Controllable Text Generation by Search-and-Learning Methods

Lili Mou
University of Alberta
lmou@ualberta.ca
lili-mou.github.io

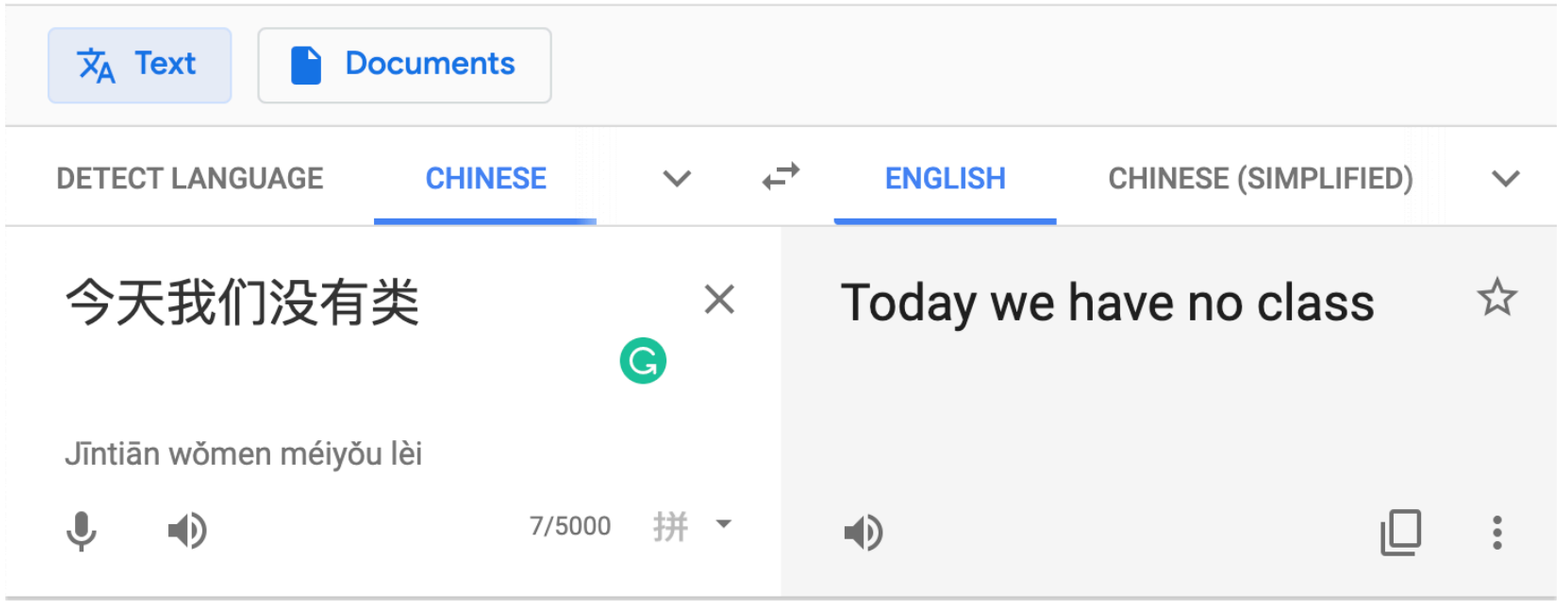
July 8, 2022

Outline

- **Introduction**
- Improving semantic coverage for data-to-text generation [AAAI'21]
- Controlling the length for summarization [ACL'20; 22]
- Conclusion and Discussion

Why NLG is interesting?

- Industrial applications
  - Machine translation
  - Headline generation for news
  - Grammatical error correction



https://translate.google.com/

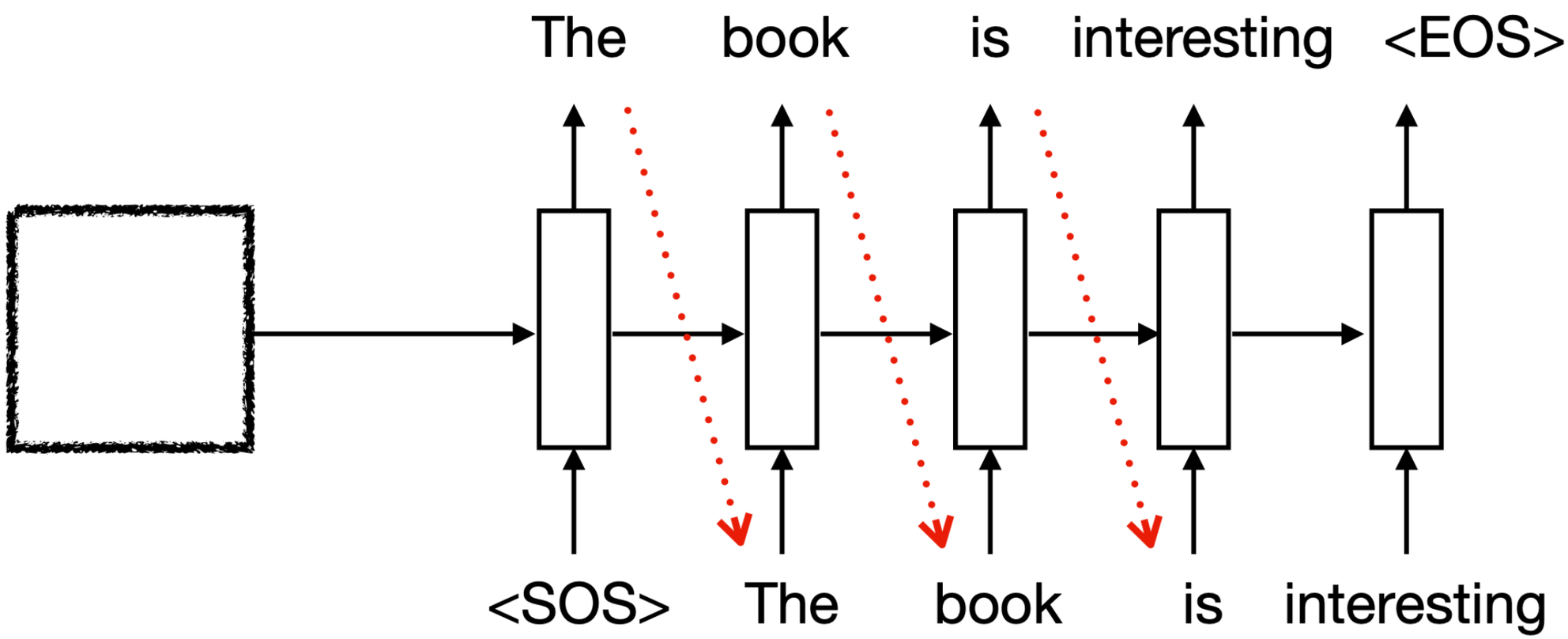
amii UNIVERSITY OF ALBERTA

Why NLG is interesting?

- Industrial applications
  - Machine translation
  - Headline generation for news
  - Grammatical error correction
- Scientific questions
  - Non-linear dynamics for long-text generation
  - Discrete “multi-modal” distribution

Supervised Text Generation

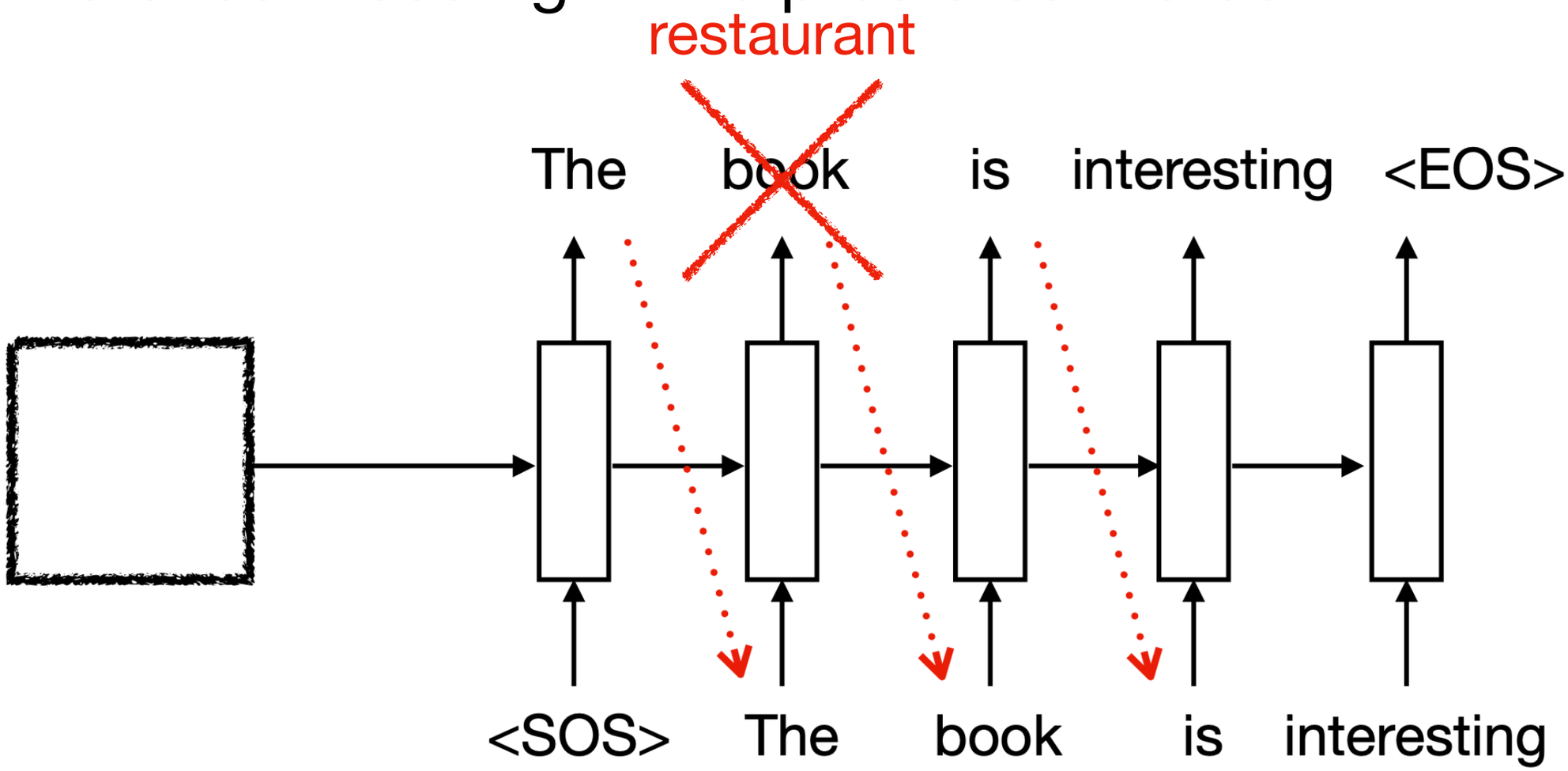
Sequence-to-sequence training



Main Disadvantages

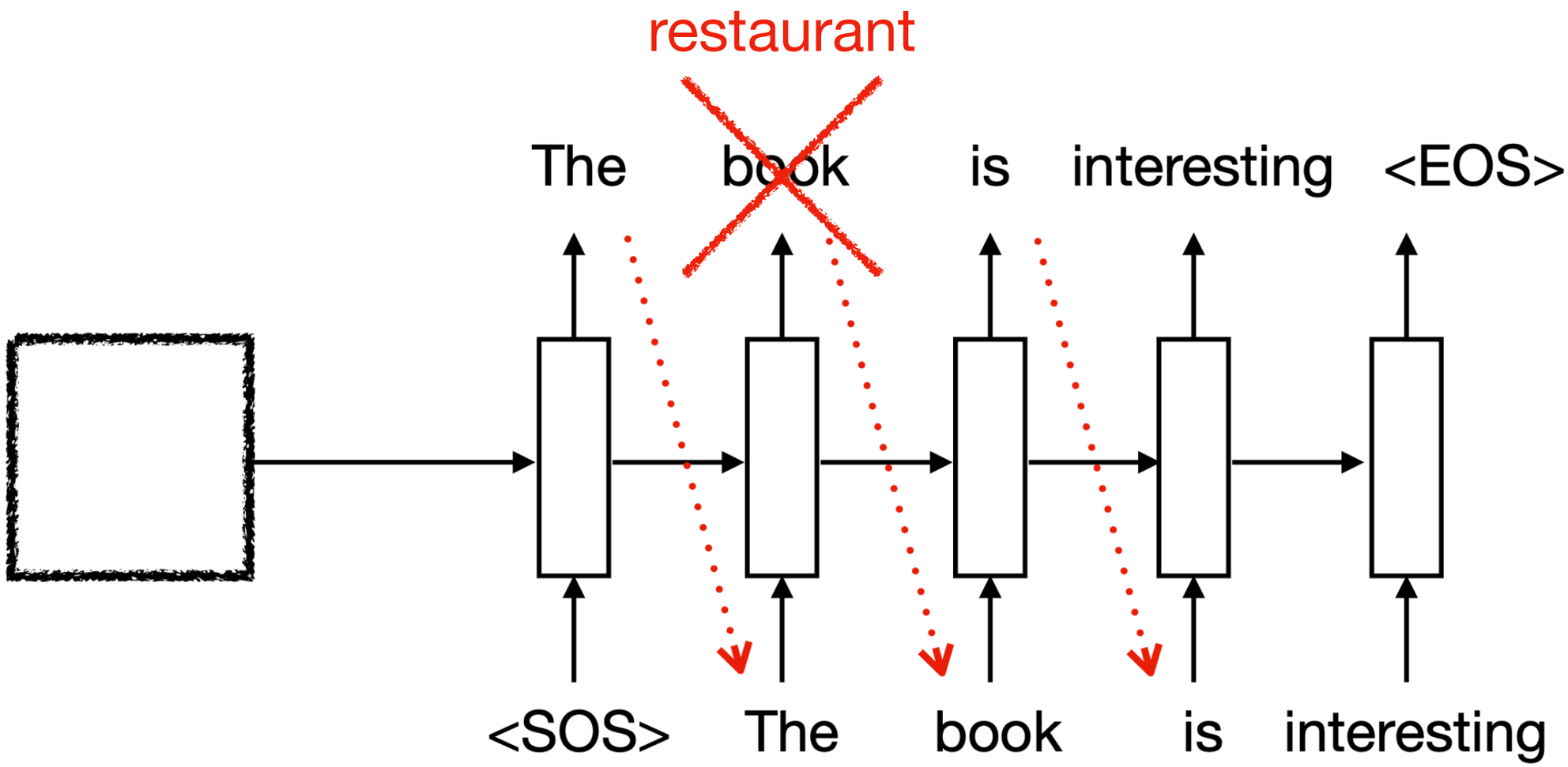
Exposure bias / Error accumulation

- Training: Feeding in the partial ground-truth
- Inference: Feeding in the predicted words



Main Disadvantages

- Reinforcement/Imitation learning partially solves the problem
- But error accumulation still exists; model not controllable



Generation by Local Changes

E.g., paraphrase generation [AAAI'19, ACL'20, NeurIPS'20]

- Suppose we have a blueprint

The book is interesting <EOS>

Generation by Local Changes

E.g., paraphrase generation [AAAI'19, ACL'20, NeurIPS'20]

- Suppose we have a blueprint

The book is interesting <EOS>

~~The~~ book is interesting <EOS>

This

Generation by Local Changes

E.g., paraphrase generation [AAAI'19, ACL'20, NeurIPS'20]

- Suppose we have a blueprint

The book is interesting <EOS>

~~The~~ book is interesting <EOS>

This

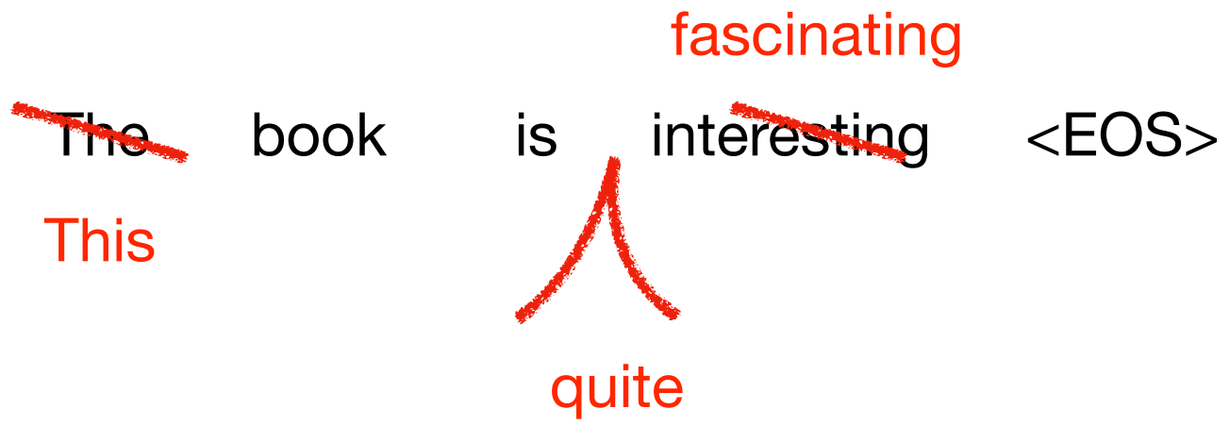
quite

Generation by Local Changes

E.g., paraphrase generation [AAAI'19, ACL'20, NeurIPS'20]

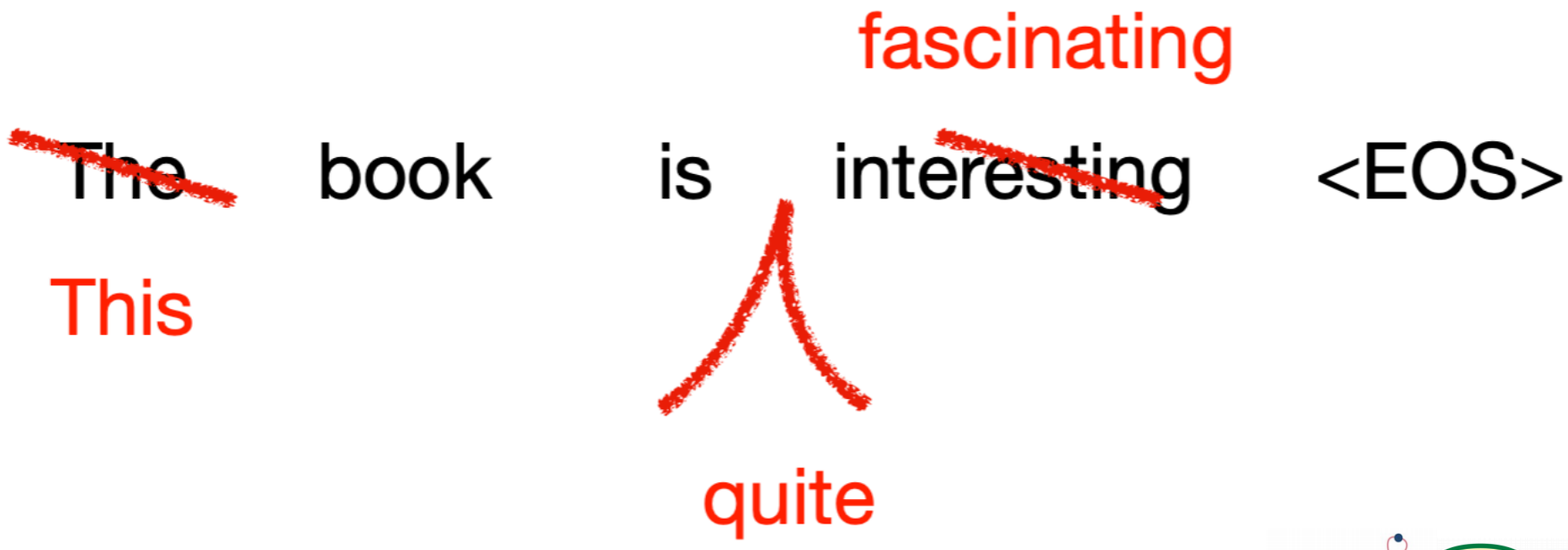
- Suppose we have a blueprint

The book is interesting <EOS>



Advantages

- Transforming generation to scoring
- Flexibility in defining the scoring function
  - Fluency, semantic coherence, task-specific constraints
- Does not suffer from the error accumulation problem
  - (Errors are distributed over the entire sentence)



Successful Applications

- Unsupervised Text Generation
  - Paraphrase generation [AAAI'19, ACL'20a, NeurIPS'20]
  - Summarization [ACL'20b,'22]
  - Simplification [ACL'20c]
- Controllable text generation
  - Emotional dialogue generation [CIKM'21]
  - Improving semantic coverage [AAAI'22]
  - Length-control summarization [ACL'20b,'22]

Improving semantic coverage for data-to-text generation

[Jolly, Zhang, Dengel, Mou, AAAI’21]

Data-to-Text Generation

Input

Slot	Value
Name	The golden curry
Food	Indian
Rating	1 out of 5
Area	Riverside
Near	Café rounge
Family Friendly	Yes

Target

The golden curry is an indian restaurant. it has a customer rating of 1 out of 5. it is family friendly. it is located near café rouge

Data-to-Text Generation

Input

Slot	Value
Name	The golden curry
Food	Indian
Rating	1 out of 5
Area	Riverside
Near	Café rounge
Family Friendly	Yes

Target

The golden curry is an indian restaurant. it has a customer rating of 1 out of 5. it is family friendly. it is located near café rouge

Our setting: Few-shot

- 100~400 samples
- Requiring less annotation
- More challenging

amii UNIVERSITY OF ALBERTA

Data-to-Text Generation

Input

Slot	Value
Name	The golden curry
Food	Indian
Rating	1 out of 5
Area	Riverside
Near	Café rounge
Family Friendly	Yes

Target

The golden curry is an indian restaurant. it has a customer rating of 1 out of 5. it is family friendly. it is located near café rouge

T5 (finetuned) output

the gold curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Problem:

Low semantic coverage

Semantic Coverage

		Hard Coverage	Soft Coverage
T5	p:42K	97.50%	97.38%
T5	p:2100	96.21%	96.28%
T5	p:420	84.19%	84.32%
T5 self-train	p:420, u:1680	81.82%	82.03%
T5 S&L	p:420, u:1680		

Secret

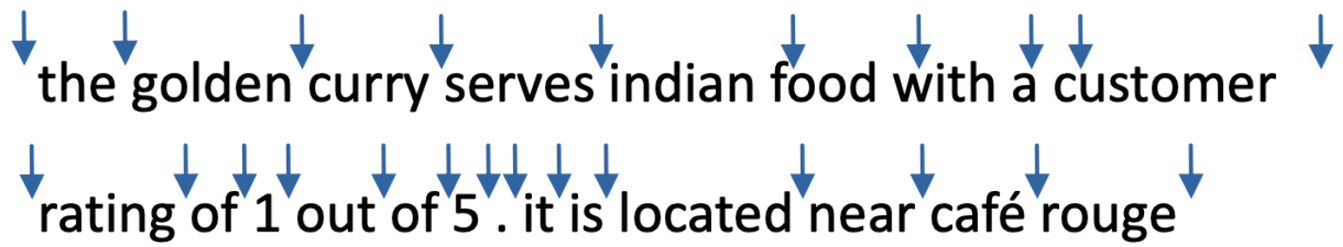
Finetuning T5

- Easy to get plausible text
- Difficult to learn the input—output correspondence

amii UNIVERSITY OF ALBERTA

Our Idea

Insert “in riverside area”



Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

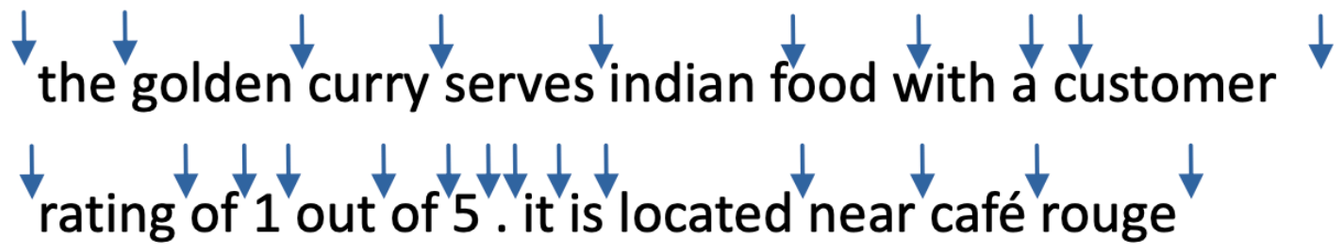
Iteratively insert all missing slots

. . .

Basic idea: Insert the missing slots

Our Idea

Insert “in riverside area”



Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Iteratively insert all missing slots . . .

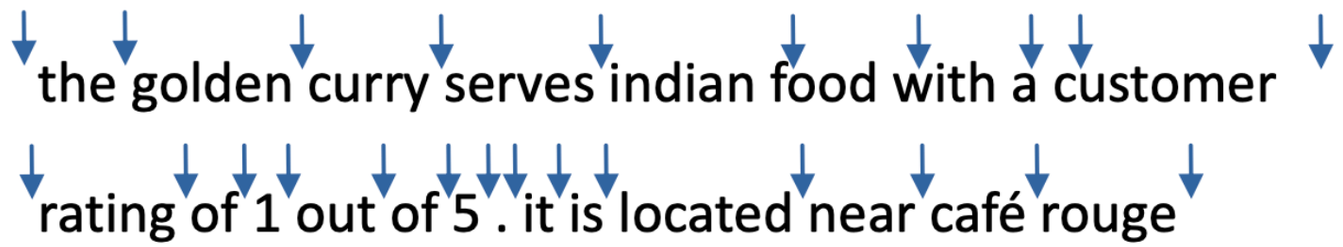
Step I: Determining the missing slot(s)

- String match or soft match (similar performance)

amii UNIVERSITY OF ALBERTA

Our Idea

Insert “in riverside area”



Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Iteratively insert all missing slots ...

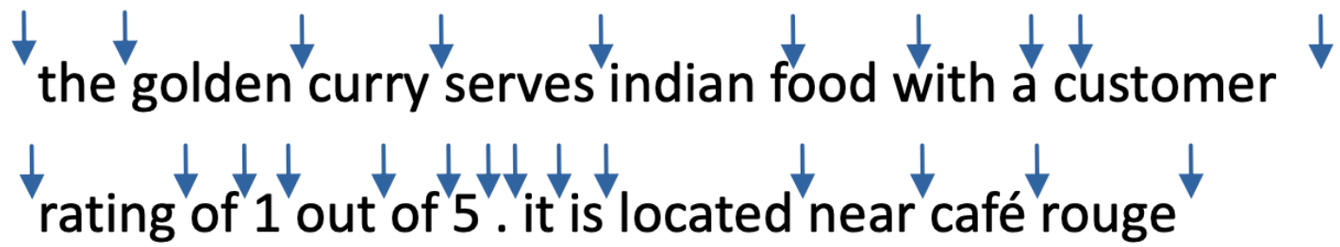
Step II: Insert the slot information

- Construct a phase `area[riverside]` → `in riverside area`
- Find the best place based on T5 probability $P(\mathbf{y}|\mathbf{T})$

amii UNIVERSITY OF ALBERTA

Our Idea

Insert “in riverside area”



Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Iteratively insert all missing slots ...

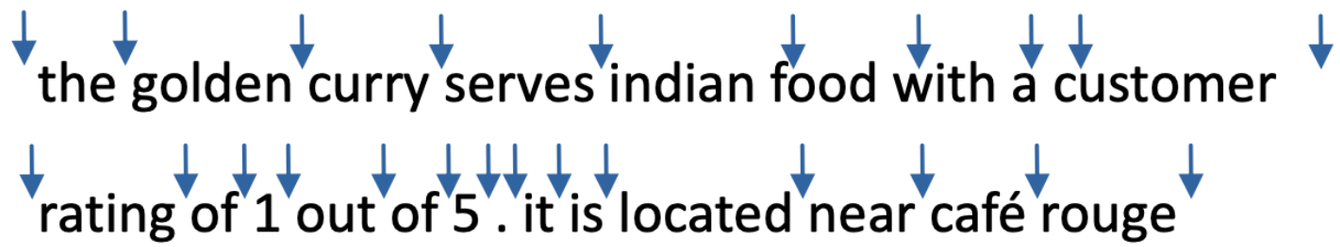
Step III: Repeat for every missing slot

- Greedy search to $\text{maximize } P(\mathbf{y}|\mathbf{T})$
  $\text{subject to } v_i \in \mathbf{y}, \forall i$
- Gradually project an infeasible solution to a feasible one

amii UNIVERSITY OF ALBERTA

Wait, is the sentence fluent?

Insert “in riverside area”



Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Iteratively insert all missing slots . . .

Unfortunately, no.

T5 search for inference: the phoenix is a restaurant that serves indian food in the price range of £20-25. it is near crowne plaza high customer rating in riverside area not family friendly hotel. (All slots are present, but the sentence is not fluent)

amii UNIVERSITY OF ALBERTA

Next idea

(a) Input data

Slot	Value
Name	The golden curry
Food	Indian
Rating	1 out of 5
Area	Riverside
Near	Café rounge
Family Friendly	Yes

(b) Reference: The golden curry is a indian restaurant. it has a customer rating of 1 out of 5. it is family friendly. it is located near café rouge.

(c) T5 output: the gold curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge. (**Missing slots**: riverside, family friendly)

(d) Search to improve semantic coverage

Insert "in riverside area"

the golden curry serves indian food with a customer

rating of 1 out of 5 . it is located near café rouge

Select the best position

in riverside area, the golden curry serves indian food with a customer rating of 1 out of 5. it is located near café rouge.

Iteratively insert all missing slots . . .

(e) Learn from search results by fine-tuning T5

- Smooth out noise
- Improve inference efficient

amii UNIVERSITY OF ALBERTA

Main Results

E2E Dataset

Model	#Train	BLEU	NIST	METEOR	RougeL	CIDEr	PARENT (P/R/F1)	PPL	AvgLen	Hard Coverage	SER	Soft Coverage
TGEN	p:42K	65.93	8.61	44.83	68.50	2.23	–	–	–	–	4.27%	95.73%
SLUG	p:42K	66.19	8.61	44.54	67.72	–	–	–	–	–	–	–
S_1^R (Shen et al. 2019)	p:42K	**68.60**	8.73	45.25	70.82	2.37	–	–	–	–	–	–
T5	p:42K	67.59	**8.81**	**45.17**	**70.44**	**2.33**	67.40 / 61.75 / 63.43	154.49	23.58	97.50%	2.62%	97.38%
T5	p:2100	62.45	8.30	44.10	67.15	2.17	64.25 / 61.69 / 62.00	136.54	24.82	96.21%	3.72%	96.28%
T5	p:420	**61.72**	7.96	40.52	65.61	1.96	65.63 / 57.25 / 60.10	**141.61**	21.97	84.19%	16.68%	84.32%
T5 self-train	p:420, u:1680	60.83	7.74	39.85	66.36	1.95	66.60 / 57.31 / 60.63	154.74	21.50	81.82%	17.97%	82.03%
T5 S&L	p:420, u:1680	60.70	8.13	43.60	65.84	**2.12**	**66.97 / 63.63 / 64.29**	160.40	25.01	98.35%	1.84%	98.16%
T5 S&L w/ SER	p:420, u:1680	60.89	**8.14**	**43.71**	**66.76**	2.07	65.16 / 62.97 / 63.04	170.26	25.71	**99.46%**	**0.80%**	**99.20%**

Main Results

Wikibio Dataset

Model	#Train	BLEU	PARENT (P/R/F1)	PPL	AvgLen	Coverage (Table)	Coverage (Reference)
GPT2+copy (Chen et al. 2020b)	p:100	29.5	–	–	–	–	–
GPT2+copy (our replication)	p:100	29.05	59.03 / 26.63 / 33.59	314.03	20.01	27.07%	55.27%
TableGPT2 (Gong et al. 2020)	p:100	34.5	–	–	–	–	–
T5	p:100	35.87	**65.21** / 29.59 / 38.00	219.03	17.35	38.45%	76.61%
T5 self-train (Recomb)	p:100	**36.00**	64.74 / 29.58 / 37.91	219.40	17.27	38.20%	76.01%
T5 S&L (Recomb)	p:100	35.41	64.10 / **30.23** / **38.34**	**218.48**	18.75	**41.29%**	**76.75%**
T5 self-train	p:100, u:400	35.62	64.68 / 29.92 / 38.19	216.19	18.17	40.22%	76.85%
T5 S&L	p:100, u:400	**35.92**	64.56 / **32.28** / **40.27**	**211.35**	19.84	**42.45%**	**79.14%**
T5 S&L (cosine similarity)	p:100, u:400	35.44	63.63 / 31.32 / 39.36	233.18	18.92	41.28%	75.68%

Not every slot should appear in the target.

We threshold the slots by frequency

amii UNIVERSITY OF ALBERTA

Human Evaluation

Model	Coverage	Fluency	Overall Quality
T5 w/ self-train	81.66%	**2.88±0.32**	2.1±0.34
T5 w/ S&L	**99.58%**	2.75±0.43	**2.81±0.39**
p-value	6.08e-24	0.00664	3.84e-22

Seach-and-Learning VS Search-only

Model	PARENT(P/R/F1)	InfTime	RelTime	PPL
S&L	**66.97/63.63/64.29**	**78.05**	**1x**	**160.40**
Search for inf.	65.71/60.17/61.67	113.4	1.45x	234.19

Table 4: Search and learning vs. search for inference. Inference time (in seconds) and Relative time were obtained by predicting the test set on a single V100 GPU.

amii UNIVERSITY OF ALBERTA

Case Study

Input table

Slot	Value
Name	The Phoenix
Eat type	Restaurant
Food	Indian
PriceRange	£ 20-25
Customer Rating	High
Area	Riverside
Near	Crowne plaza hotel
Family Friendly	No

Reference 1: the phoenix is a restaurant that also serves indian food priced between £20-25, located near crowne plaza hotel on the riverside. it's customer rating is high, and the establishment is kids friendly. (All slots are present)

Reference 2: the phoenix, located near crowne plaza hotel on the riverside, is a restaurant that also serves indian food. it is kids friendly and food is priced between £20-25. (Missing slot: customer rating)

T5 few-shot fine-tuned: the phoenix is a restaurant that serves indian food in the price range of £20-25. it is near crowne plaza hotel. it has a high customer rating. (Missing slots: riverside, family friendly)

T5 self-train: the phoenix is a restaurant that serves indian food in the price range of £20-25. it is near crowne plaza hotel. (Missing slots: high, riverside, family friendly)

T5 search for inference: the phoenix is a restaurant that serves indian food in the price range of £20-25. it is near crowne plaza high customer rating in riverside area not family friendly hotel. (All slots are present, but the sentence is not fluent)

T5 S&L: in riverside area the phoenix is a restaurant that serves indian food in the price range of £20-25. it is near crowne plaza hotel. it has a high customer rating and is not family-friendly. (All slots are present)

amii UNIVERSITY OF ALBERTA

Controlling the length for summarization

[Schumann, Mou, Lu, Vechtomova, Markert, ACL'20]

[Liu, Huang, Mou, ACL'22]

Summarization Task

- Applications: headline generation
- Granularities:
  - Single-document summarization
  - Multi-document summarization
  - Sentence-level summarization 
    Generate summaries for an input sentence

Example: The amphibia, which is the animal class to which our frogs and toads belong, were the first animal to crawl from the sea and inhabit the earth -> The first animals to leave the sea and live on land were the amphibia.

Unsupervised Summarization

- Supervised methods
  - State-of-the-art
  - Data-hungry
  - Inapplicable to less popular domains
- Our work considers unsupervised summarization
  - Word extraction with order preserved

amii UNIVERSITY OF ALBERTA

Search-Based Method

- Defining a heuristic score
  - Fluency, semantic coherence, length constraint
- Maximize the score by local editing
  - Swapping the section/non-selection of two words

Search Objective

- Fluency

$$\overleftrightarrow{\mathrm{PPL}}(\mathbf{y}) = \sqrt[2|\mathbf{y}|]{\prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overrightarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{<i})} \prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overleftarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{>i})}}.$$

- Semantic coherence
- Length constraint

Search Objective

- Fluency

$$\overleftrightarrow{\mathrm{PPL}}(\mathbf{y}) = \sqrt[2|\mathbf{y}|]{\prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overrightarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{<i})} \prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overleftarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{>i})}}.$$

- Semantic coherence

$$s_{semantic} = \mathrm{normalize}[\cos(\boldsymbol{e}(\mathbf{y}), \boldsymbol{e}(\mathbf{x}))]$$

- Length constraint

Search Objective

- Fluency

$$\overleftrightarrow{\mathrm{PPL}}(\mathbf{y}) = \sqrt[2|\mathbf{y}|]{\prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overrightarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{<i})} \prod_{i}^{|\mathbf{y}|} \frac{1}{p_{\overleftarrow{\mathrm{LM}}}(y_i|\mathbf{y}_{>i})}}.$$

- Semantic coherence

$$s_{semantic} = \mathrm{normalize}[\cos(\boldsymbol{e}(\mathbf{y}), \boldsymbol{e}(\mathbf{x}))]$$

- Length constraint
  - Key to the application
  - Fair comparison

$$f_{\mathrm{LEN}}(\mathbf{y}; s) = \begin{cases} 1, & \text{if } |\mathbf{y}| = s, \\ -\infty, & \text{otherwise.} \end{cases}$$

Search Operation

- Always select the correct number of words
- Swapping two words at each search step
    - Much simpler than generation from the vocabulary

$$|\mathscr{V}|^{|\mathbf{y}|} \Longrightarrow \begin{pmatrix} |\mathbf{x}| \\ S \end{pmatrix}$$

Search Algorithm

- Greedy hill-climbing
  - Good enough for the simple search problem

Start with $\mathbf{y}_0$ # an initial candidate sentence

Loop within budget at step t:

$\mathbf{y}' \sim \text{Neighbor}(\mathbf{y}_t)$ # a new candidate in the neighbor

Either reject or accept $\mathbf{y}'$ whenever $\mathbf{y}'$ is better than $\mathbf{y}_{t-1}$

If accepted, $\mathbf{y}_t = \mathbf{y}'$, or otherwise $\mathbf{y}_t = \mathbf{y}_{t-1}$

Return the best scored $\mathbf{y}_*$

Main Results

	Model	Data			Len D	Rouge F1			Len O
		article	title	external		R-1	R-2	R-L	
	Lead-N-8	✓			8	21.39	7.42	20.03	7.9
A	*HC_article_8*	✓			8	23.09	7.50	21.29	7.9
	HC_title_8		✓		8	**26.32**	**9.63**	**24.19**	7.9
	Lead-N-10	✓			10	23.03	7.95	21.29	9.8
	Wang and Lee (2018)	✓	✓		-	27.29	10.01	24.59	10.8
	Zhou and Rush (2019)		✓	billion	-	26.48	10.05	24.41	9.3
B	*HC_article_10*	✓			10	24.44	8.01	22.21	9.8
	HC_title_10		✓		10	27.52	10.27	24.91	9.8
	HC_title+twitter_10		✓	twitter	10	28.26	10.42	25.43	9.8
	HC_title+billion_10		✓	billion	10	**28.80**	**10.66**	**25.82**	9.8
	Lead-P-50	✓			50%	24.97	8.65	22.43	14.6
	Fevry and Phang (2018)	✓		SNLI	50%	23.16	5.93	20.11	14.8
C	Baziotis et al. (2019)	✓			50%	24.70	7.97	22.14	15.1
	HC_article_50p	✓			50%	25.58	8.44	22.66	14.9
	HC_title_50p		✓		50%	**27.05**	**9.75**	**23.89**	14.9

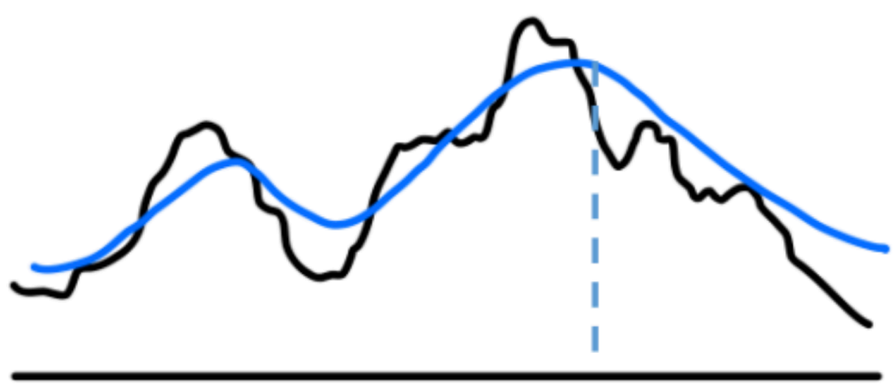
amii UNIVERSITY OF ALBERTA

Main Disadvantages

- Slow inference (1000x slower than seq2seq)
- Search noise
  - Objective function may be imperfect
  - Optimization leads to suboptimal

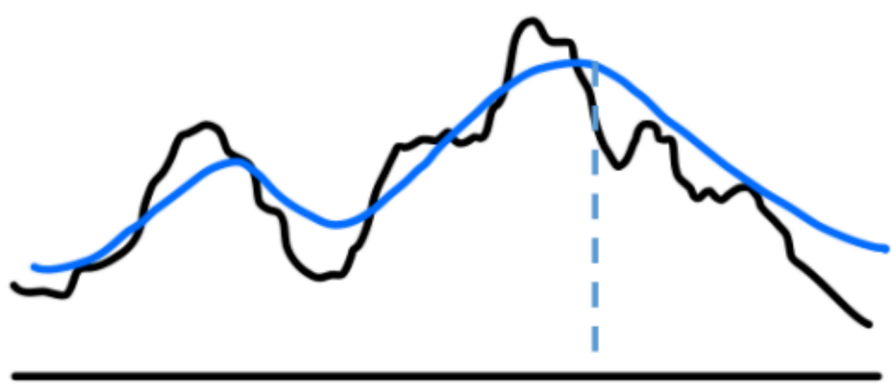
Main Disadvantages

- Slow inference (1000x slower than seq2seq)
- Search noise
  - Objective function may be imperfect
  - Optimization leads to suboptimal
- Idea: Learning from search results
  - Much faster
  - Smoothing out search noise



Main Disadvantages

- Slow inference (1000x slower than seq2seq)
- Search noise
  - Objective function may be imperfect
  - Optimization leads to suboptimal
- Idea: Learning from search results
  - Much faster
  - Smoothing out search noise
  - But, length constraint may be violated

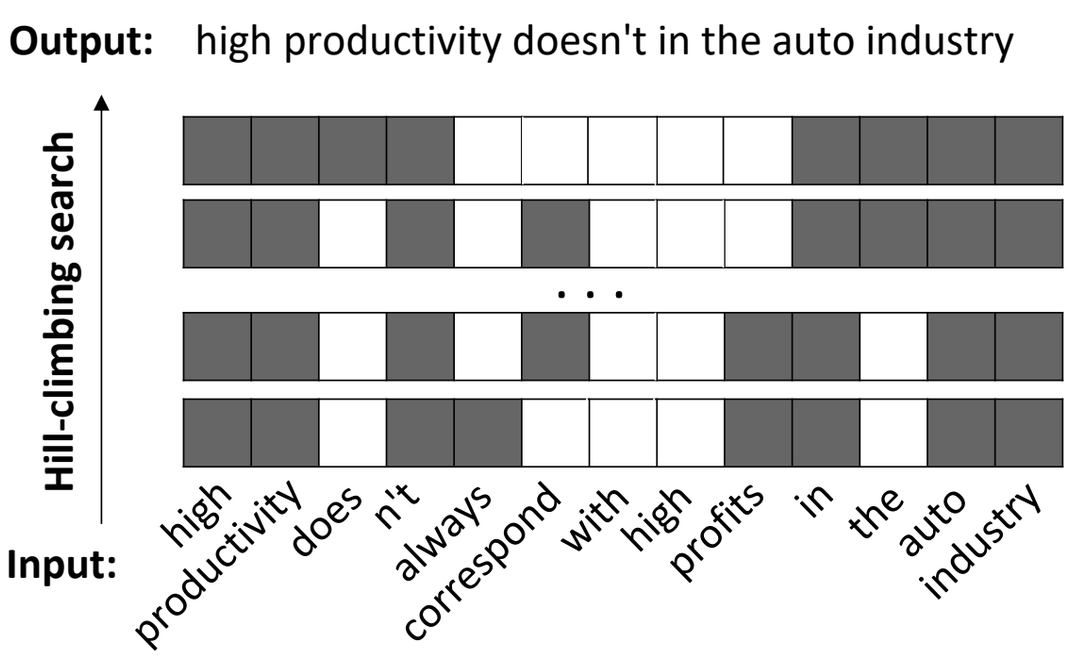


→ More dedicated learning models are needed

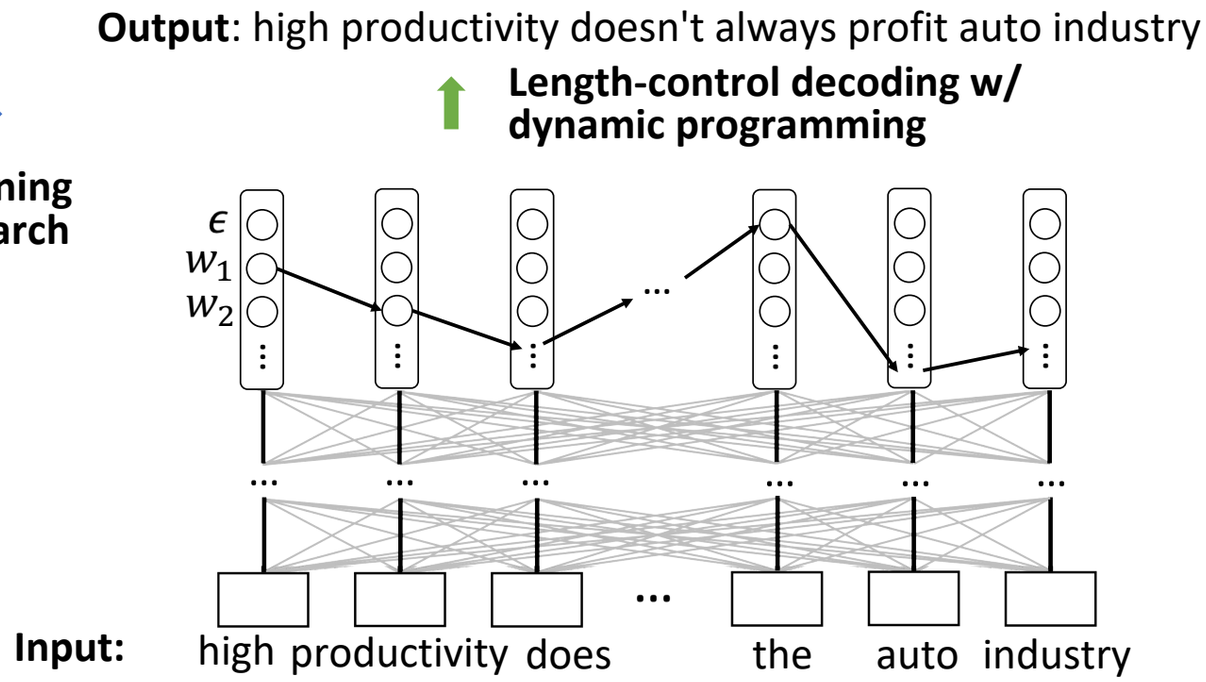
amii UNIVERSITY OF ALBERTA

Our Approach

- Non-autoregressive (NAR) generation
  - All words are predicted simultaneously
  - Even more efficient than an AR model

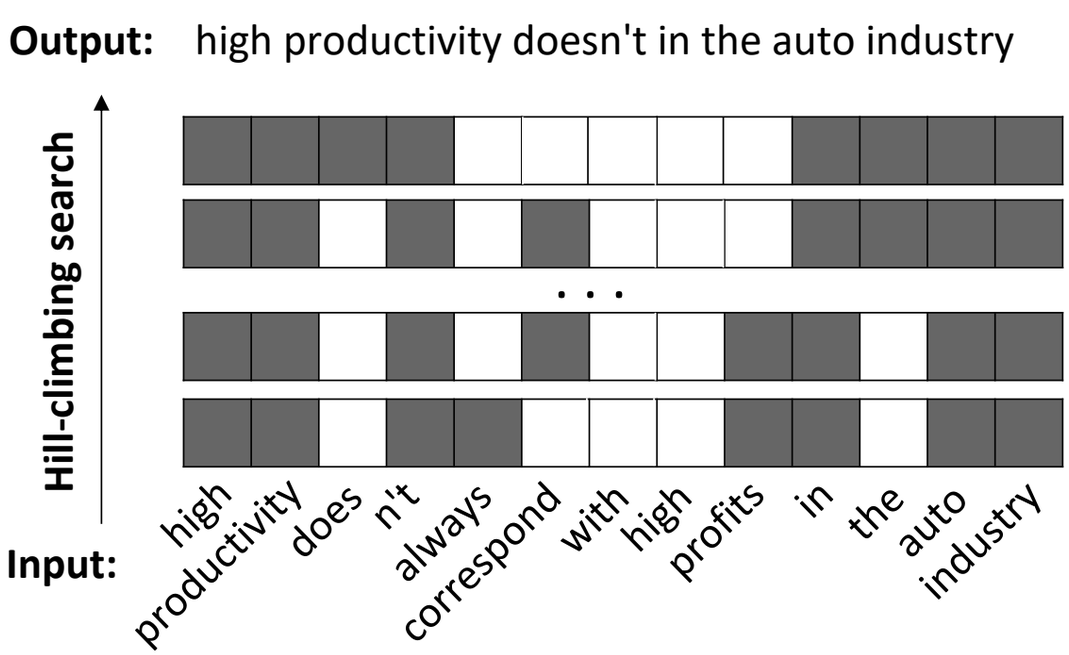


CTC training from search results

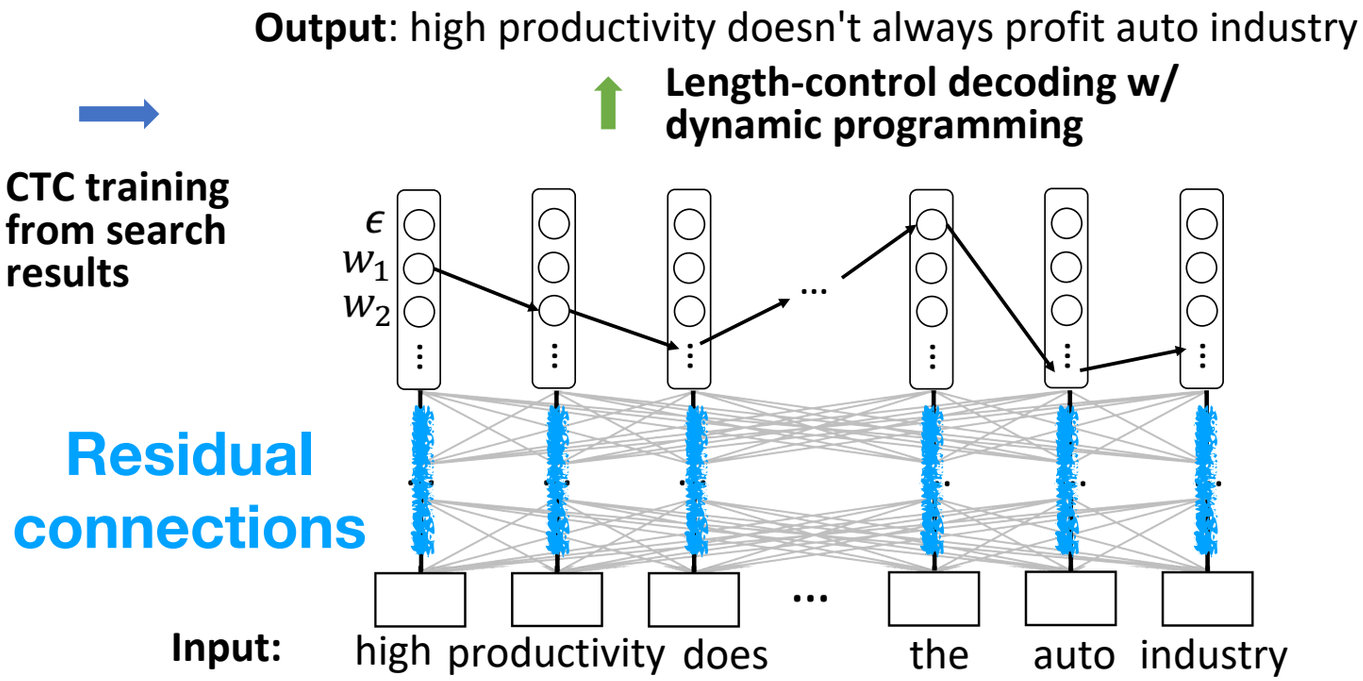


Our Approach

- Connectionist Temporal Classification (CTC) Training
  - Allowing empty token ϵ scattering over the sentence
  - Maintaining input—output correspondence



CTC training from search results



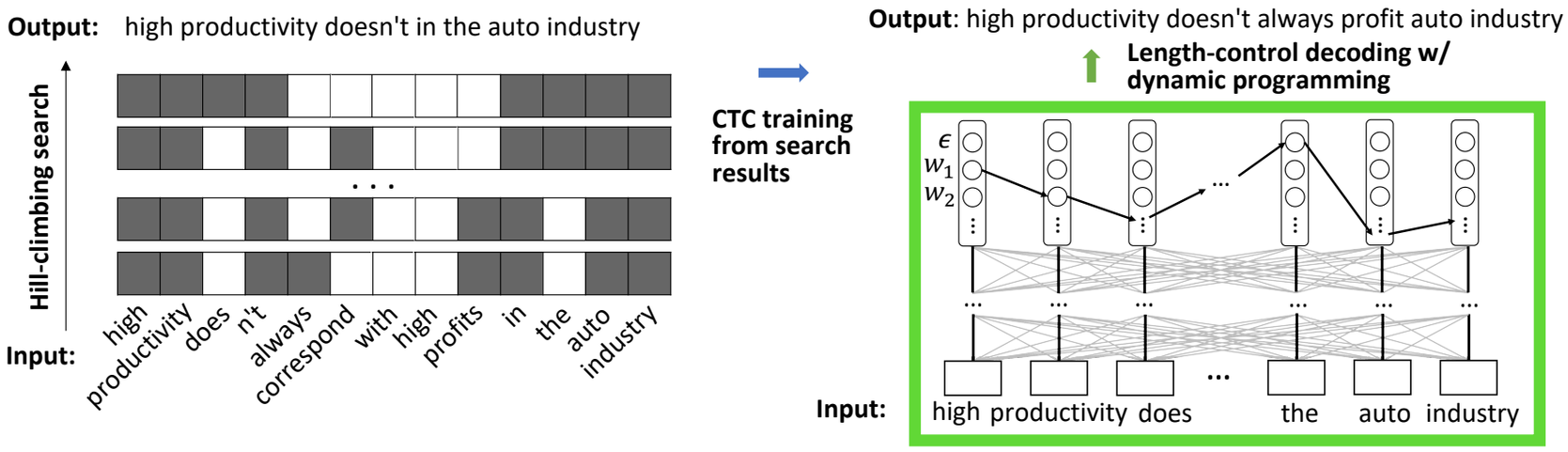
Our Approach

- Connectionist Temporal Classification (CTC) training
  - Including an empty token ϵ
  - Shorter generation as summary

Example: I ϵ like reading ϵ ϵ ϵ ϵ books

$\Rightarrow$ I like reading books

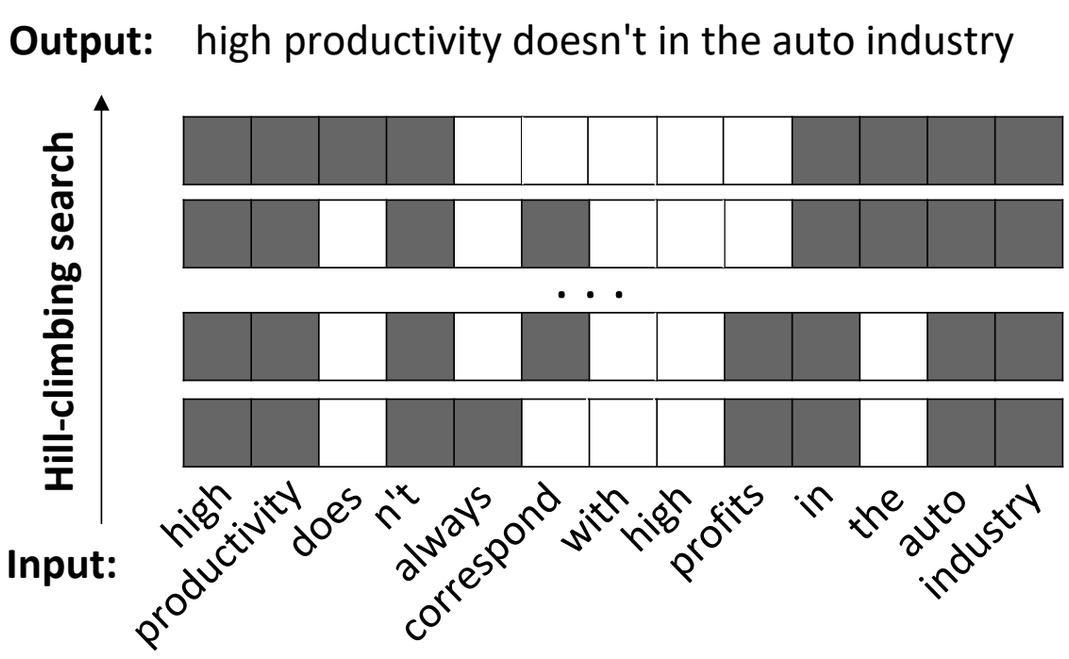
- Training objective: MLE $\sum_{\mathbf{w}:\Gamma(\mathbf{w})=\mathbf{y}} P(\mathbf{w}|\mathbf{x})$
  - Computed by dynamic programming



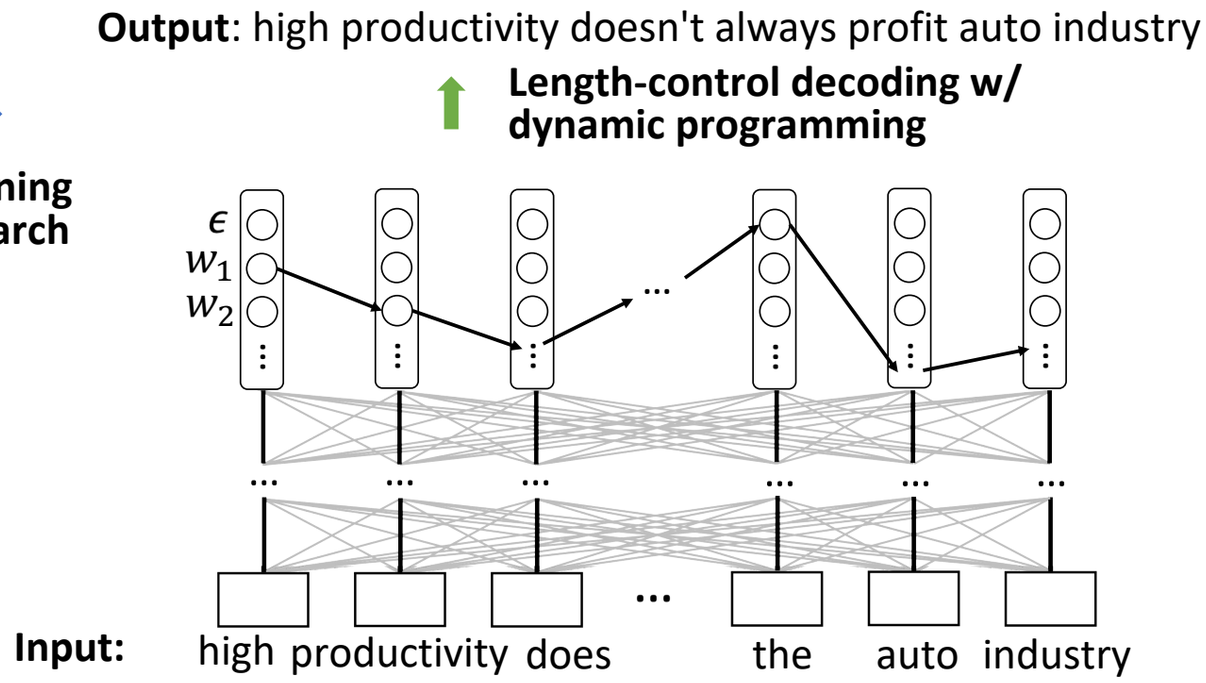
Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

Our Approach

- Our insights
  - NAR predicts “local” probabilities
  - Dynamic programming is possible
    - Predicted tokens VS length budget

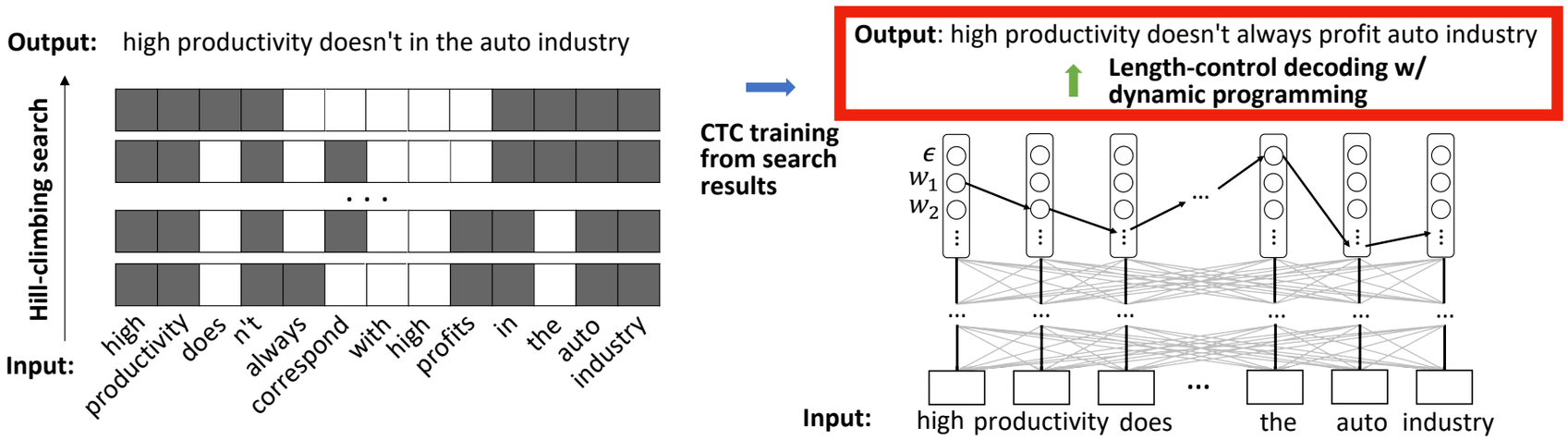


CTC training from search results



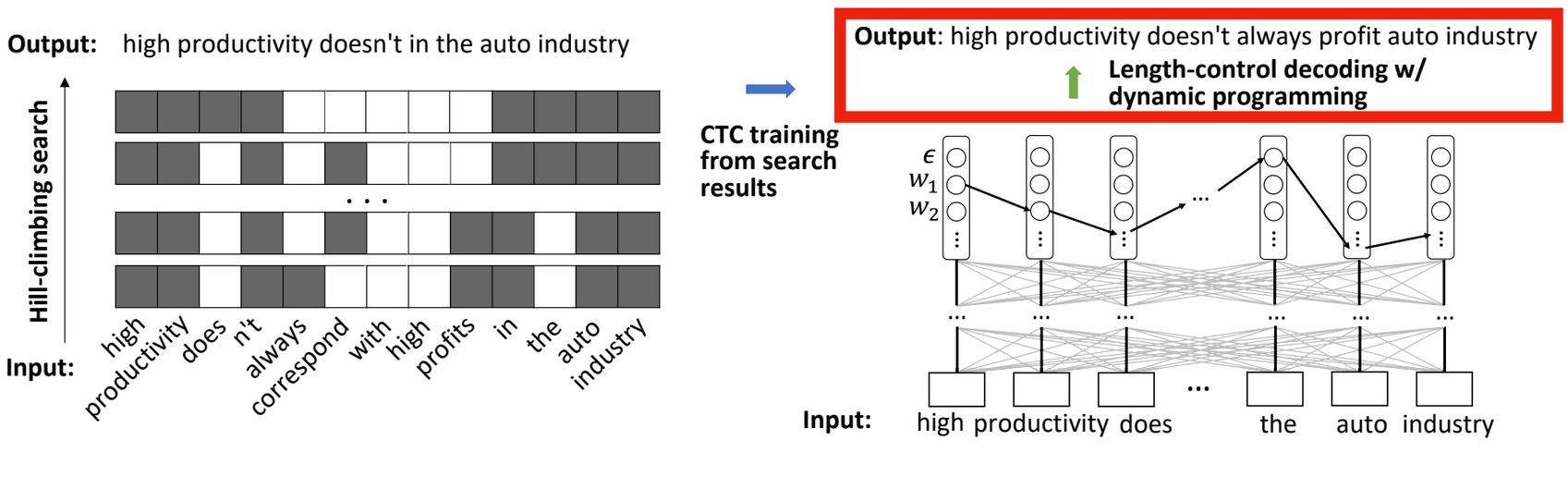
Our Approach

- Length-control algorithm



Our Approach

- Length-control algorithm
  - Dynamic Programming (DP) algorithm
  - Recursion variable
    $\mathscr{B}_{s,t}$: most probable t-word summary given the first s model outputs.
  - Base case
    $\mathscr{B}_{1,0}$: empty summary of a ϵ token
    $\mathscr{B}_{1,1}$: most probable word in the first output

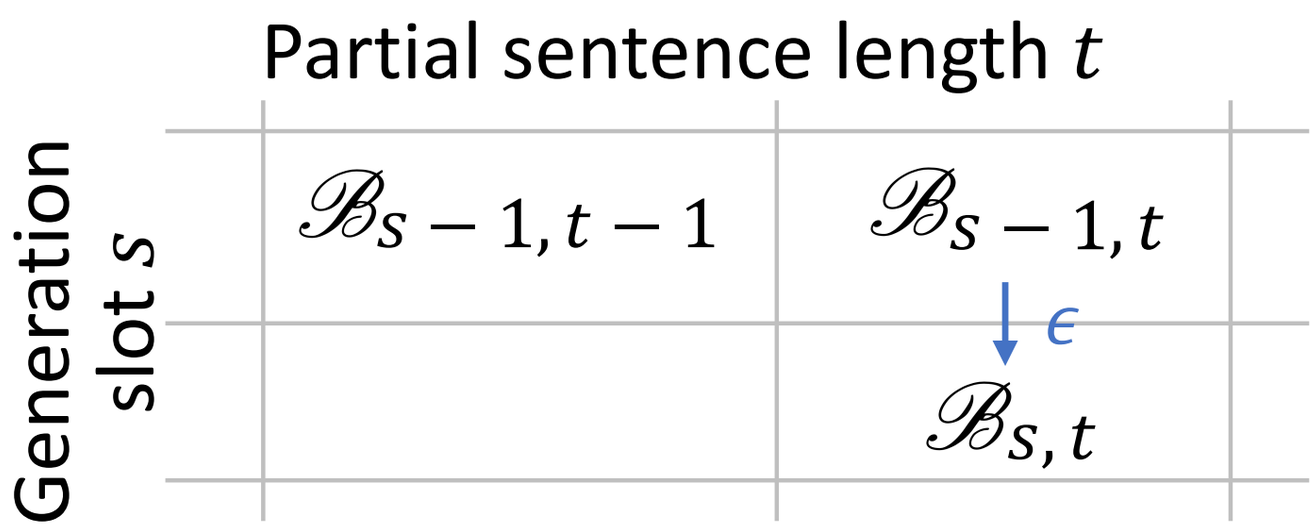


DP Recursion

$\mathscr{B}_{s,t}$: most probable t-word summary given the first s model outputs.

- Case 1 [next word is ϵ]

$$\mathscr{B}^{(1)}_{s,t} = \left\{ \mathbf{b} \oplus \epsilon \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t} \right\}$$



DP Recursion

$\mathscr{B}_{s,t}$: most probable t-word summary given the first s model outputs.

- Case 1

$$\mathscr{B}^{(1)}_{s,t} = \left\{ \mathbf{b} \oplus \epsilon \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t} \right\}$$

- Case 2

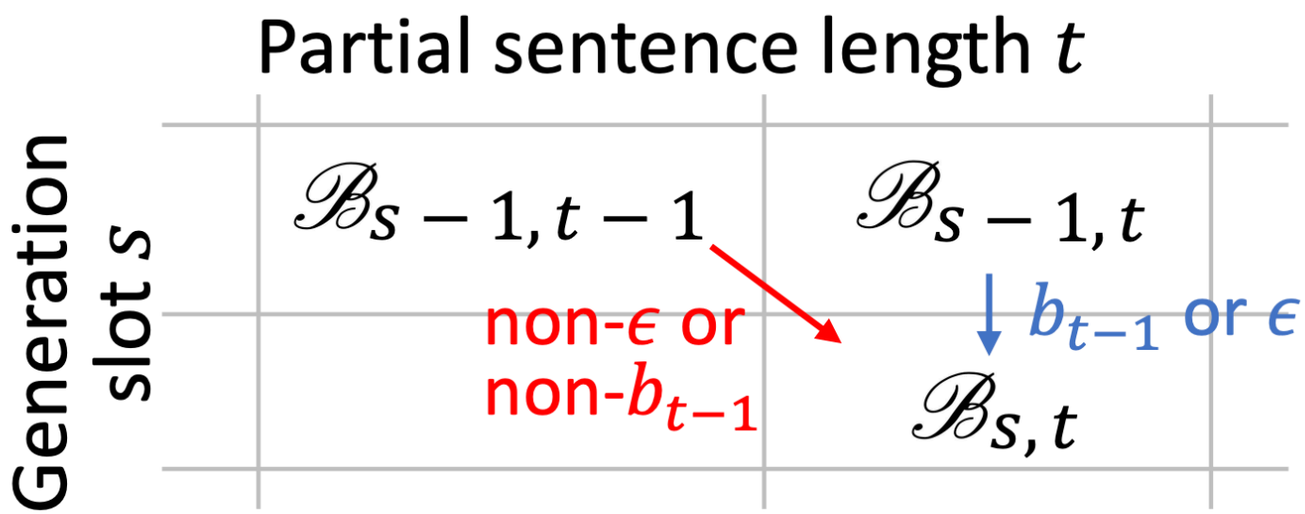
$$\mathscr{B}^{(2)}_{s,t} = \left\{ \mathbf{b} \oplus \mathrm{b}_{s-1} \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t} \right\}$$

- Case 3

$$\mathscr{B}^{(3)}_{s,t} = \max \left\{ \mathbf{b} \oplus \mathrm{w} \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t-1}, \mathrm{w}_s \neq \epsilon, \mathrm{w}_s \neq \mathrm{b}_{s-1} \right\}$$

- Take the max

$$\mathscr{B}_{s,t} = \max(\mathscr{B}^{(1)}_{s,t} \cup \mathscr{B}^{(2)}_{s,t} \cup \mathscr{B}^{(3)}_{s,t})$$



Algorithm Analysis

- CTC merges repetitive tokens
  - `I like like like reading` ⇒ `I like reading`

- Theorem
  - Not merging repetitions ⇒ DP is exact
  - Merging repetitions ⇒ DP is approximate as it establishes dependencies between DP steps

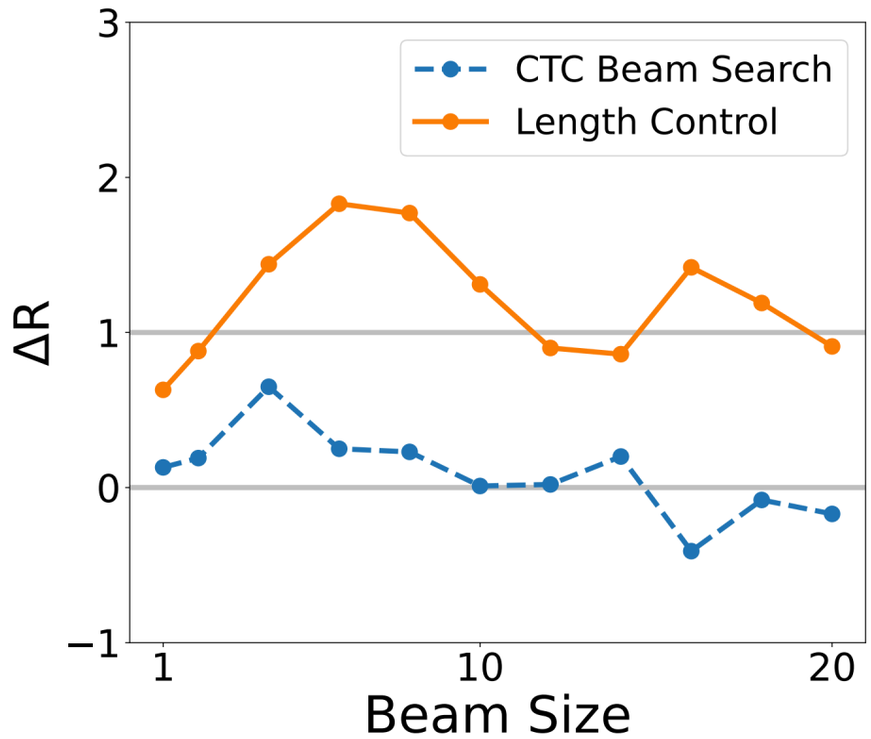
Theorem 1. *(1) If repeating tokens are not merged, then the proposed length-control algorithm with beam size* $B = 1$ *finds the exact optimum* $\mathscr{B}_{S,T}$ *being the most probable length-*T *sentence given by* S *prediction slots. (2) If we merge repeating tokens predicted by CTC-trained models, the above algorithm may not be exact.*

amii UNIVERSITY OF ALBERTA

Algorithm Analysis

- Enhancing DP with beam search

$\mathscr{B}_{s,t}$: ~~most~~ top-K probable t-word summaries given the first s model outputs.



$$\mathscr{B}_{s,t}^{(1)} = \left\{ \mathbf{b} \oplus \epsilon \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t} \right\}$$

$$\mathscr{B}_{s,t}^{(2)} = \left\{ \mathbf{b} \oplus \mathrm{b}_{s-1} \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t} \right\}$$

$$\mathscr{B}_{s,t}^{(3)} = \text{~~max~~}^{\text{top-K}} \left\{ \mathbf{b} \oplus \mathrm{w} \ : \ \mathbf{b} \in \mathscr{B}_{s-1,t-1}, \mathrm{w}_s \neq \epsilon, \mathrm{w}_s \neq \mathrm{b}_{s-1} \right\}$$

$$\mathscr{B}_{s,t} = \text{~~max~~}^{\text{top-K}} \left(\mathscr{B}_{s,t}^{(1)} \cup \mathscr{B}_{s,t}^{(2)} \cup \mathscr{B}_{s,t}^{(3)} \right)$$

amii UNIVERSITY OF ALBERTA

Experiments

- Gigaword

Group	#	Approach		Len	ROUGE F1				Inf.Time	Speedup
					R-1	R-2	R-L	ΔR		
A (desired length 8)	1	Baseline	Lead (8 words)†	7.9	21.39	7.42	20.03	-11.12	–	–
	2	Search	Schumann et al. (2020)†	7.9	26.32	9.63	24.19	0.18	–	–
	3		Our replication	7.9	26.17	**9.69**	24.10	0	6.846	1x
	4	Learn from search	Su et al. (2021)	7.7	26.88	9.37	24.54	0.83	0.017	403x
	5		NAUS (truncate)	7.8	27.27	9.49	24.96	1.76	**0.005**	**1369x**
	6		NAUS (length control)	7.8	**27.94**	9.24	**25.51**	**2.73**	0.041	167x
B (desired length 10)	7	Baseline	Lead (10 words)†	9.8	23.03	7.95	21.29	-10.2	–	–
	8		Wang and Lee (2018)†	10.8	27.29	10.01	24.59	-0.58	–	–
	9		Zhou and Rush (2019)†	9.3	26.48	10.05	24.41	-1.53	–	–
	10	Search	Schumann et al. (2020)†	9.8	27.52	**10.27**	24.91	0.23	–	–
	11		Our replication	9.8	27.35	10.25	24.87	0	9.217	1x
	12	Learn from search	Su et al. (2021)	9.4	27.86	9.88	25.51	0.78	0.020	461x
	13		NAUS (truncate)	9.8	28.24	10.04	25.40	1.21	**0.005**	**1843x**
	14		NAUS (length control)	9.8	**28.55**	9.97	**25.78**	**1.83**	0.044	210x

Table 1: Results on the Gigaword headline generation test set. **Len:** Average length of predicted summaries. **R-1, R-2, R-L:** ROUGE-1, ROUGE-2, ROUGE-L. **ΔR:** The difference of total ROUGE (sum of R-1, R-2, and R-L) in comparison with the (previous) state-of-the-art search method under replication. **Inf.Time:** Average inference time in seconds for one sample on an i9-9940X CPU and a RTX6000 GPU. **Speedup:** Relative to Schumann et al. (2020). †Results quoted from previous papers; others are given by our experiments.

Experiments

- DUC2004

Model	ROUGE Recall				Time	Speedup
	R-1	R-2	R-L	ΔR		
Lead (75 characters)†	22.50	6.49	19.72	-8.34	–	–
Zajic et al. (2004)†	25.12	6.46	20.12	-5.35	–	–
Baziotis et al. (2019)†	22.13	6.18	19.30	-9.44	–	–
West et al. (2019)†	22.85	5.71	19.87	-8.62	–	–
Schumann et al. (2020)†	26.04	**8.06**	22.90	-0.05	–	–
Our replication	26.14	8.03	22.88	0	12.314	1x
Su et al. (2021)	26.25	7.66	22.83	-0.31	0.022	559x
NAUS (truncate)	26.52	7.88	22.91	0.26	**0.005**	**2463x**
NAUS (length control)	**26.71**	7.68	**23.06**	**0.40**	0.048	257x

Table 2: Results on the DUC2004 dataset. †Quoted from previous papers.

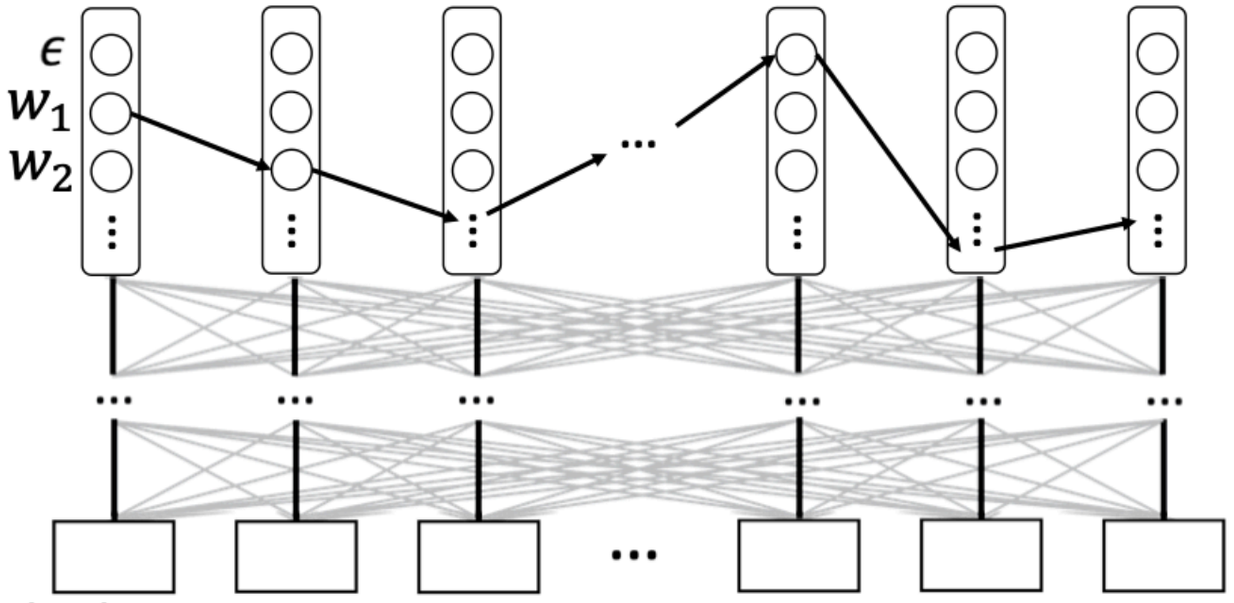
Experiments

- Human evaluation

	Decoding	Wins	Ties	Loses	p-val
Overall quality	Truncate	18.67%	40.67%	40.67%	0.0004
	Length control	**40.67%**	40.67%	**18.67%**	
Completeness & fluency	Truncate	24.67%	26.67%	48.67%	0.0005
	Length control	**48.67%**	26.67%	**24.67%**	

Table 5: Human evaluation comparing truncating and length control for our NAUS model on 50 samples in the Gigaword headline generation task. The results are statistically significant, where the p-value is given by a one-sided binomial test.

Model Analysis



- In our experiments:
  - CTC > vanilla
  - Encoder-only > encoder-decoder

#	Approach		ROUGE Recall				Speedup
			R-1	R-2	R-L	ΔR	
Group A (desired length 8)							
1	Search	Schumann et al.	26.32	9.63	24.19	0.18	–
2		Our replication	26.17	**9.69**	24.10	0	1x
3	AR	Transformer (T)	26.65	9.51	24.67	0.87	58x
4	NAR enc-dec	Vanilla	24.87	8.33	22.74	-4.02	571x
5		CTC (T)	27.30	9.20	24.96	1.5	571x
6		CTC (LC)	27.76	9.13	25.33	2.26	149x
7	NAR enc-only	Su et al. (2021)	26.88	9.37	24.54	0.83	403x
8		Our NAUS (T)	27.27	9.49	24.96	1.76	**1396x**
9		Our NAUS (LC)	**27.94**	9.24	**25.51**	2.73	167x

Length-Transfer Generation

- Long → Short

	Len	ROUGE F1				Inf.Time	Speedup
		R-1	R-2	R-L	ΔR		
Su et al. (2021)$_{8\to8}$	7.7	26.88	9.37	24.54	0.83	0.017	403x
Su et al. (2021)$_{10\to8}$	8.4	25.71	8.94	23.65	-1.84	0.018	380x
NAUS (truncate)	7.8	27.27	9.49	24.96	1.76	**0.005**	**1369x**
NAUS$_{8\to8}$	7.8	**27.94**	9.24	**25.50**	**2.73**	0.041	167x
NAUS$_{10\to8}$	7.9	27.12	9.08	24.86	1.10		

- Short → Long

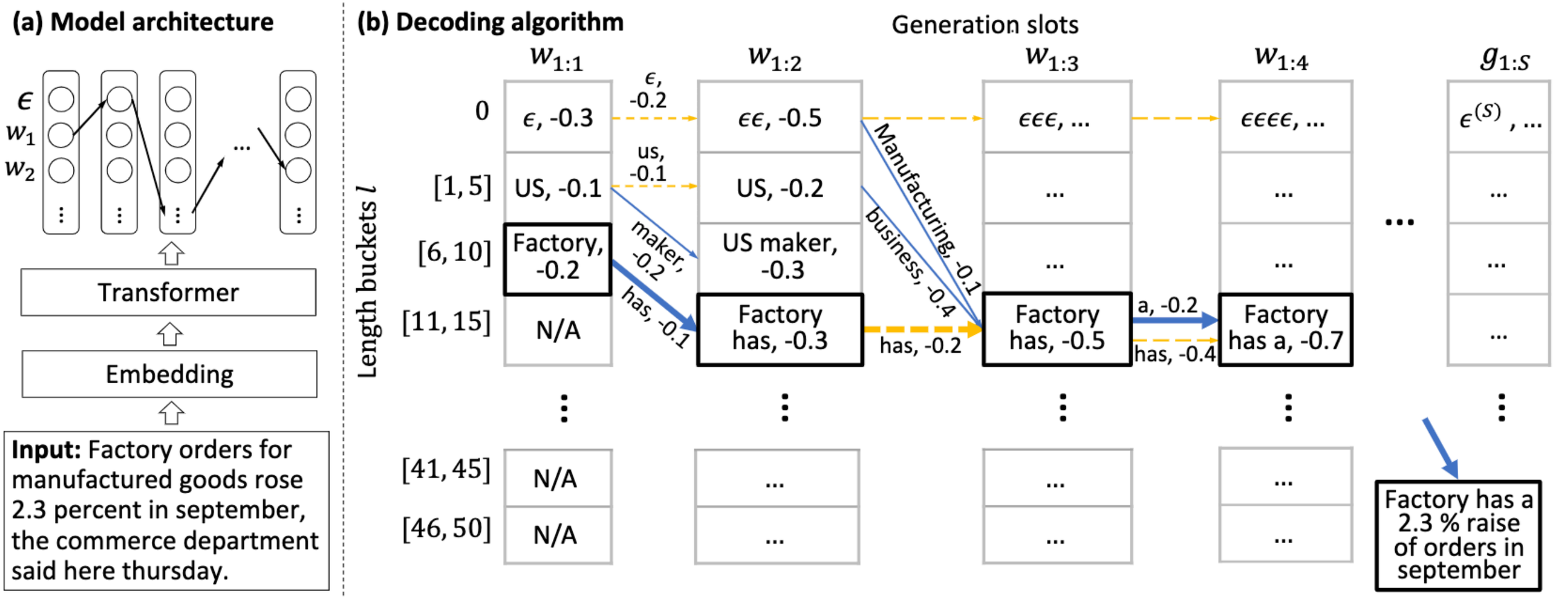
	Len	ROUGE F1				Inf.Time	Speedup
		R-1	R-2	R-L	ΔR		
Su et al. (2021)$_{8\to50\%}$	–	–	–	–	–	–	–
Su et al. (2021)$_{10\to50\%}$	–	–	–	–	–	–	–
NAUS$_{8\to50\%}$	14.9	28.39	9.78	24.94	2.48	**0.052**	**336x**
NAUS$_{10\to50\%}$	14.9	**28.53**	**9.88**	**25.10**	**2.88**		

amii UNIVERSITY OF ALBERTA

One More Extension

[arXiv:2205.14522]

- Character-level length control
  - Knapsack-like DP: value=predicted prob.; weight=#char
  - Dynamic knapsack: value changes over time



Conclusion

- Search and learning approaches
  - Local search
    - Better controlling the output
  - Learning from search
    - Improving inference efficiency
    - Smoothing out search noise
    - More dedicated design is sometimes needed

Conclusion

- Search and learning approaches
  - Local search
    - Better controlling the output
  - Learning from search
    - Improving inference efficiency
    - Smoothing out search noise
    - More dedicated design is sometimes needed
- Applications
  - Improving semantic coverage for data2text generation
  - Length control summarization

amii UNIVERSITY OF ALBERTA

Future Work

Efficient search

- Current methods
  - Task-specific search operations
  - General operations: word insertion/deletion/replacement
- Future work
  - Syntax-based
  - Improving connectivity

amii UNIVERSITY OF ALBERTA

Future Work

Efficient learning

- Current methods
  - Mainly cross-entropy loss
- Future work
  - RL-style learning for local search

Thank you!

Q&A