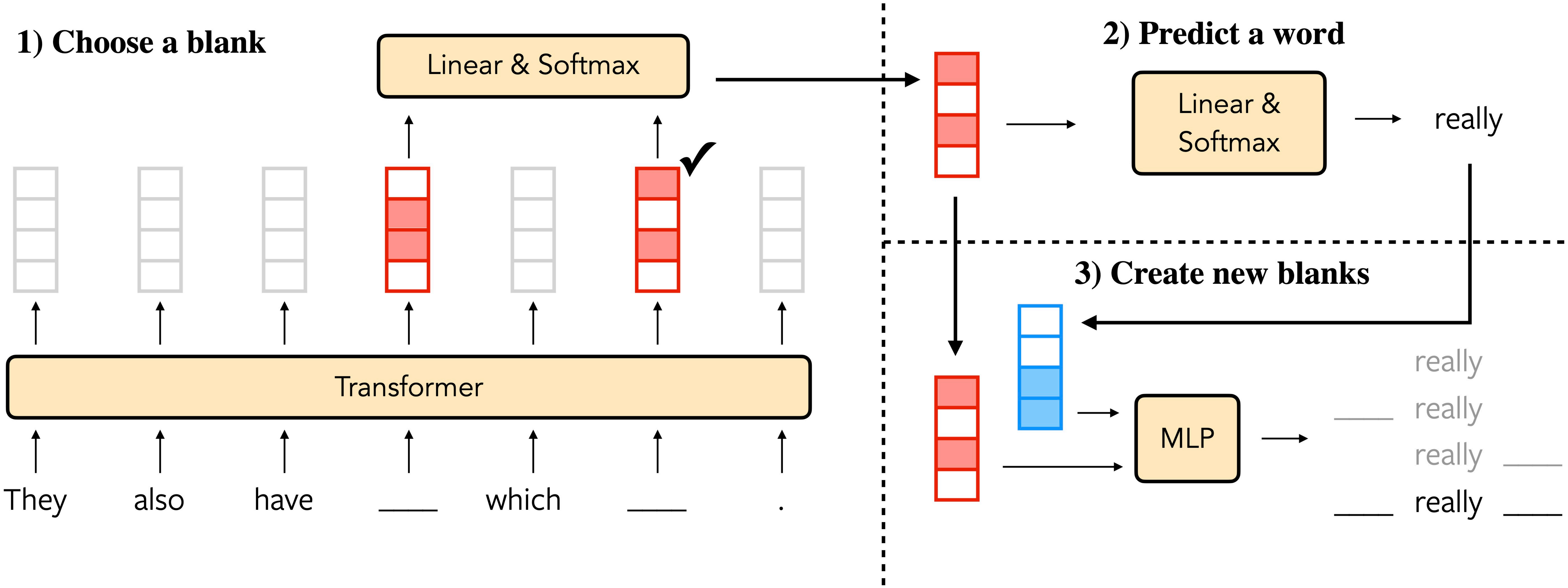
Blank Language Model — Architecture



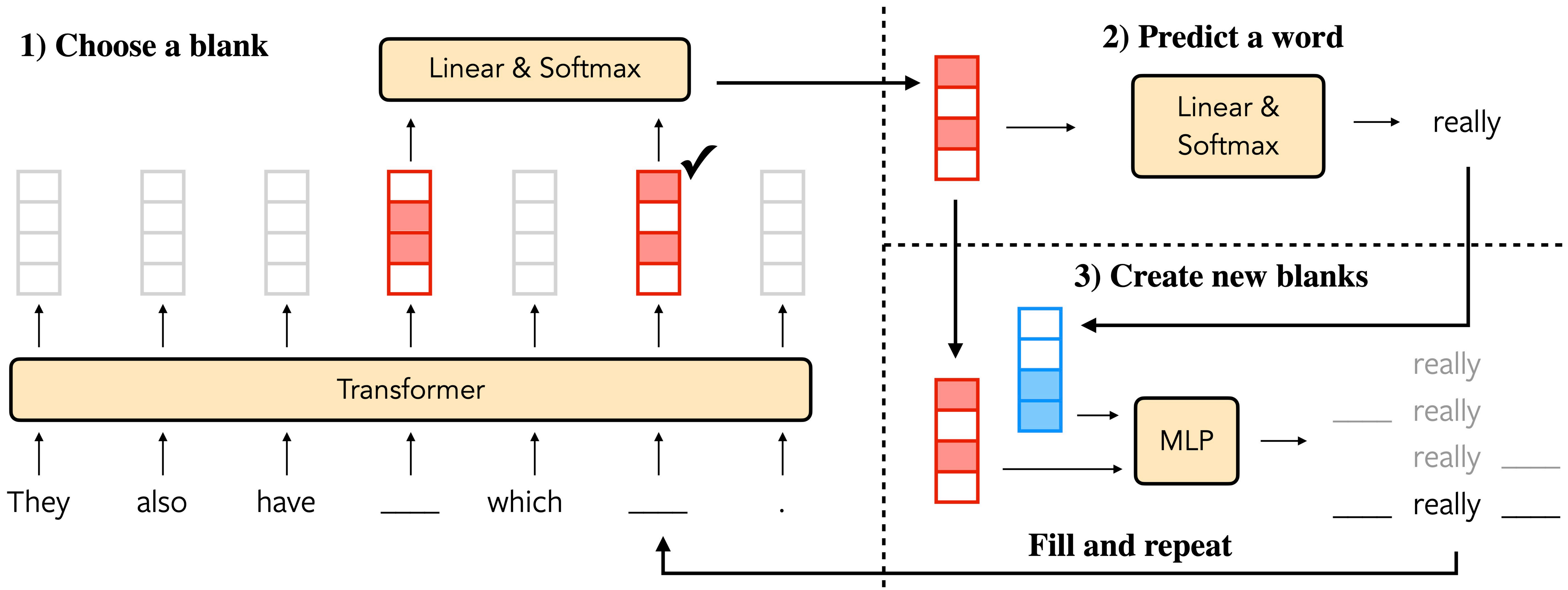
Blank Language Model — Architecture



Blank Language Model — Architecture



Blank Language Model — Architecture



Blank Language Model — Likelihood

trajectory

Step t	Canvas c	Location b		Word w	Action a (Left blank l,	Right blank r)
0.	__#1__	#1	3	have	Yes	Yes
1.	__#1__ have __#2__	#1	1	They	No	Yes
2.	They __#1__ have __#2__	#2	10	.	Yes	No
3.	They __#1__ have __#2__ .	#2	6	which	Yes	Yes
4.	They __#1__ have __#2__ which __#3__ .	#1	2	also	No	No
5.	They also have __#1__ which __#2__ .	#2	8	really	Yes	Yes
6.	They also have __#1__ which __#2__ really __#3__ .	#1	4	ice	No	Yes
7.	They also have ice __#1__ which __#2__ really __#3__ .	#2	7	is	No	No
8.	They also have ice __#1__ which is really __#2__ .	#1	5	cream	No	No
9.	They also have ice cream which is really __#1__ .	#1	9	good	No	No
10.	They also have ice cream which is really good .				-End-	

1 2 3 4 5 6 7 8 9 10

A sentence x with n words can be realized by $n!$ trajectories, each corresponds to a different word insertion order

$$p(x;\theta) = \sum_{\sigma \in S_n} p(x, \sigma; \theta) = \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

order

action, canvas at step t

Blank Language Model — Training

$$\log p(x;\theta) = \log \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

intractable

$$\log\left(\frac{1}{m}\sum_{i=1}^{m} b_i\right) \geq \frac{1}{m}\sum_{i=1}^{m} \log b_i$$

$$\geq \log(n!) + \frac{1}{n!}\sum_{\sigma \in S_n}\sum_{t=0}^{n-1} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

Blank Language Model — Training

$$\log p(x;\theta) = \log \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$ intractable

$$\log\left(\frac{1}{m}\sum_{i=1}^{m} b_i\right) \geq \frac{1}{m}\sum_{i=1}^{m} \log b_i$$

$$\geq \log(n!) + \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{t=0}^{n-1} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

1. Uniformly sample σ from S_n
2. Uniformly sample t from 0 to $n-1$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - n \cdot \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

one action loss per pass :(

Blank Language Model — Training

$c_t^{x,\sigma}$ only depends on $\sigma_{1:t}$

→ combine losses of trajectories with the same first t steps and different $(t+1)$-th step

$$\geq \log(n!) + \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{t=0}^{n-1} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

Blank Language Model — Training

$c_t^{x,\sigma}$ only depends on $\sigma_{1:t}$

→ combine losses of trajectories with the same first t steps and different $(t+1)$-th step

$$\geq \log(n!) + \sum_{t=0}^{n-1} \frac{1}{n!} \sum_{\sigma \in S_n} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$

$$= \log(n!) + n \cdot \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \mathbb{E}_{\sigma_{t+1}} \mathbb{E}_{\sigma_{t+2:n}} [\log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)]$$

$$= \log(n!) + \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \left[\frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta) \right]$$

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

$$= \log(n!) + \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \left[\frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta) \right]$$

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

	1	2	3	4	5	6	7	8	9	10
$x =$	They	also	have	ice	cream	which	is	really	good	.

$n = 10$

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

	1	2	3	4	5	6	7	8	9	10
$x =$	They	also	have	ice	cream	which	is	really	good	.

$n = 10$

$t = 5$

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

	1	2	3	4	5	6	7	8	9	10
$x =$	They	also	have	ice	cream	which	is	really	good	.

$n = 10$

$t = 5$

$\sigma_{1:t} = (6, 2, 1, 3, 10)$

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

	1	2	3	4	5	6	7	8	9	10
$x =$	They	also	have	ice	cream	which	is	really	good	.

$n = 10$

$t = 5$

$\sigma_{1:t} = (6, 2, 1, 3, 10)$

$c_t^{x,\sigma}$ They also have ___ which ___ .

Blank Language Model — Training

1. Uniformly sample t from 0 to $n-1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss $-\log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$

n/2 action losses per pass :)

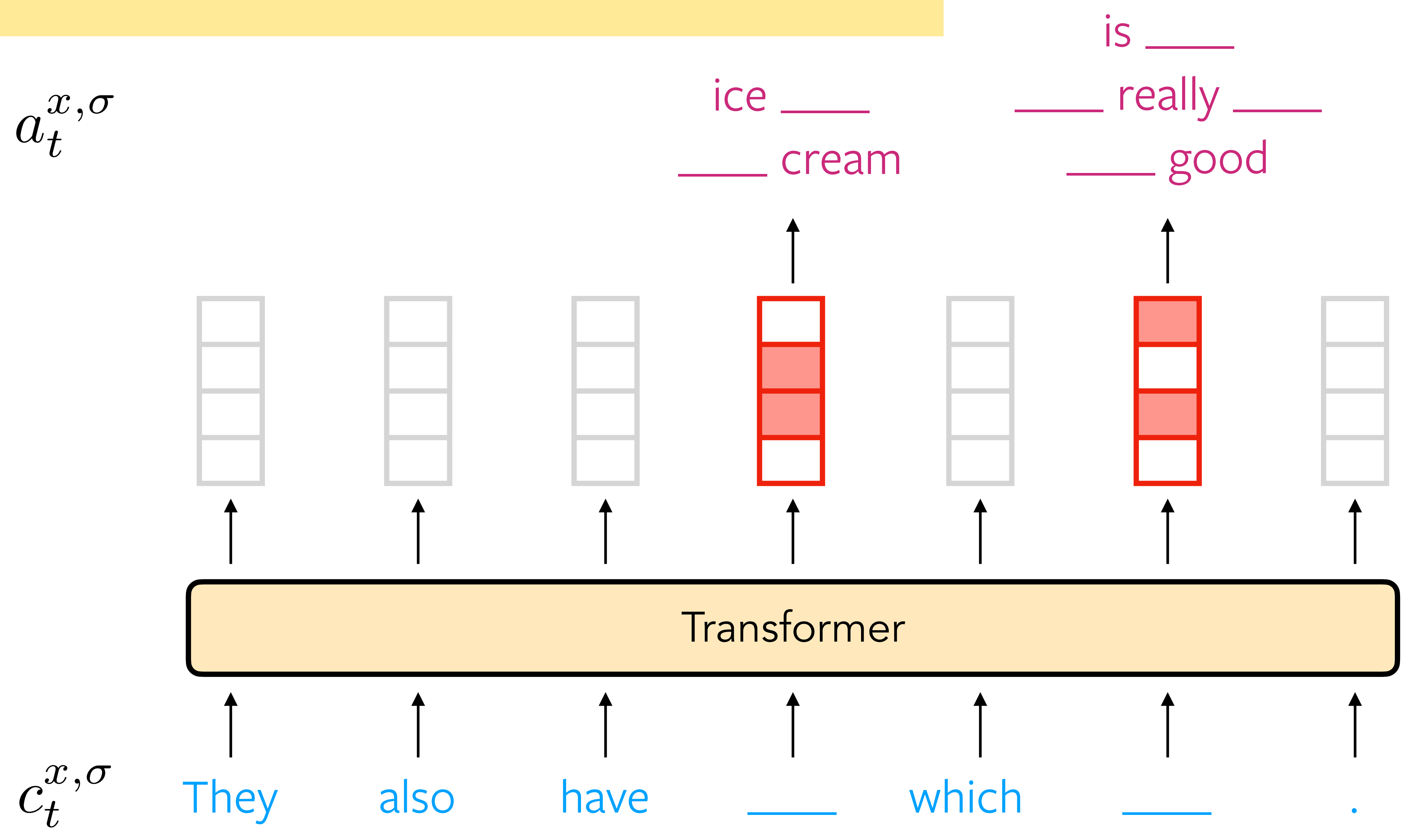
1	2	3	4	5	6	7	8	9	10
They	also	have	ice	cream	which	is	really	good	.

$x =$ They also have ice cream which is really good .

$n = 10$

$t = 5$

$\sigma_{1:t} = (6, 2, 1, 3, 10)$



Blank Language Model — Inference

✓ Simple greedy decoding or beam search to fill in the blanks in input

Experiments — Overview

Text Infilling	Input:	They also have ____ which ____ .
	Output:	They also have ice cream which is really good .
Ancient Text Restoration	Input:	τε εγγονον εισαι???????σοφιαι
	Output:	τε εγγονον εισαιου του σοφιαι
Sentiment Transfer	Input:	The employees were **super nice** and **efficient** !
	Output:	The employees were rude and unprofessional !
Language Modeling	Output:	They also have ice cream which is really good.

Text Infilling — Dataset

- Yahoo Answers dataset (100K documents, max length 200 words)
- Randomly mask tokens with different ratios
- Contiguous masked tokens → "___"

Mask Ratio 10%	when time flies , **where** does it go ? **to** the center of the **universe** to be recycled **and** made into new time . when time flies , _____ does it go ? _____ the center of the _____ to be recycled _____ made into new time .
Mask Ratio 50%	when time **flies** , where **does it go** ? **to** the **center** of **the** universe to **be** recycled **and** made into **new time** . when time _____ , where _____ ? _____ the _____ of _____ universe to _____ recycled _____ made into _____ .

Text Infilling — Metrics

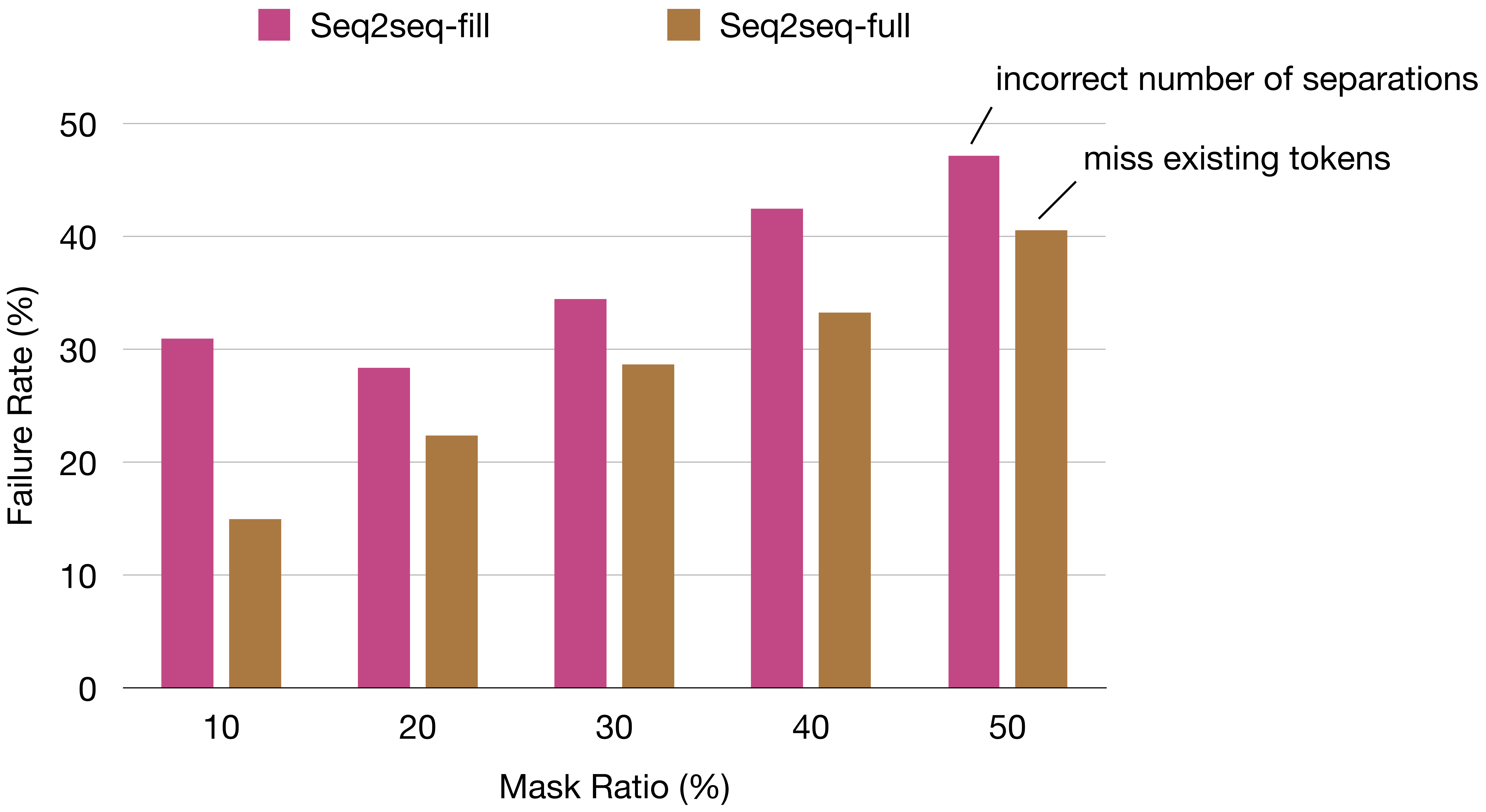
- Accuracy: BLEU score against original document
- Fluency: perplexity evaluated by a pre-trained left-to-right LM

Mask Ratio 10%	when time flies , **where** does it go ? **to** the center of the **universe** to be recycled **and** made into new time . when time flies , _____ does it go ? _____ the center of the _____ to be recycled _____ made into new time .
Mask Ratio 50%	when time **flies** , where **does it go** ? **to** the **center** of **the** universe to **be** recycled **and** made into **new time** . when time _____ , where _____ ? _____ the _____ of _____ universe to _____ recycled _____ made into _____ .

Text Infilling — Baselines

- Seq2seq-fill [Donahue et al., 2020]
  - output tokens to fill in the blanks, separated by "|"
- Seq2seq-full [Donahue et al., 2020]
  - output full document from input

Text Infilling — Results



Text Infilling — Baselines

- BERT+LM
  - feed BERT representation of each blank to left-to-right LM that learns to generate tokens in that blank
  - at test time, fill in the blanks one by one

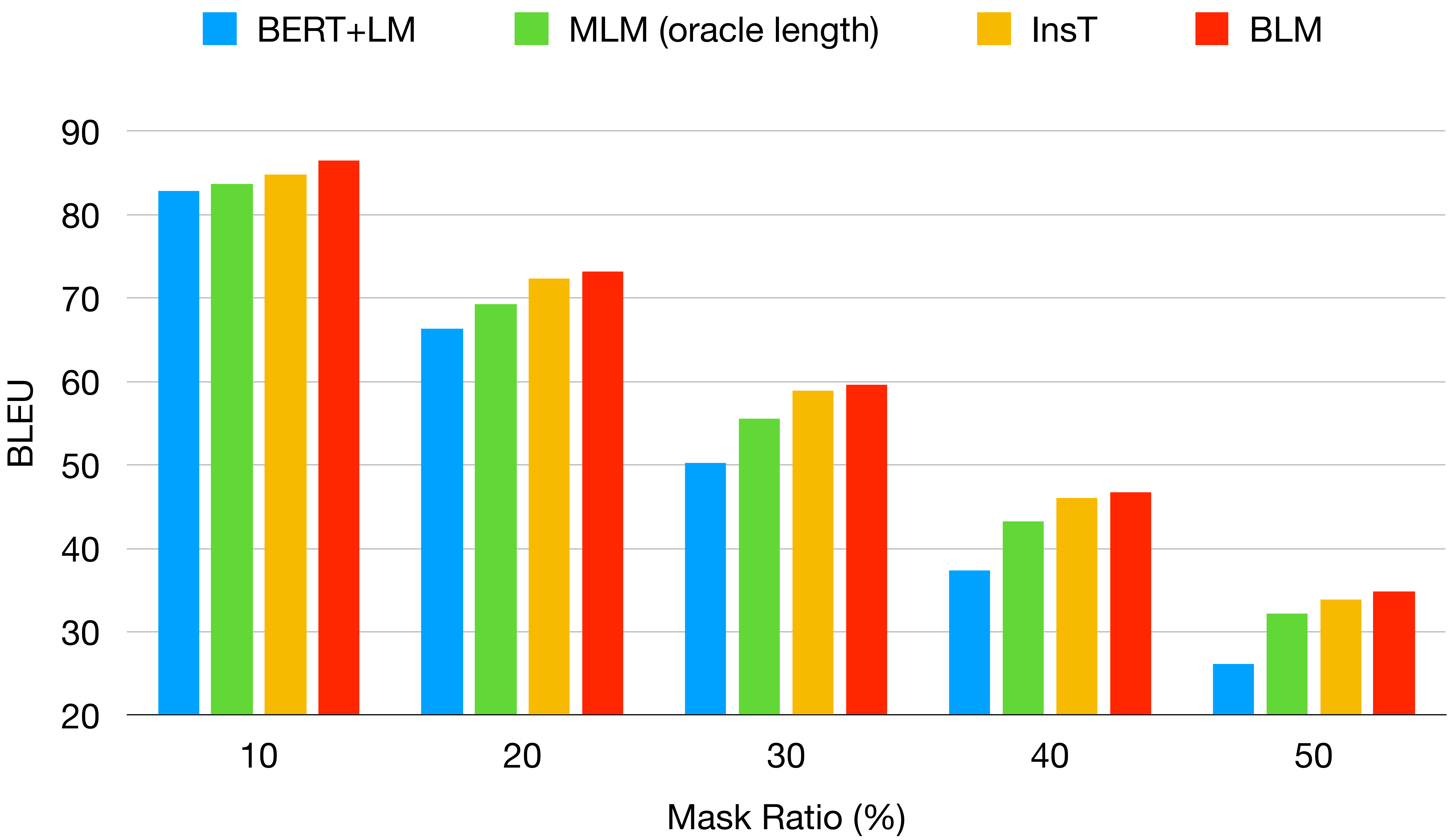
Text Infilling — Baselines

- BERT+LM
- Masked Language Model (MLM) with oracle length
  - replace blanks with the target number of masks
  - fill the masks autoregressively by most-confident-first heuristic

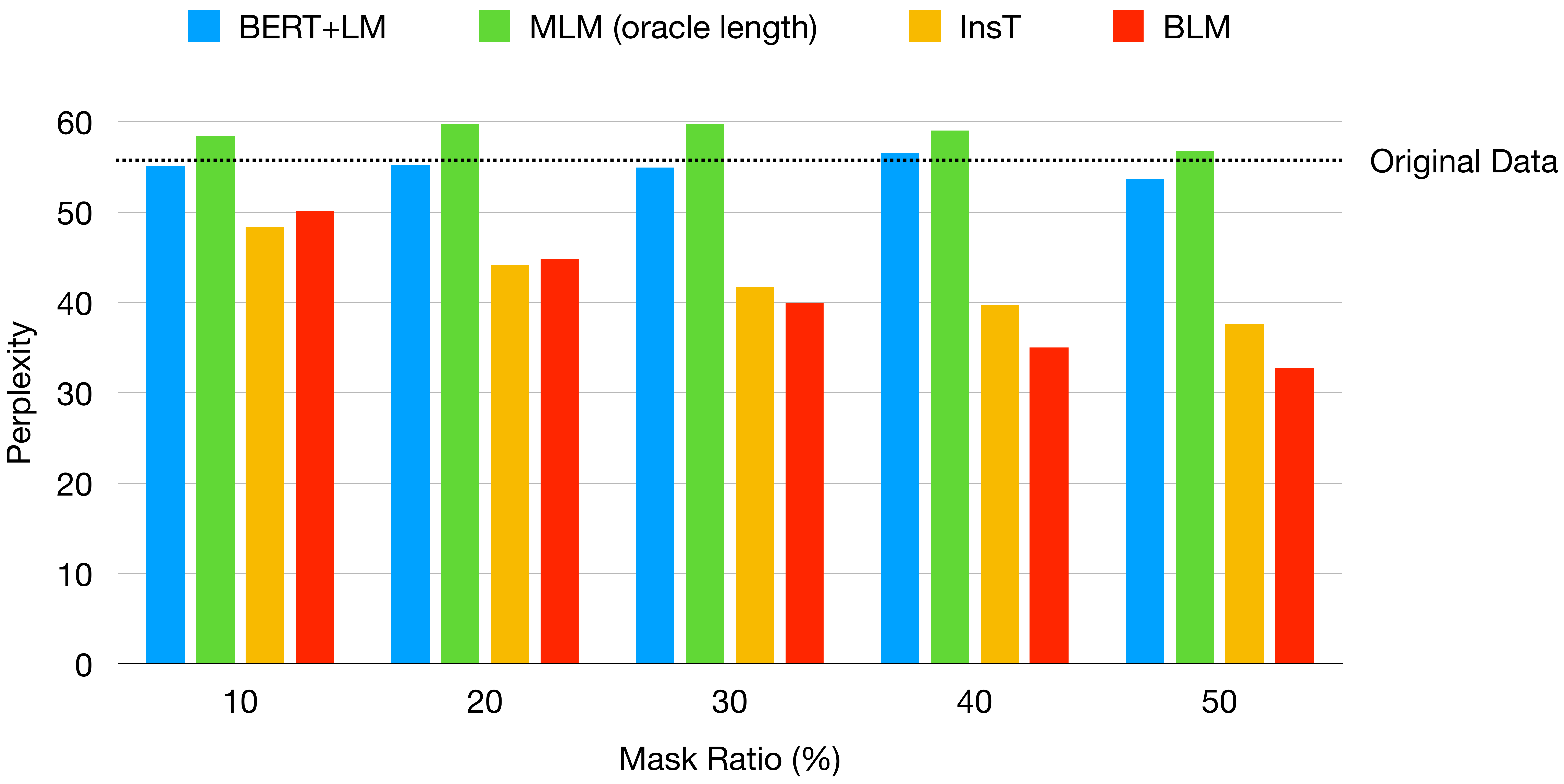
Text Infilling — Baselines

- BERT+LM
- Masked Language Model (MLM) with oracle length
- Insertion Transformer [Stern et al., 2019]
  - cannot specify insertion position
  - force it to generate at valid locations

Text Infilling — Results



Text Infilling — Results



Text Infilling — Examples

Original	when time flies , **where** does it go ? **to** the center of the **universe** to be recycled **and** made into new time .
Blanked	when time flies , ____ does it go ? ____ the center of the ____ to be recycled ____ made into new time .
BERT+LM	when time flies , <u>where</u> does it go ? <u>to</u> the center of the <u>earth</u> to be recycled <u>came</u> made into new time .
MLM (oracle len)	when time flies , <u>where</u> does it go ? <u>from</u> the center of the <u>earth</u> to be recycled <u>converted</u> made into new time .
InsT	when time flies , <u>where</u> does it go ? <u>for</u> the center of the <u>earth has</u> to be recycled <u>and</u> made into new time .
BLM	when time flies , <u>where</u> does it go ? <u>for</u> the center of the <u>earth</u> to be recycled <u>and</u> made into new time .

Mask Ratio 10%

Text Infilling — Examples

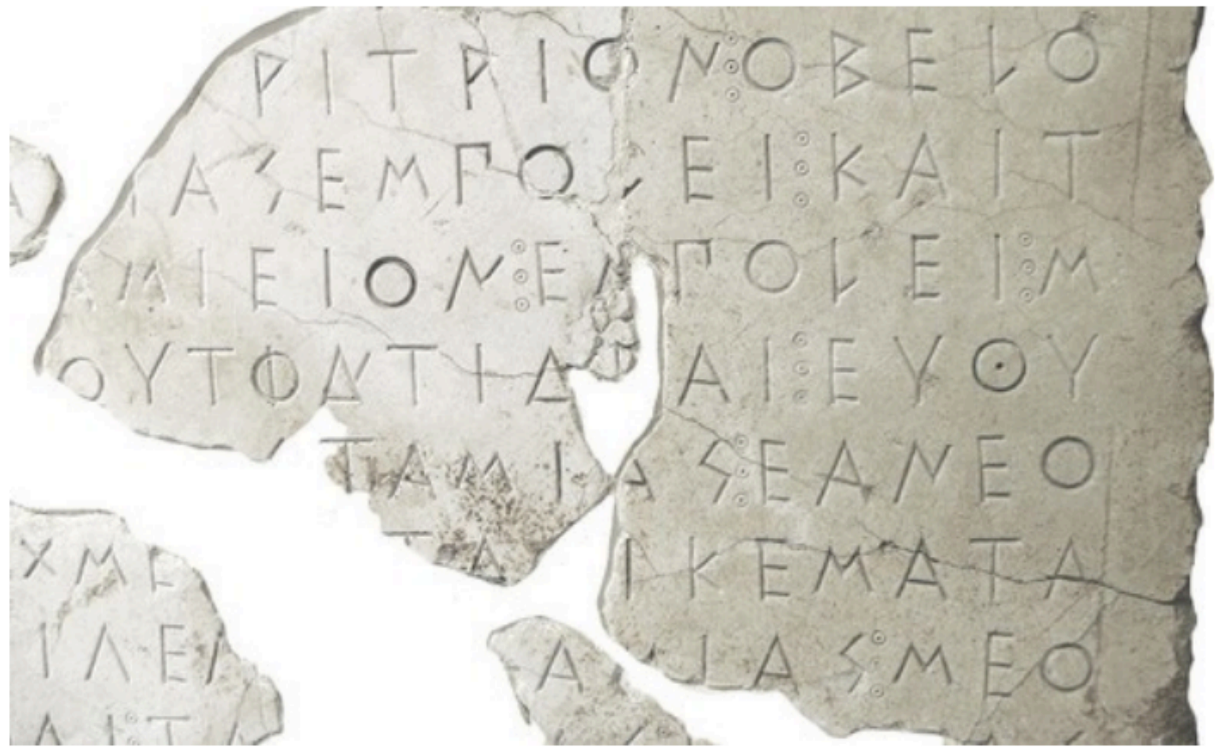
Original	when time **flies** , where **does it go** ? **to** the **center** of **the** universe to **be** recycled **and** made into **new time** .
Blanked	when time ____ , where ____ ? ____ the ____ of ____ universe to ____ recycled ____ made into ____ .
BERT+LM	when time is , where to ? i need to find the way of the universe to be recycled and made into a lot .
MLM (oracle len)	when time is , where is the universe ? from the creation of the universe to be recycled and made into the universe .
InsT	when time was created , where was it ? what was the name of the universe to be recycled and made into space .
BLM	when time was created , where did it come from ? it was the first part of the universe to be recycled and made into space .

Mask Ratio 50%

Ancient Text Restoration — Setup

Ancient Greek Inscriptions dataset (18M characters / 3M words) [Assael et al., 2019]

- number of characters to recover is assumed to be known



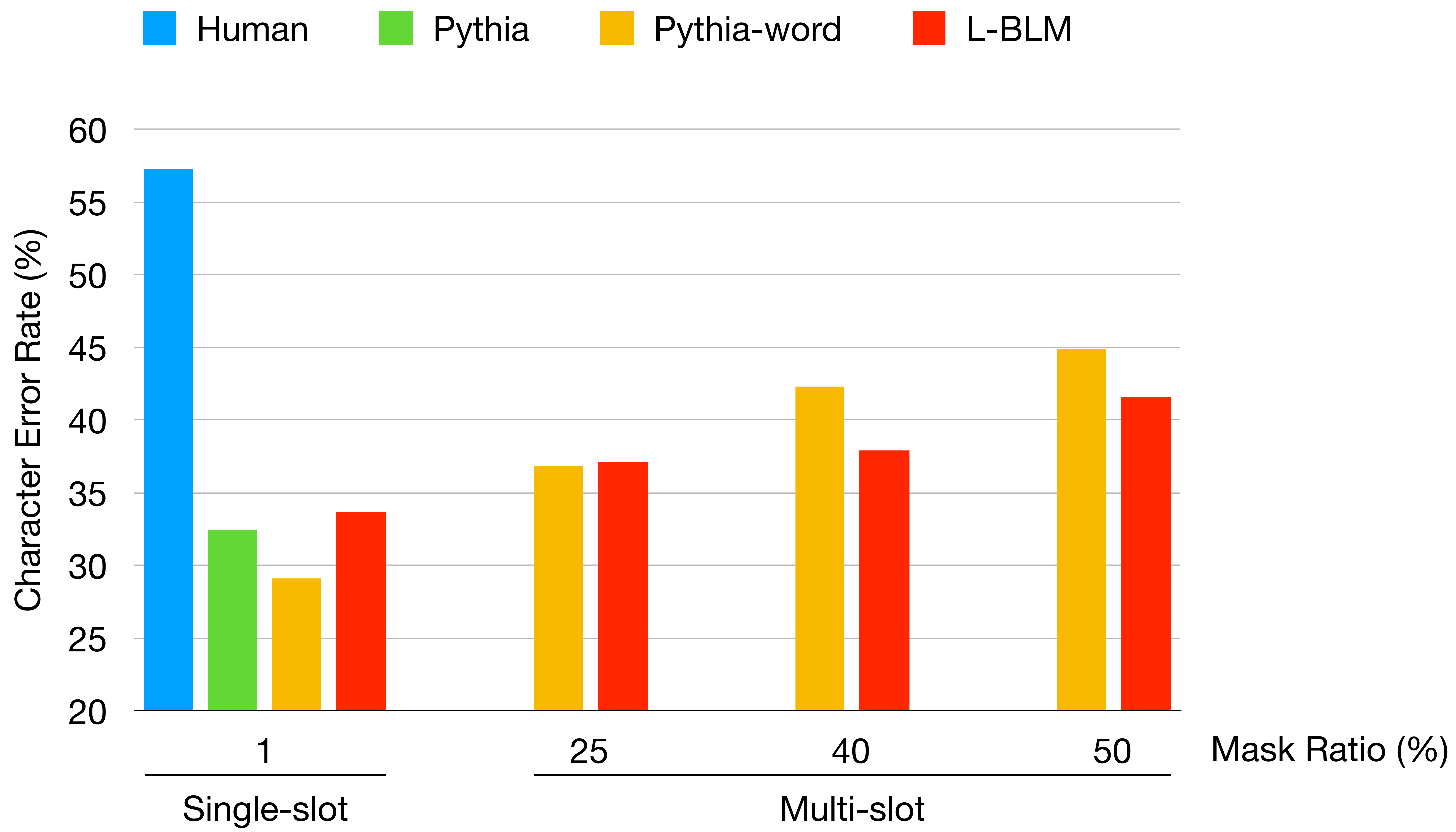
Length-aware BLM (L-BLM)

- __[t]__ → __[k]__ w __[t-1-k]__

Baselines [Assael et al., 2019]:

- Pythia: character-level seq2seq model to fill in one slot at a time
- Pythia-word: use both character and word representations

Ancient Text Restoration — Results



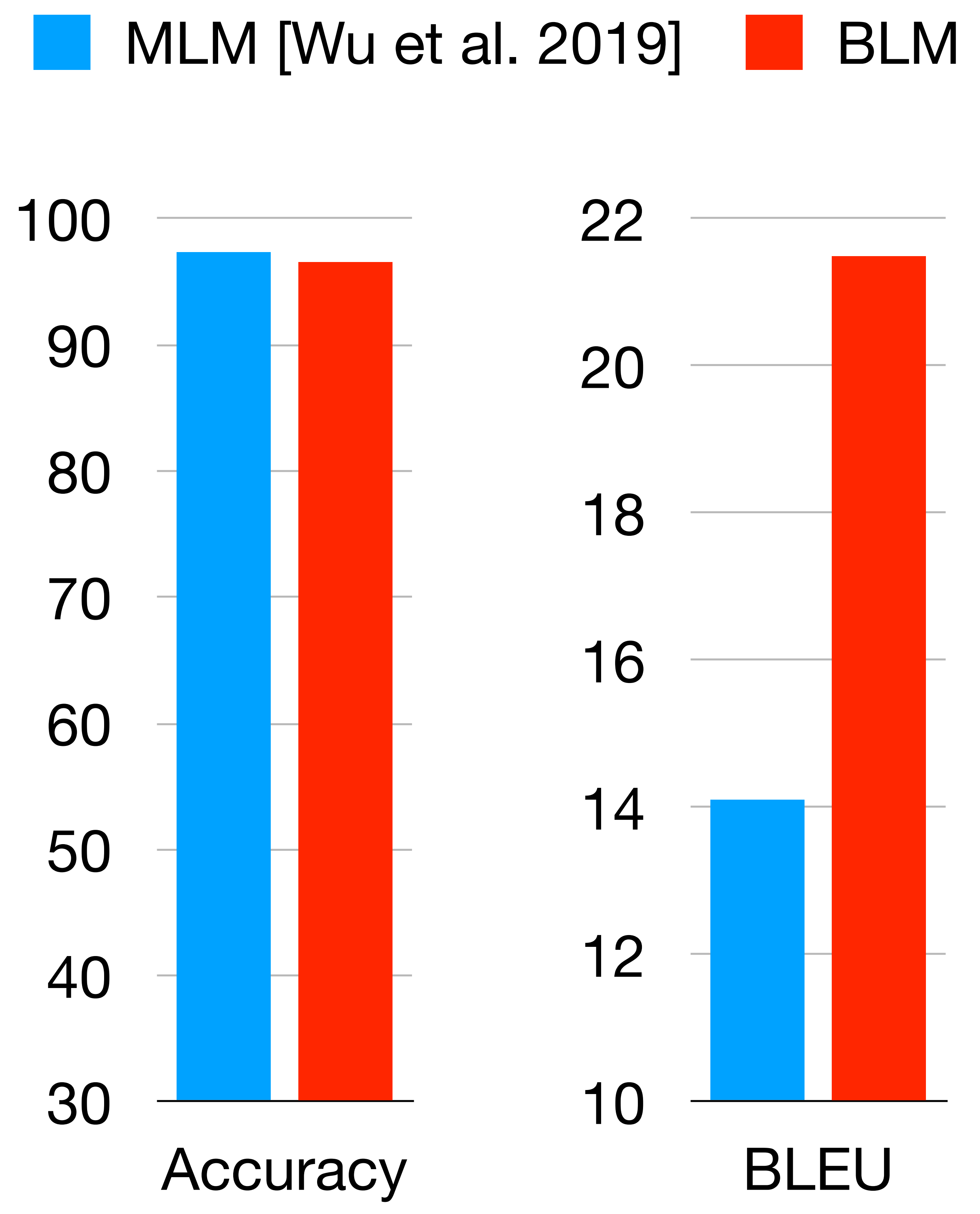
Sentiment Transfer — Approach

1. Remove expressions of high polarity
   - train a sentiment classifier and mask words with attention weight above average
2. Complete the partial sentence with expressions of the target sentiment
   - train two instances of BLM, one for each sentiment

Sentiment Transfer — Results

Input	everyone that i spoke with was **very helpful** and **kind** .
BLM	everyone that i spoke with was **rude** and **unprofessional** .
Input	there is **definitely not** enough **room** in that part of the venue .
BLM	there is **always** enough **parking** in that part of the venue .
Input	it is n't **terrible** , but it is **n't** very good either .
BLM	it is n't **fancy** , but it is **still** very good either .

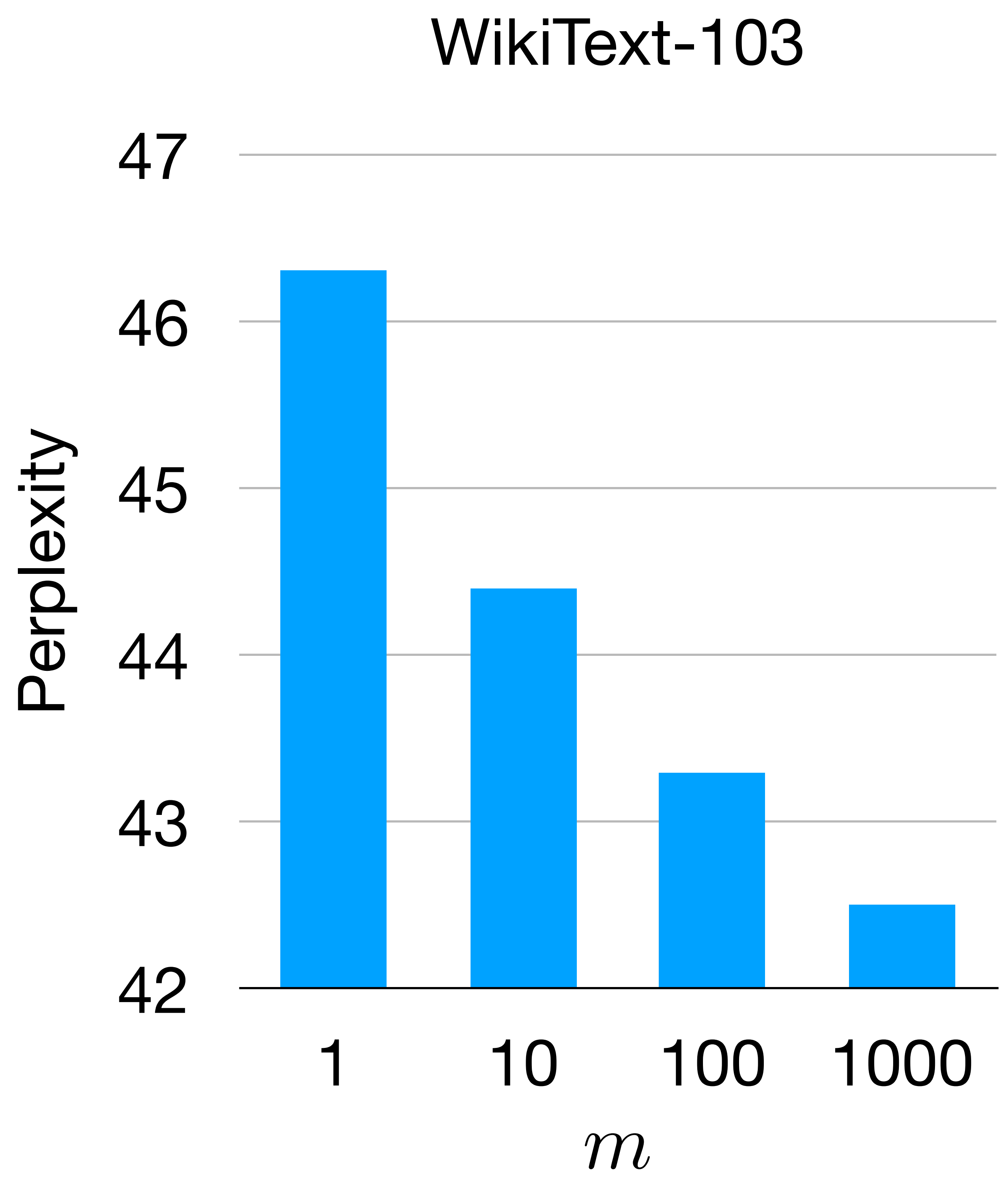
Yelp Reviews Dataset



Language Modeling — Estimation

Monte-Carlo sampling $p(x;\theta) = \sum_{\sigma \in S_n} p(x,\sigma;\theta) \leftarrow \frac{n!}{m} \sum_{i=1}^{m} p(x,\sigma_i;\theta)$

- estimated perplexity is likely to be higher than actual perplexity
- as m increases, it converges to actual perplexity



Language Modeling — Results

Datasets: Penn Treebank (1M tokens), WikiText-2 (2M), WikiText-103 (103M)

	PTB	WT2	WT103
LSTM (Grave et al., 2016)	82.3	99.3	48.7
TCN (Bai et al., 2018)	88.7	-	45.2
AWD-LSTM (Merity et al., 2017)	57.3	**65.8**	-
Transformer (Dai et al., 2019)	-	-	30.1
Adaptive (Baevski and Auli, 2018)	-	-	18.7
Transformer-XL (Dai et al., 2019)	**54.5**	-	**18.3**
InsT (our implementation)	77.3	91.4	39.4
BLM	69.2	81.2	42.5

Room for improvements!

Summary

Input: They also have ____ which ____ .

Output: They also have ice cream which is really good .

https://github.com/Varal7/blank_language_model

Thank you!

- Dynamically create and fill in blanks
- Effective on text infilling, ancient text restoration, style transfer

More Applications

- Template filling, information fusion, assisting human writing…
- Rewrite to mitigate toxicity and bias
- Representation learning

Extensions

- Add representation for blanks
- Conditional BLM: edit and refine machine translation, dialogue system…