Tsinghua University

Constrained Text Generation: Monte-Carlo Meets Neural Nets

Lei Li

ByteDance AI Lab

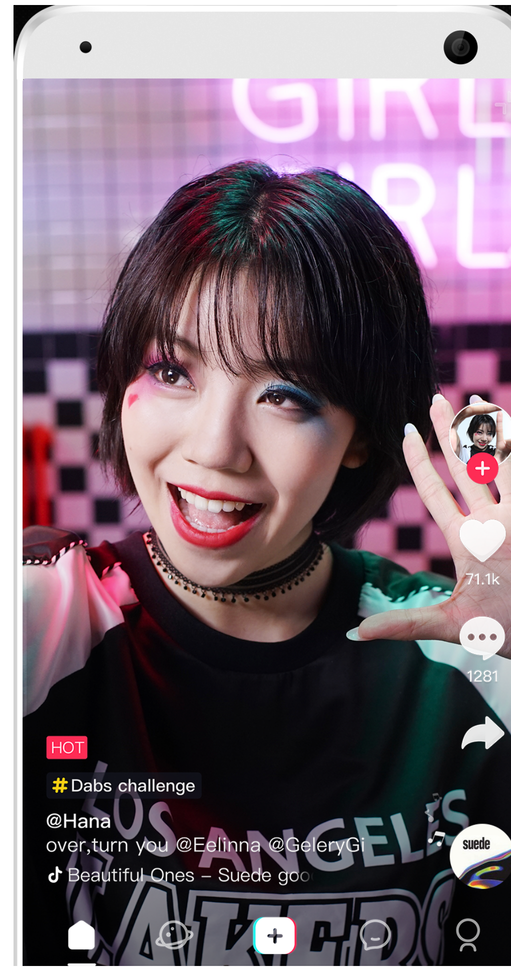
10/8/2020

The Rise of New Media Platforms

Toutiao

Helo

Douyin/Tiktok

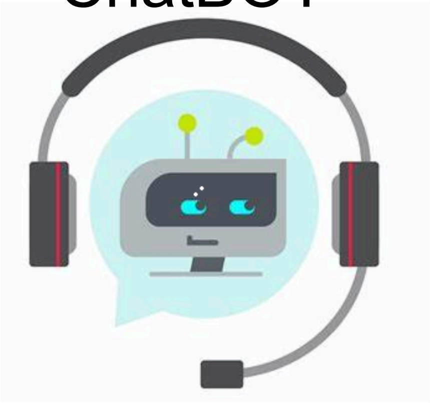


Huge Demand for NLG

Machine Writing

Question Answering

ChatBOT



Machine Translation

Machine Translation has quietly increased international trade by over 10%!

Equivalent to making the world 26% smaller!

informs
http://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 65, No. 12, December 2019, pp. 5449–5460
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Does Machine Translation Affect International Trade? Evidence from a Large Digital Platform

Erik Brynjolfsson,[a] **Xiang Hui,**[b] **Meng Liu**[b]

[a] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142; [b] Marketing, Olin School of Business, Washington University in St. Louis, St. Louis, Missouri 63130

Contact: erikb@mit.edu, http://orcid.org/0000-0002-8031-6990 (EB); hui@wustl.edu, http://orcid.org/0000-0001-7595-3461 (XH); mengl@wustl.edu, http://orcid.org/0000-0002-5512-7952 (ML)

Received: April 18, 2019
Revised: April 18, 2019
Accepted: April 18, 2019
Published Online in Articles in Advance: September 3, 2019

https://doi.org/10.1287/mnsc.2019.3388

Abstract. Artificial intelligence (AI) is surpassing human performance in a growing number of domains. However, there is limited evidence of its economic effects. Using data from a digital platform, we study a key application of AI: machine translation. We find that the introduction of a new machine translation system has significantly increased international trade on this platform, increasing exports by 10.9%. Furthermore, heterogeneous treatment effects are consistent with a substantial reduction in translation costs. Our results provide causal evidence that language barriers significantly hinder trade and that AI has already begun to improve economic efficiency in at least one domain.

History: Accepted by Joshua Gans, business strategy.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/mnsc.2019.3388.

Keywords: artificial intelligence • international trade • machine translation • machine learning • digital platforms

The New York Times

Soon a Robot Will Be Writing This Headline

Gabriel Alcala

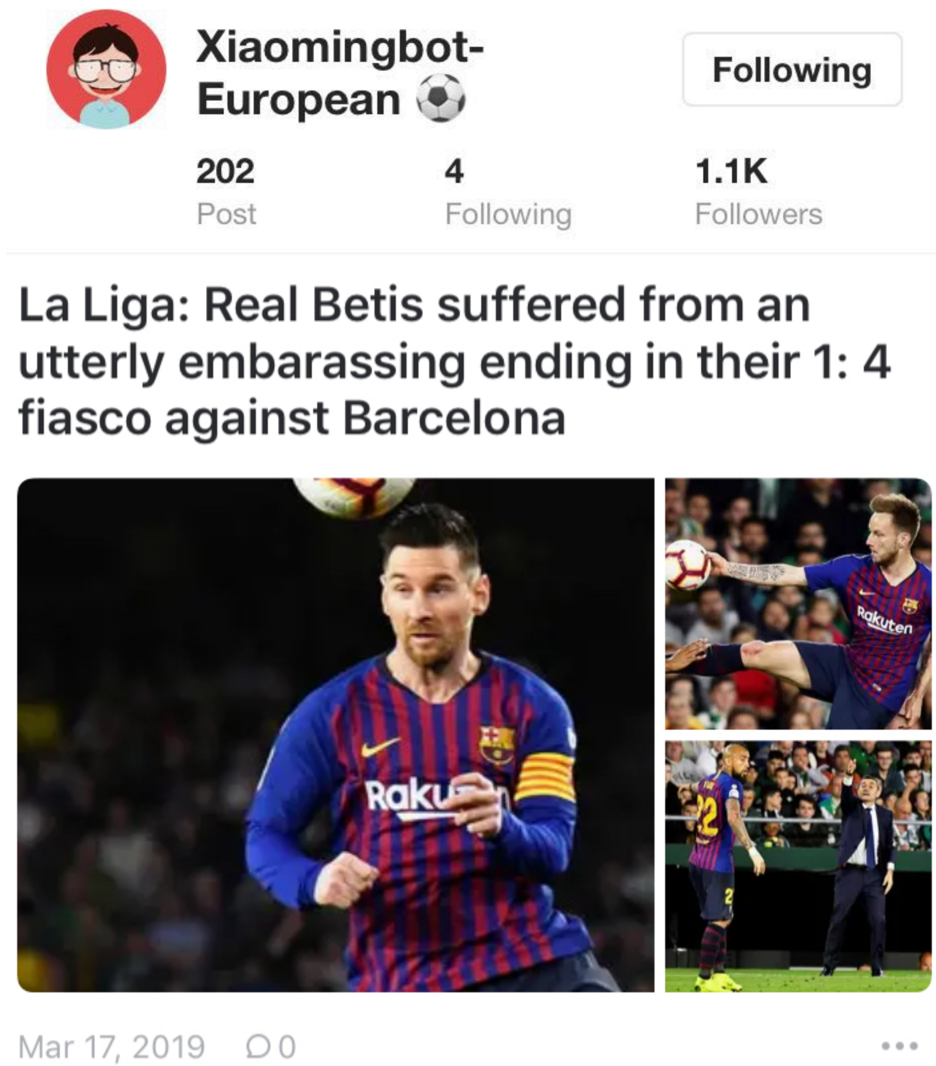
BUY BOOK ▾ When you purchase an independently reviewed book through our site, we earn an affiliate commission.

By Alana Semuels

Jan. 14, 2020

Automated News Writing

Xiaomingbot is deployed and constantly producing news on social media platforms (TopBuzz & Toutiao).



Modeling a Sequence - a Probabilistic Perspective

The quick brown fox jumps over the lazy dog .

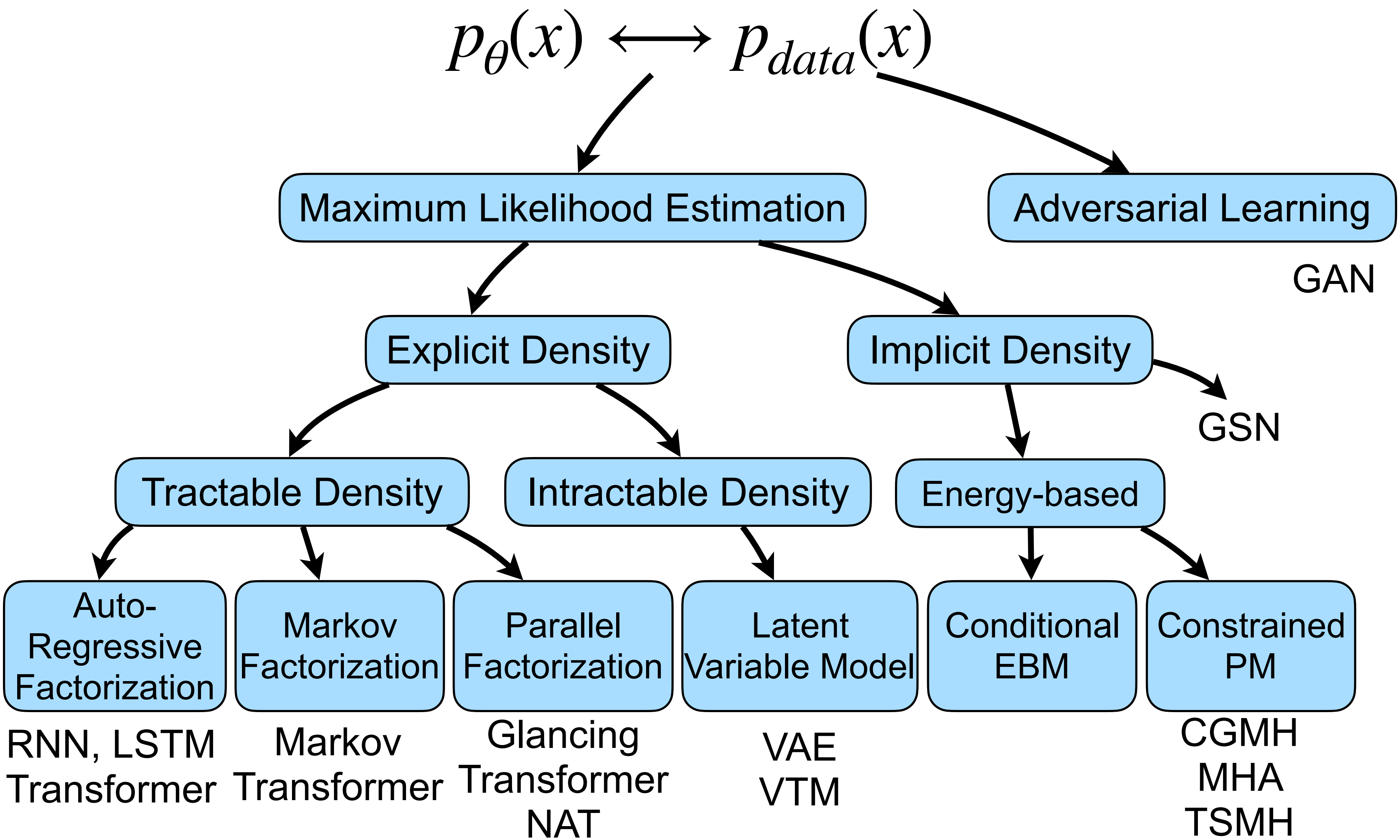
$$x = (x_1 , x_2 , x_3 , x_4, x_5 , x_6 , x_7, x_8, x_9, x_{10})$$

The central problem of *language modeling* is to find the *joint probability distribution*:

$$p_\theta(x) = p_\theta(x_1, \cdots, x_L)$$

There are many ways to represent and learn the joint probability model.

DGM Taxonomy

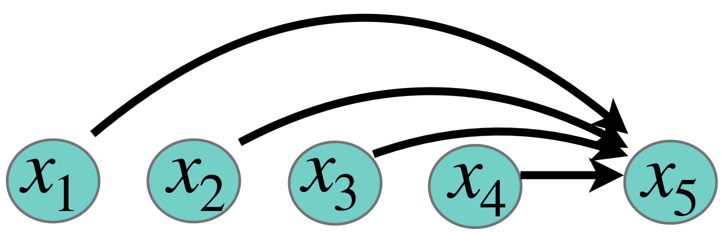


Basic Neural Generative Model

Decompose the joint distribution as a product of tractable conditional probabilities:

Given $x = [x_1, x_2, x_3 \ldots, x_n]$

$$p_\theta = \prod_{i=1}^{n} p_\theta(x_i \,|\, x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p_\theta(x_i \,|\, x_{<i})$$



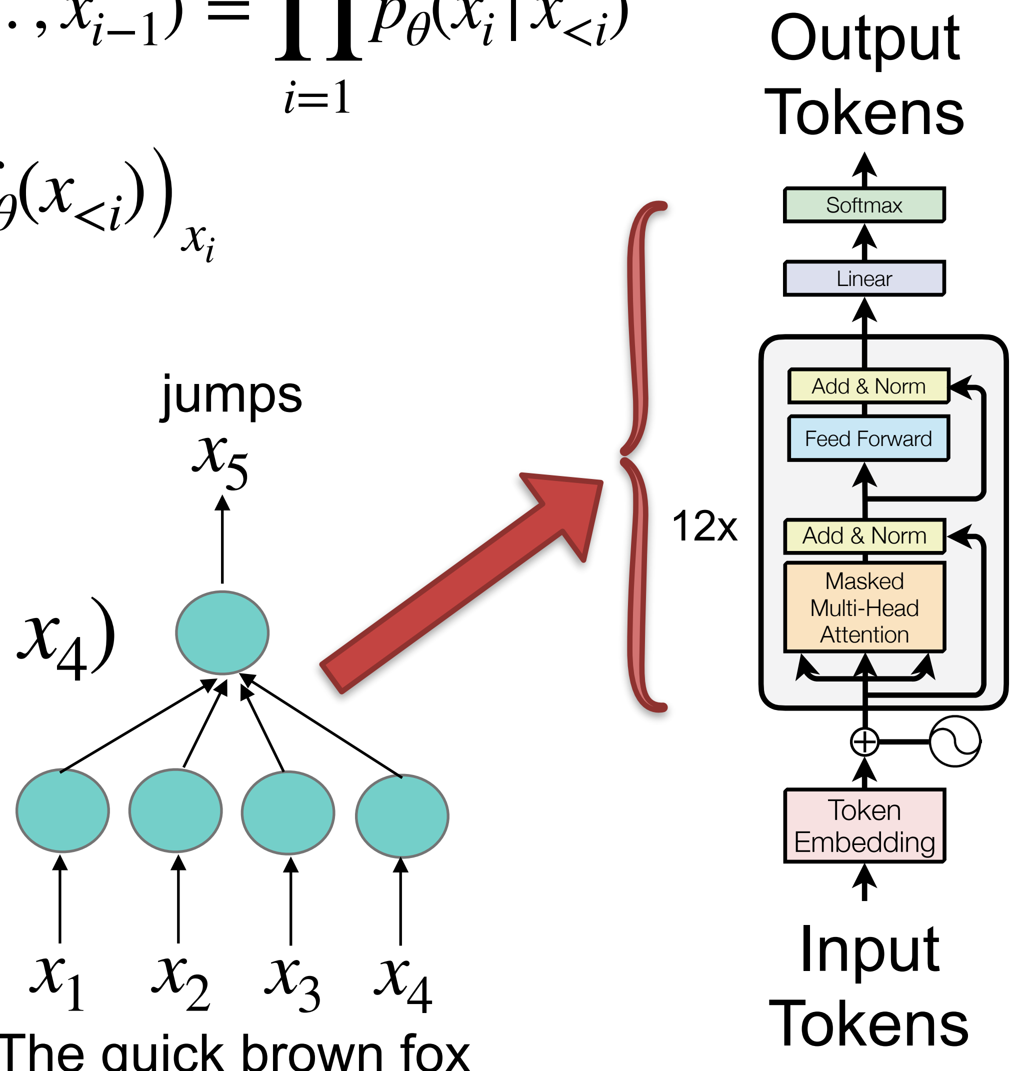
Auto-Regressive Factorization - Token Probability from a Neural Network

$$p_\theta = \prod_{i=1}^{n} p_\theta(x_i \,|\, x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p_\theta(x_i \,|\, x_{<i})$$

$$p_\theta(x_i \,|\, x_{<i}) = \mathrm{Softmax}\left(f_\theta(x_{<i})\right)_{x_i}$$

$$\mathrm{Softmax}(\mathrm{x})_j = \frac{\exp x_j}{\sum_k \exp x_k}$$

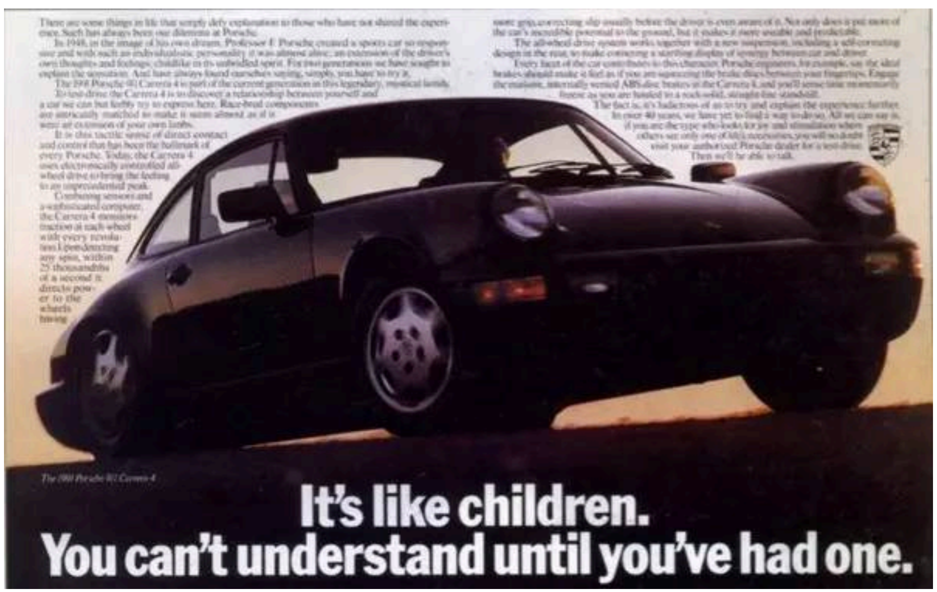
$$p_\theta(x_5 \,|\, x_1, x_2, x_3, x_4)$$



Outline

1. Overview
2. Generic Monte-Carlo Framework for Constrained NLG
3. Generating Adversarial Sentences with Semantic Category Constraint
4. Generation under Logic Constraint
5. Tailoring the Generation Density
6. Summary

Automate Creative Advertisement Design



Constrained Text Generation

To generate sentences that are:

- Fluent
- Constraint-satisfying
  - e.g. keyword-occurrence constraint

"Autumn"
"Sports shoes"

Comfortable **sports shoes**, a breathing pair of man's shoes, accompanying you in **autumn**

Why is Constrained Text Generation important?

- One generic formulation for many tasks
- Ads creative slogan design given product highlighting attributes
- Title generation for articles given keywords
- Writer assistant: automatic sentence error correction
- Machine translation with bilingual entity-dictionary

Why is Text Generation difficult?

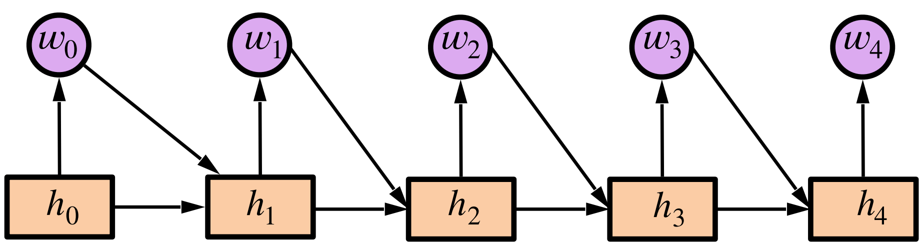
- Text space is discrete
  - Interpolation and smoothing in the surface level would not work
- High-dimensional space: exponential search space for sentence
- Controlling the generation with desired properties is challenging
- The lack of labeled data pairs <constraint, ground-truth sentence> ➔ learning without supervision!

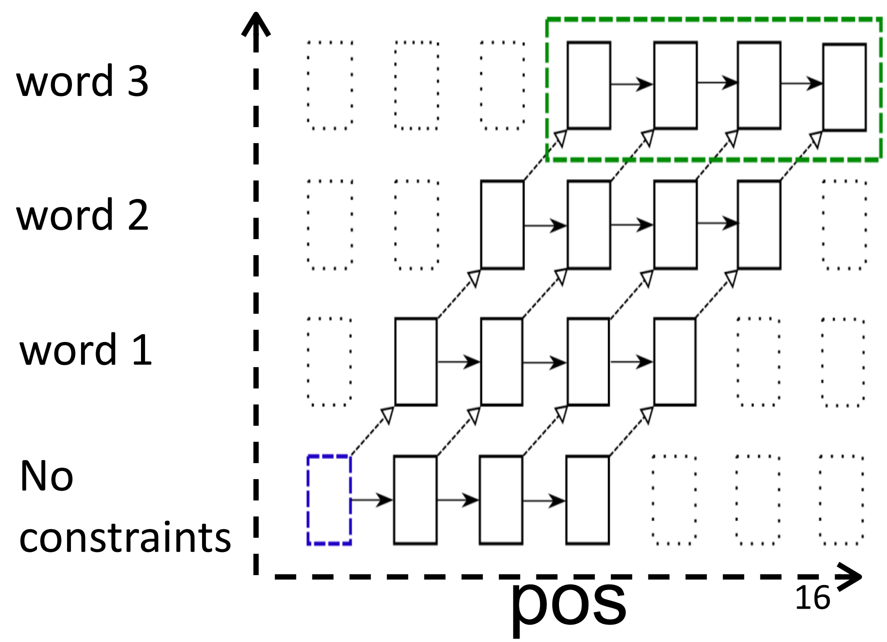
Why is Constrained Text Generation difficult?

Exponential search space, $O((N\text{-}k)^V)$

RNN grid beam search [Hokamp & Liu 2017]

does not usually produce high quality sentences





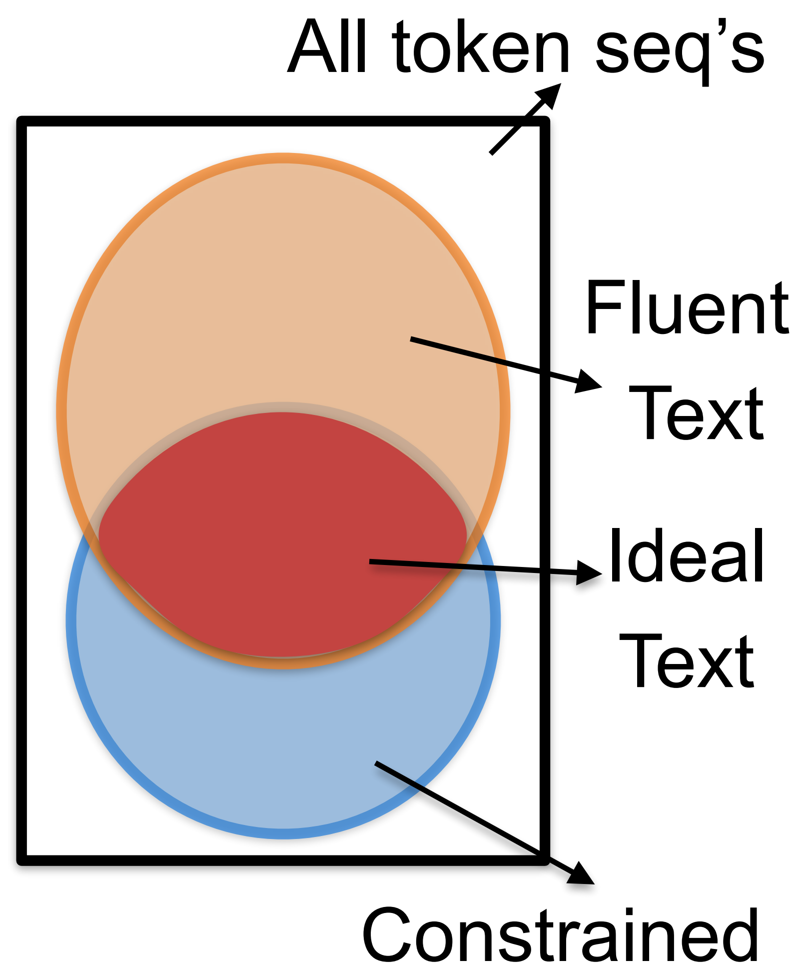
Constrained Sentence Generation via Metropolis-Hastings Sampling

- Key idea: To generation samples from the *implicit* distribution by iterative editing (MH sampling)

$$\pi(x) = \prod_{i} P(x_i \mid x_{0:i-1}) \cdot \prod_{j} P_C^j(x)$$

pre-trained language model prob.

indicator (0-1) function for constraints



CGMH [N. Miao, H. Zhou, L. Mou, R. Yan, **Lei Li**, AAAI19]

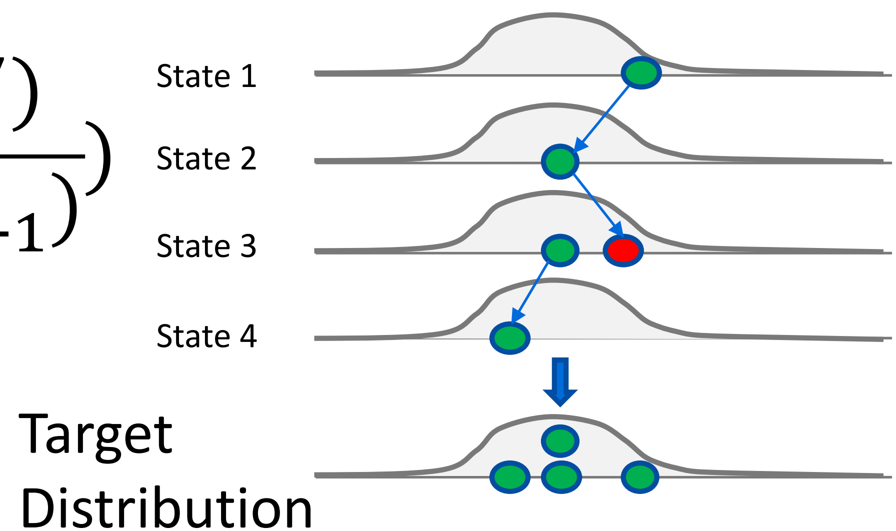
Metropolis-Hastings Sampling

One case of Markov chain Monte Carlo methods, Metropolis-Hastings(MH) performs sampling by first **proposes** a transition, and then **accepts or rejects** the transition.

$$A(x'|x_{t-1}) = \min\left(1, \frac{\pi(x') \cdot g(x_{t-1}|x')}{\pi(x_{t-1}) \cdot g(x'|x_{t-1})}\right)$$

π is the target density,

g is proposal distribution, which is easy to sample



CGMH: Main Idea

- CGMH performs constrained generation by:
  1. Pretrain Neural Language Model (e.g. GPT2);
  2. Iterative Editing:
     1) Start from a initial sentence x_0;
     2) Propose a new sentence x_t from x_{t-1}, and accept/ reject the action. Action proposal include:
        I. Replacement: change a word to another one
        II. Insertion: add a word
        III. Deletion: remove a word

BMW, the sports car of daily life

BMW, the sports car of today's life
BMW, the sports car of future life
BMW, the sports car of new life
BMW, the sports car of happy life
…

CGMH [N. Miao, H. Zhou, L. Mou, R. Yan, **Lei Li**, AAAI19]

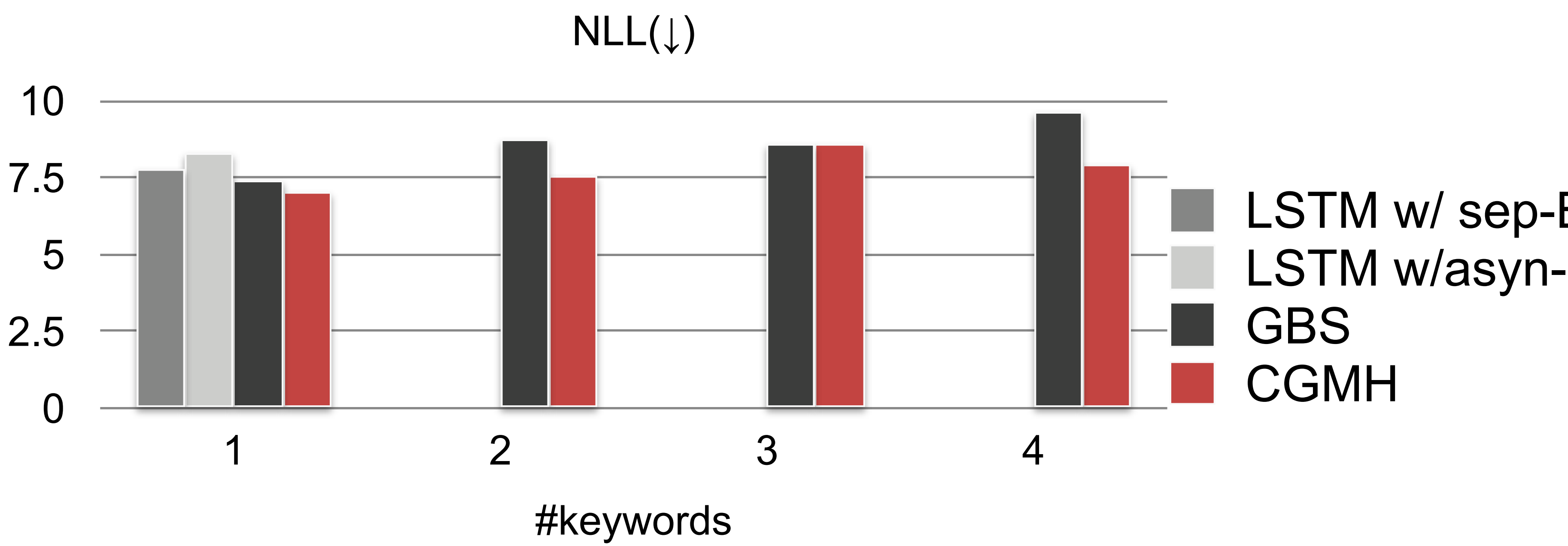
CGMH Iteratively Edits Candidates

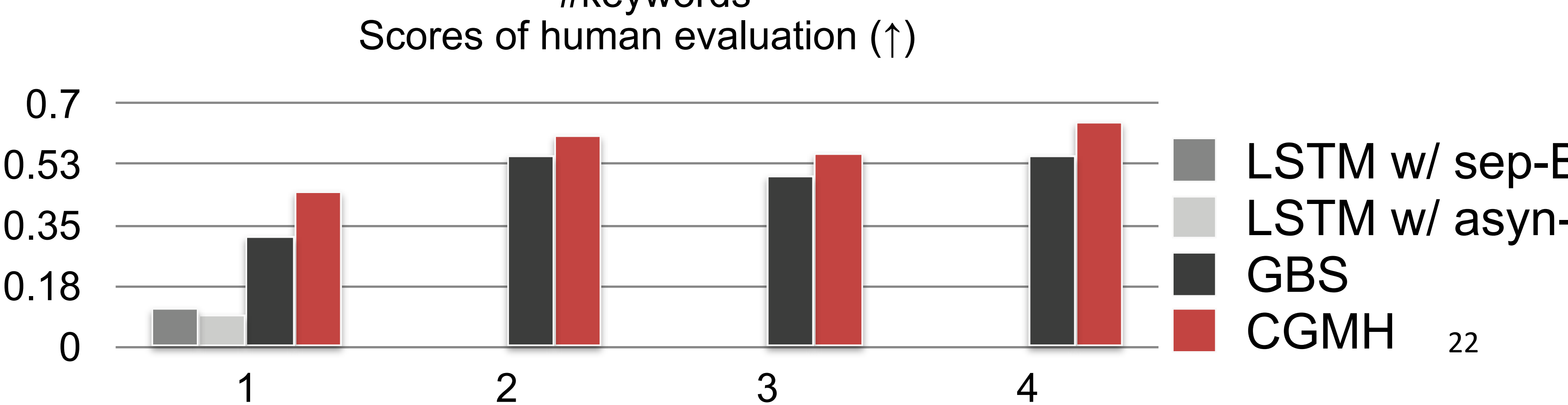
Step	Action	Acc/Rej	Sentences
0	[Input]		BMW sports
1	Insert	Accept	BMW sports car
2	Insert	Accept	BMW the sports car
…	…	…	…
6	Insert	Accept	BMW , the sports car of daily life
7	Replace	Accept	BMW , the sports car of ~~daily~~future life
8	Insert	Accept	BMW , the sports car of the future life
9	Delete	Reject	BMW , the sports ~~car~~ of the future life
10	Delete	Accept	BMW , the sports car of the future ~~life~~
11	[Output]		BMW , the sports car of the future

Evaluation 1: Keyword to Sentence

- Keywords to sentence generation (hard constraints)
  - Aim: To generate fluent sentences containing the given set of words.
  - Dataset: A subset of one-billion-word corpus (5M)
  - Input: Keywords random selected from the target sentence.
  - Constraint: $1_{\text{keywords occur in sentence}}$

CGMH generates better sentences from keywords





Keyword-to-Sentence: Showcase

Keyword(s)	CGMH	GBS
friends	My good friends were in danger .	But friends and family have been arrested .
project	The first project of the scheme .	The project , which is expected to be completed next year
have, trip	But many people have never made the trip .	But the trip has be completed .
lottery, scholarships	But the lottery has provided scholarships.	The lottery is a scholarship .
decision, build, home	The decision is to build a new home.	The decision builds a house for home .
attempt, copy, painting, denounced	The first attempt to copy the painting was denounced.	But attempt to copy painting will be denounced.

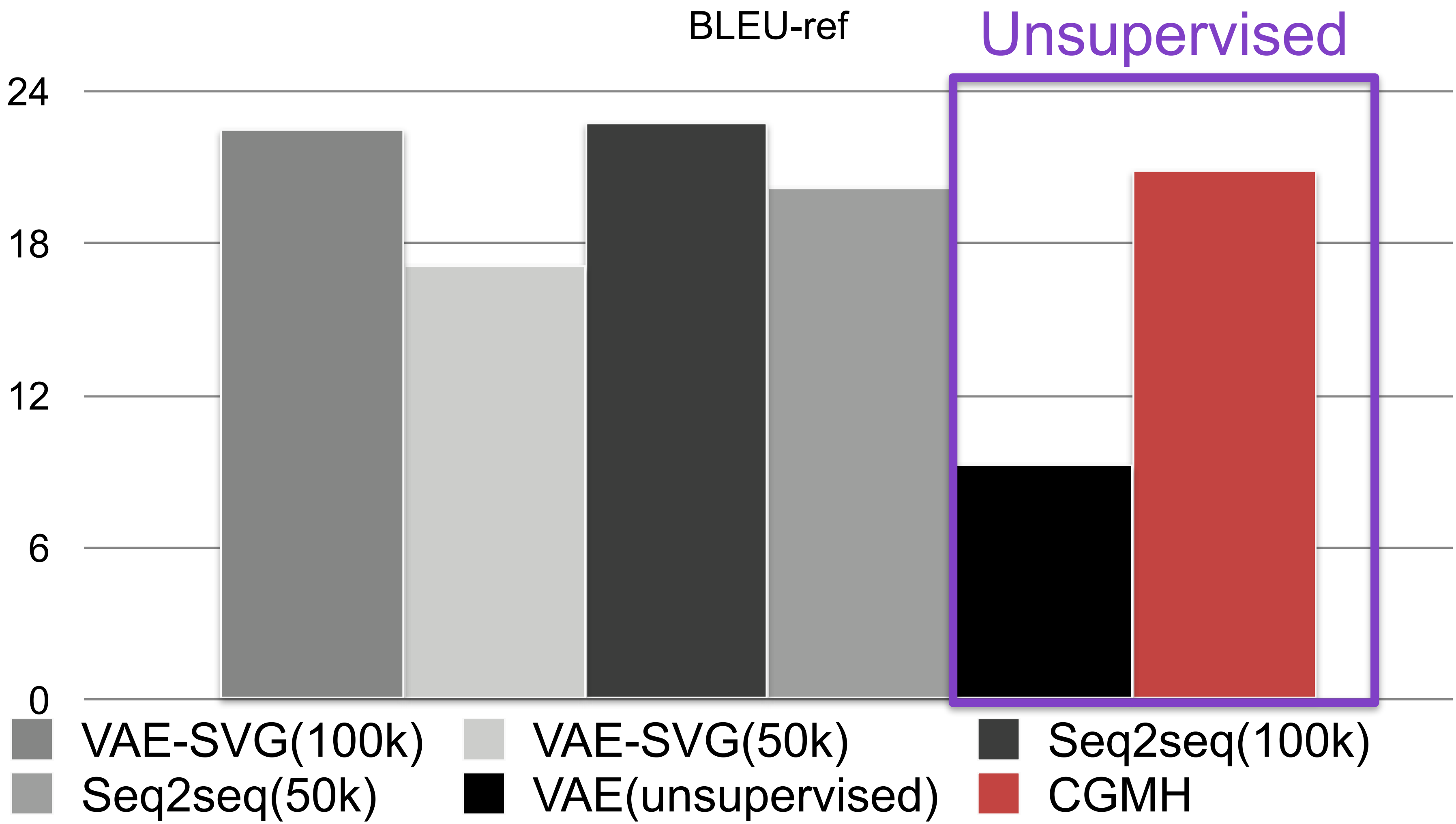
Evaluation 2: Paraphrase Generation

- Unsupervised paraphrase generation (soft constraints)
  - Aim: To generate sentences with similar meaning of the given one.

what's the best plan to lose weight

➜ what's the best way to slim down quickly

CGMH is the first unsupervised model to achieve comparable results with supervised models.



Impact

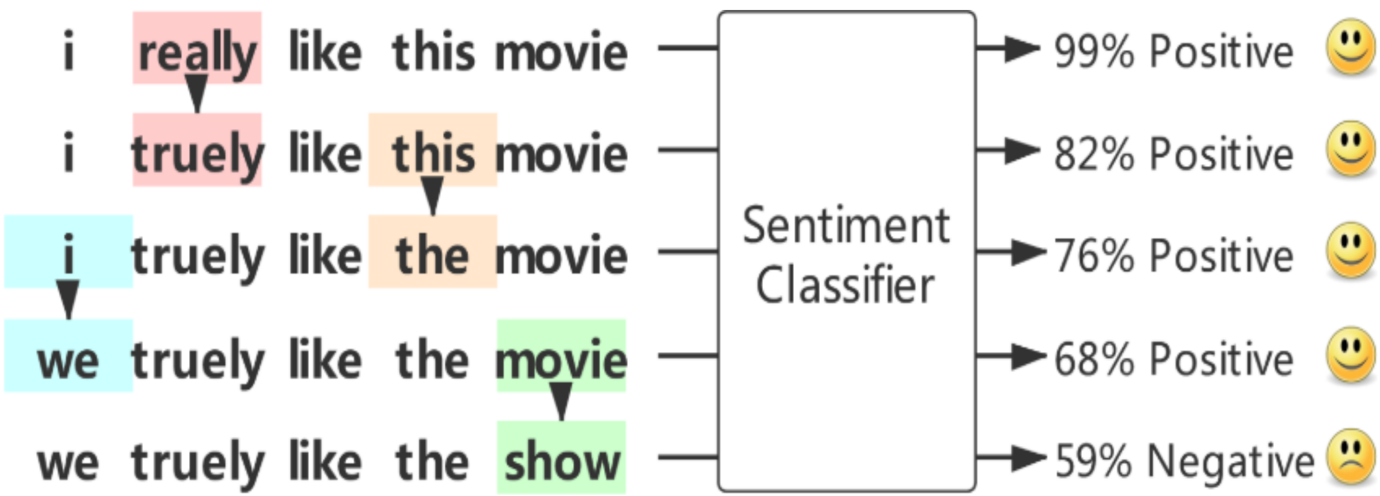
- CGMH is deployed in a large-scale online ads creation platform
- Active used by 100,000 merchants and organizations
- Adoption rate: ~75%

“Autumn”
“Sports shoes”

Comfortable **sports shoes**, a breathing pair of man's shoes, accompanying you in **autumn**

Generating Adversarial Fluent Sentence Generation

- Machine learning models are vulnerable to noises and attacks.
- Generating fluent adversarial text is challenging, due to the discreteness in text! (Ebrahimi et al., 2018; Alzantot et al., 2018)
- Our MHA achieves higher attack success rate

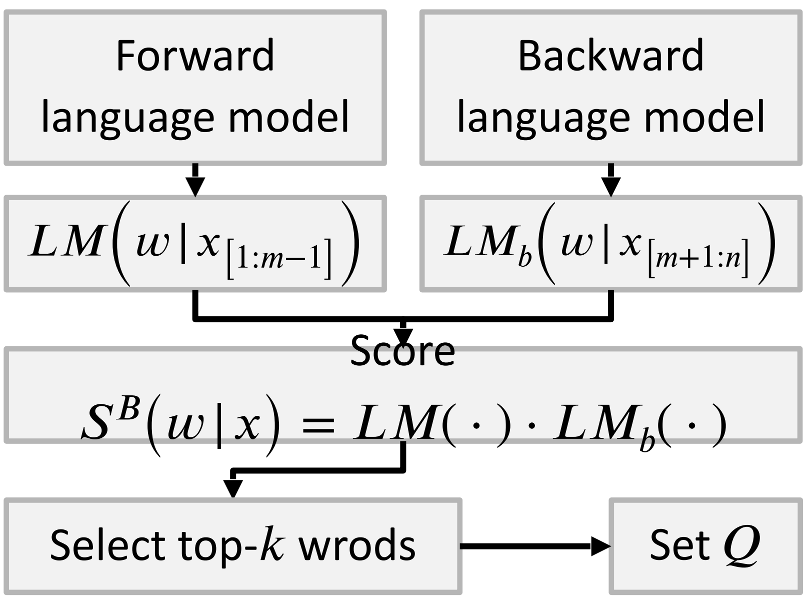


MHA [H. Zhang, N. Miao, H. Zhou, **Lei Li**, ACL19a]

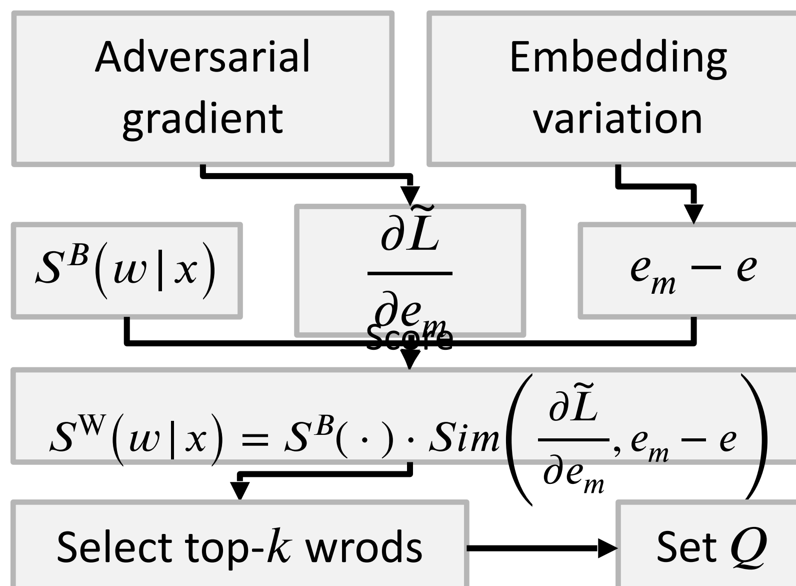
Adversarial Sentence Generation via MCMC

Reuse the CGMH algorithm

- *Blackbox b*-MHA
  - Black-box setting
  - Pre-select set Q with a forward language model and a backward language model



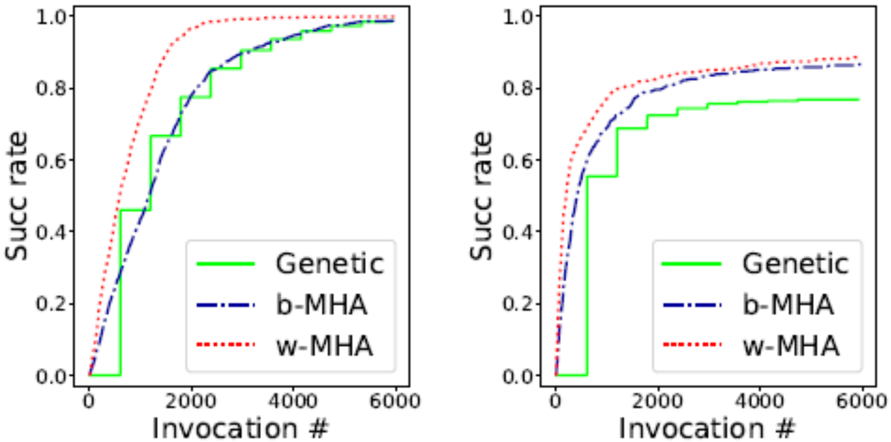
- *Whitebox w*-MHA
  - White-box setting
  - Pre-select set Q with a forward language model, a backward language model and the similarity of embedding variation and adversarial gradients.



Higher Attack Success Rate and Improved Text Classifier!

- MHA achieves higher attack success rate with fewer invocations, and gives lower perplexity, than the genetic approach (Alzantot et al., 2018) baseline.
- Examples generated by MHA may improve the adversarial robustness and the classification accuracy after adversarial training.

Attack Success Rate



(a) IMDB (b) SNLI

Accuracy w/ Adversaries

Model	Acc (%) Train # = 10K	30K	100K
Victim model	58.9	65.8	73.0
+ Genetic adv training	58.8	66.1	**73.6**
+ *w*-MHA adv training	**60.0**	**66.9**	**73.5**

Zhang et al., Generating Fluent Adversarial Examples for Natural Languages, in ACL, 2019, short paper.

Generation under Combinatorial Constraints

- Logical and Combinatorial constraints
- E.g. generating a question for the following statement.
  - Paris is located in France.
  - ==> Is Paris located in France?
  - ==> Which country is Paris located in?

TSMH [M. Zhang, N. Jiang, **Lei Li**, Yexiang Xue, EMNLP20e]

Generation under Combinatorial Constraints

- Logical and Combinatorial constraints

$$\pi(x) = \underbrace{P_{\text{LM}}(x;\theta)}_{\text{Language Model}} \cdot \underbrace{\phi(x)}_{\text{Constraint}}$$

$$\phi(x) = \beta^{M - \sum_i c_i(x)}, \quad 0 < \beta < 1$$

$c_i(x)$ is a formula or logical constraint. e.g. the first word must be Wh- words.

Method: Tree search enhanced Metropolis-Hastings

details in

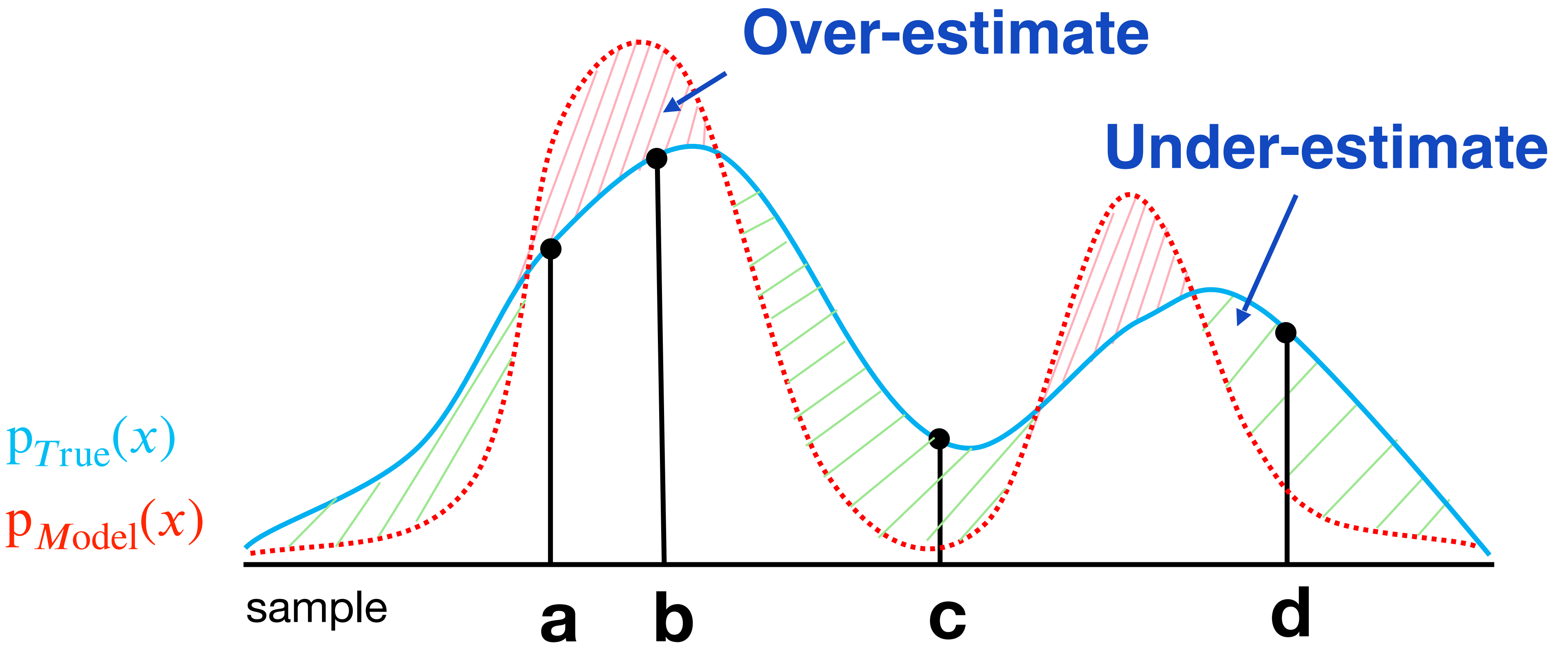
TSMH [M. Zhang, N. Jiang, **Lei Li**, Yexiang Xue, EMNLP20e]

Use the Right Scissor: Monte-Carlo Tailoring

- Pre-trained language model needs to be fine-tuned on specific tasks
- e.g. use the generic GPT-2/GPT-3 to generate news articles
  - How to ensure domain-specific style?

Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

Problem: Over- and Under-estimated Density



Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

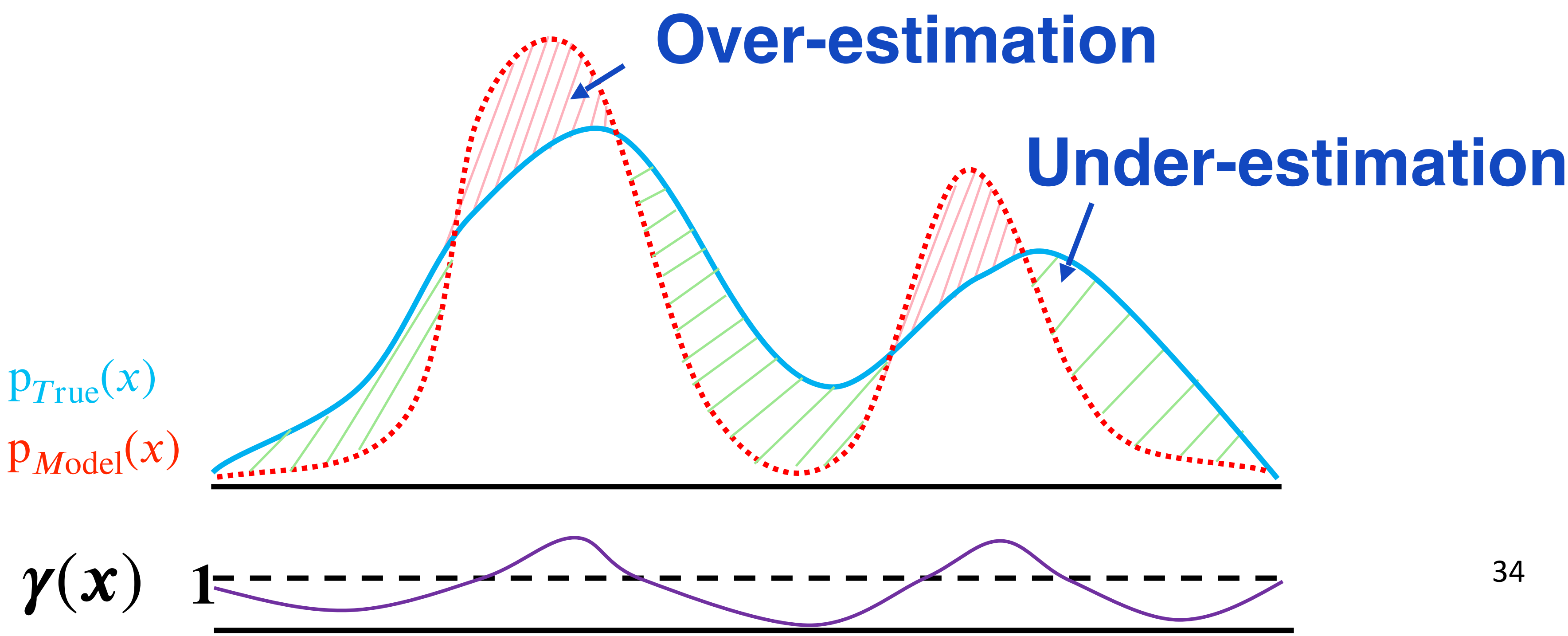
Approach: Ratio Estimator

We first define ratio estimator as

$$\gamma(x) = \frac{P_{Model}(x)}{P_{Real}(x)} \qquad P_{Tailor}(x) \propto \frac{P_{Model}(x)}{max(1,\ \gamma(x))}$$

When $\gamma(x) > 1$, the model over-estimates real probabilities;

When $\gamma(x) < 1$, the model under-estimates real probabilities;



Challenge: How to estimate ratio

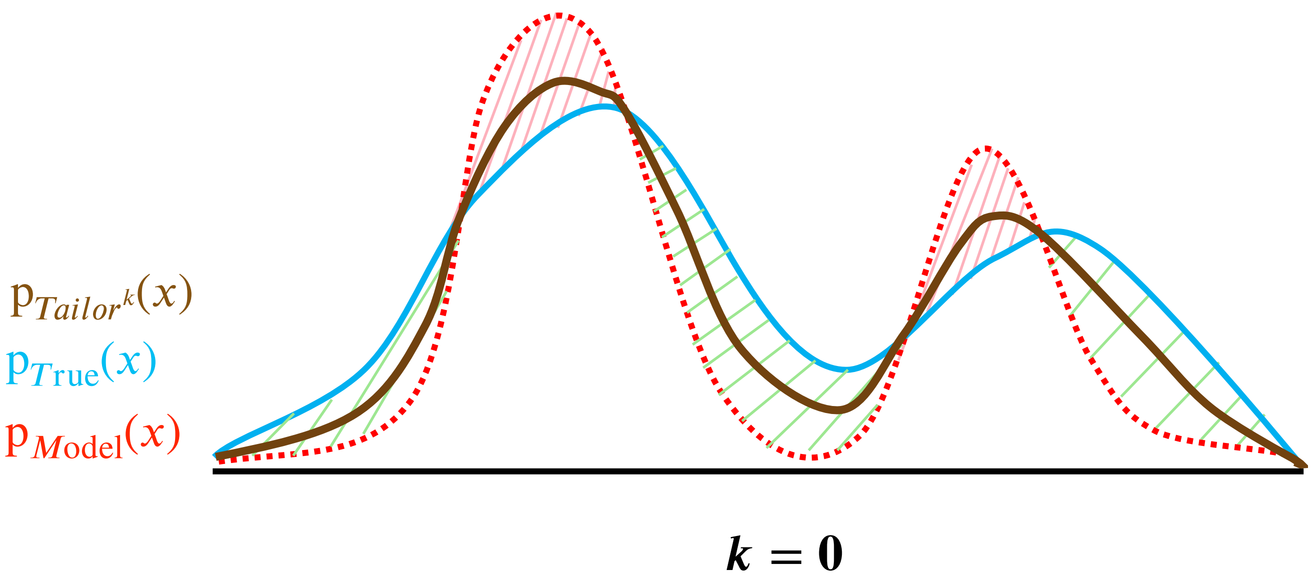
A single ratio estimator may not be powerful enough to accurately estimate $\boldsymbol{\gamma}(\boldsymbol{x}) = \dfrac{\boldsymbol{P}_{\boldsymbol{Model}}(\boldsymbol{x})}{\boldsymbol{P}_{\boldsymbol{Real}}(\boldsymbol{x})}$

Approach – Hierarchical γ and Tailor

A single ratio estimator may not be powerful enough to accurately estimate $\frac{P_{Model}(x)}{P_{Real}(x)}$

We boost several ratio estimators by:

1. Estimate $\gamma_0(x) = \frac{P_{Model}(x)}{P_{Real}(x)}$, and get $P^0_{Tailor} \propto \frac{P_{Model}(x)}{min(1, \gamma_0(x))}$

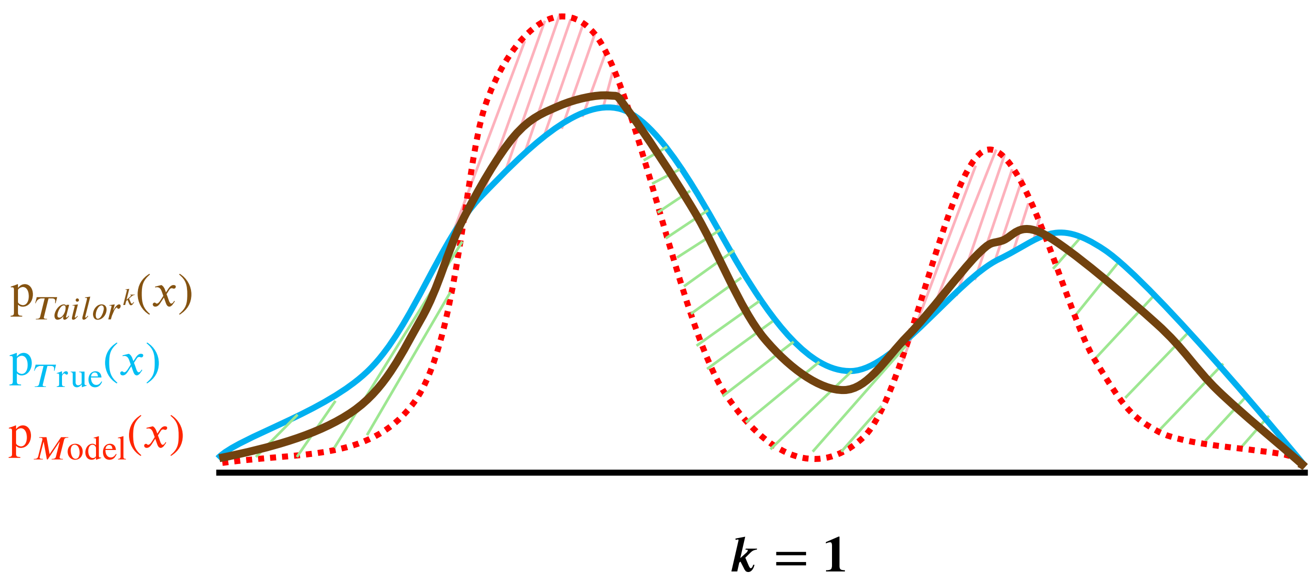


Approach – Hierarchical γ and Tailor

A single ratio estimator may not be powerful enough to accurately estimate $\frac{P_{Model}(x)}{P_{Real}(x)}$

We boost several ratio estimators by:

1. Estimate $\gamma_0(x) = \frac{P_{Model}(x)}{P_{Real}(x)}$, and get $P^0_{Tailor} \propto \frac{P_{Model}(x)}{min(1, \gamma_0(x))}$
2. Estimate $\gamma_1(x) = \frac{P^0_{Tailor}(x)}{P_{Real}(x)}$, and get $P^1_{Tailor} \propto \frac{P^0_{Tailor}(x)}{min(1, \gamma_1(x))}$
3. …
4. Output P^k_{Tailor}



How to estimate efficiently?

The most direct idea is **reject sampling**.

But Rejection Sampling is inefficient!

1. Generate a sentence from P_{Model}
2. Reject the sample with probability

$$1 - \frac{1}{max(1, \gamma(x))} \text{ or}$$

$$1 - \frac{P_{Model}(x)}{\prod_{i=1}^{k} max(1,, \gamma_i(x))}$$

How to estimate efficiently?

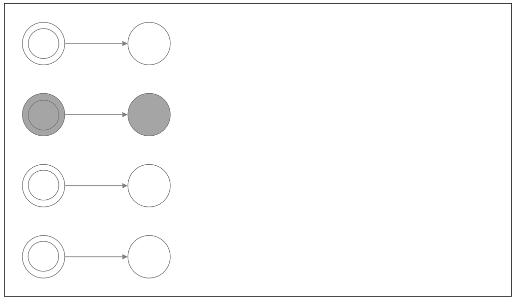
The most direct idea is **reject sampling**.

But Rejection Sampling is inefficient!

1. Generate a sentence from P_{Model}
2. Reject the sample with probability

$$1 - \frac{1}{max(1, \gamma(x))} \text{ or}$$

$$1 - \frac{P_{Model}(x)}{\prod_{i=1}^{k} max(1,, \gamma_i(x))}$$



How to estimate efficiently?

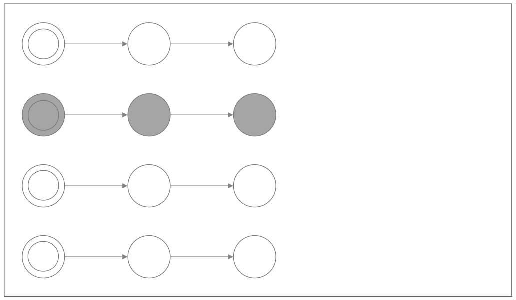
The most direct idea is **reject sampling**.

But Rejection Sampling is inefficient!

1. Generate a sentence from P_{Model}
2. Reject the sample with probability

$$1 - \frac{1}{max(1, \gamma(x))} \text{ or}$$

$$1 - \frac{P_{Model}(x)}{\prod_{i=1}^{k} max(1,, \gamma_i(x))}$$



How to estimate efficiently?

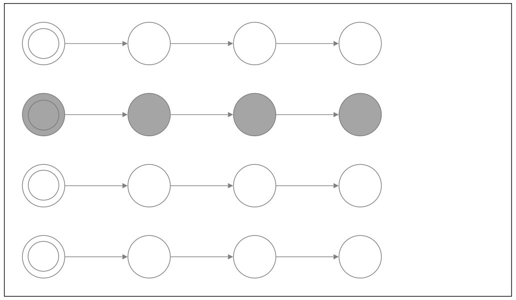
The most direct idea is **reject sampling**.

But Rejection Sampling is inefficient!

1. Generate a sentence from P_{Model}
2. Reject the sample with probability

$$1 - \frac{1}{max(1, \gamma(x))} \text{ or}$$

$$1 - \frac{P_{Model}(x)}{\prod_{i=1}^{k} max(1,, \gamma_i(x))}$$



How to estimate efficiently?

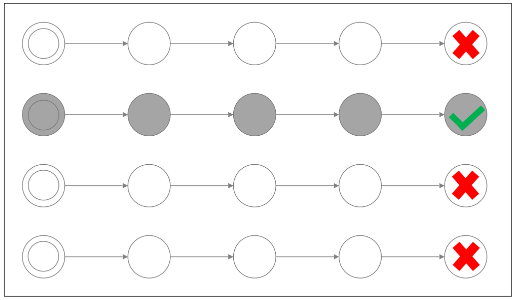
The most direct idea is **reject sampling (RS)**.

But Rejection Sampling is inefficient!

1. Generate a sentence from P_{Model}
2. Reject the sample with probability

$$1 - \frac{1}{max(1, \gamma(x))} \text{ or}$$

$$1 - \frac{P_{Model}(x)}{\prod_{i=1}^{k} max(1,, \gamma_i(x))}$$



Since most samples are finally rejected, RS is highly inefficient.

Observation from an Example

Luckily, an interesting property may help us!

For example, assume we are finetuning GPT-2 on a news domain.

When sampling from $P_{Model}(x)$, we get a sentence

'My mom cooked …'

Observation from an Example

Luckily, an interesting property may help us!

For example, assume we are finetuning GPT-2 on a news domain.

When sampling from $P_{Model}(x)$, we get a sentence

'My mom cooked …'

We can safely reject this sentence without generating the whole sentence, because it doesn't look like news at all.

Sequential Monte Carlo Sampling

So we need to have a ratio estimator for unfinished sentences,

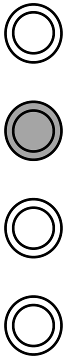
$$\boldsymbol{\gamma}'\left(\hat{\boldsymbol{x}}_{[\mathbf{1}:\boldsymbol{i}]}\right) = \boldsymbol{min}_{\boldsymbol{x}_{[\mathbf{1}:\boldsymbol{i}]}=\hat{\boldsymbol{x}}_{[\mathbf{1}:\boldsymbol{i}]}}(\boldsymbol{\gamma}(\boldsymbol{x}))$$

$\gamma'\left(\hat{x}_{[1:i]}\right)$ is the minimum $\gamma(x)$ with the same prefix $\hat{x}_{[1:i]}$.

If $\gamma'\left(\hat{x}_{[1:i]}\right)$ is large, we can safely reject the sample at step i, because all sentences with this prefix are heavily over-estimated.

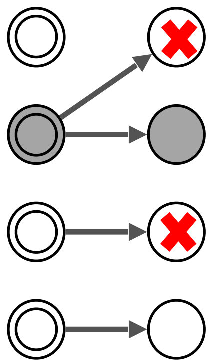
Sequential Monte Carlo Sampling

With $\boldsymbol{\gamma}'\left(\hat{\boldsymbol{x}}_{[\mathbf{1}:\boldsymbol{i}]}\right)$, SMC (Sequential Monte Carlo) can be easily performed.



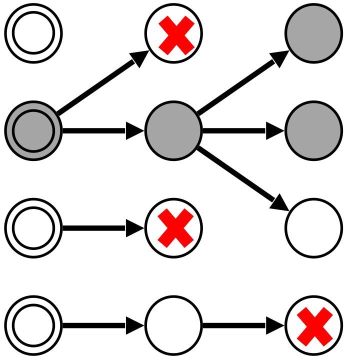
Sequential Monte Carlo Sampling

With $\gamma'\left(\hat{x}_{[1:i]}\right)$, SMC (Sequential Monte Carlo) can be easily performed.



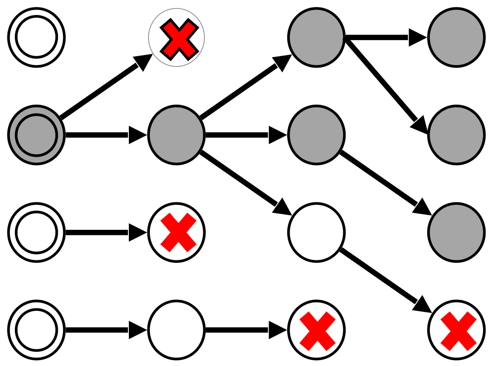
Sequential Monte Carlo Sampling

With $\gamma'\left(\hat{\boldsymbol{x}}_{[1:i]}\right)$, SMC (Sequential Monte Carlo) can be easily performed.



Sequential Monte Carlo Sampling

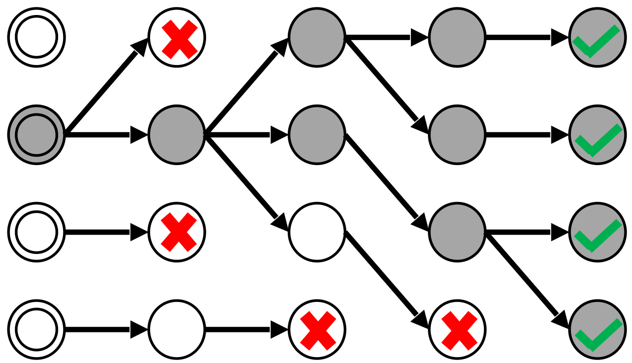
With $\gamma'\left(\hat{x}_{[1:i]}\right)$, SMC (Sequential Monte Carlo) can be easily performed.



But, SMC has a problem…

However, SMC leads to severe *degeneracy* problem.

Generated samples in a batch are only slightly different.

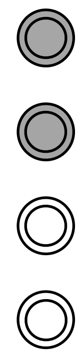


This year , the total amount invested was 3,800 billion US dollars .
This year , the total amount invested was 2,500 billion US dollars .
This year , the total amount invested was 2,500 billion pounds .
This year , the total amount invested was 2,500 million dollars .

MC-Tailor with ERS

To solve the degeneracy problem of SMC, we propose **ERS(Early Rejection Sampling).**

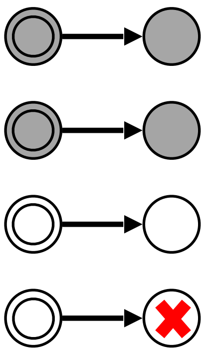
Instead of preform resampling, ERS **directly kills unpromising samples** and release computation resource to parallel threads.



Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

MC-Tailor with ERS

To solve the degeneracy problem of SMC, we propose **ERS(Early Rejection Sampling).**

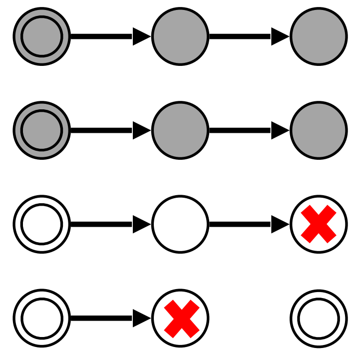


Instead of preform resampling, ERS **directly kills unpromising samples** and release computation resource to parallel threads.

Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

MC-Tailor with ERS

To solve the degeneracy problem of SMC, we propose **ERS(Early Rejection Sampling).**

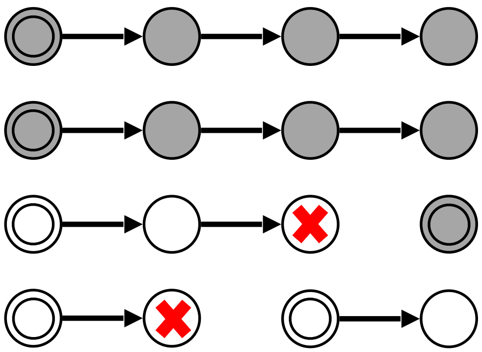


Instead of preform resampling, ERS **directly kills unpromising samples** and release computation resource to parallel threads.

Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

MC-Tailor with ERS

To solve the degeneracy problem of SMC, we propose **ERS(Early Rejection Sampling).**



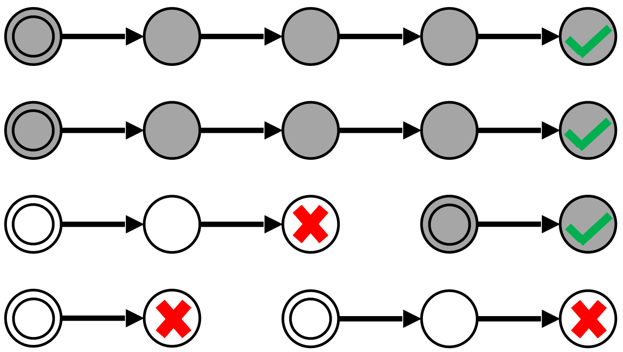
Instead of preform resampling, ERS **directly kills unpromising samples** and release computation resource to parallel threads.

Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

MC-Tailor with ERS

To solve the degeneracy problem of SMC, we propose **ERS(Early Rejection Sampling).**

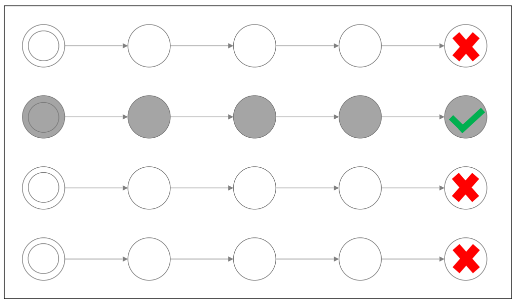
Instead of preform resampling, ERS **directly kills unpromising samples** and release computation resource to parallel threads.



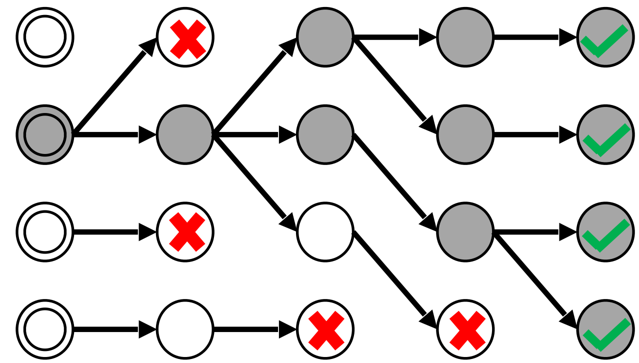
Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

Comparing Sampling Methods

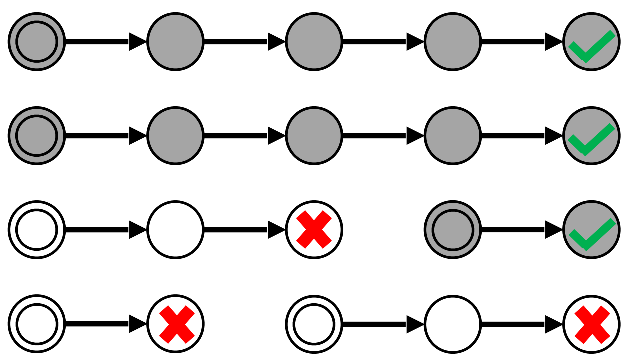
Rejection-Sampling



SMC



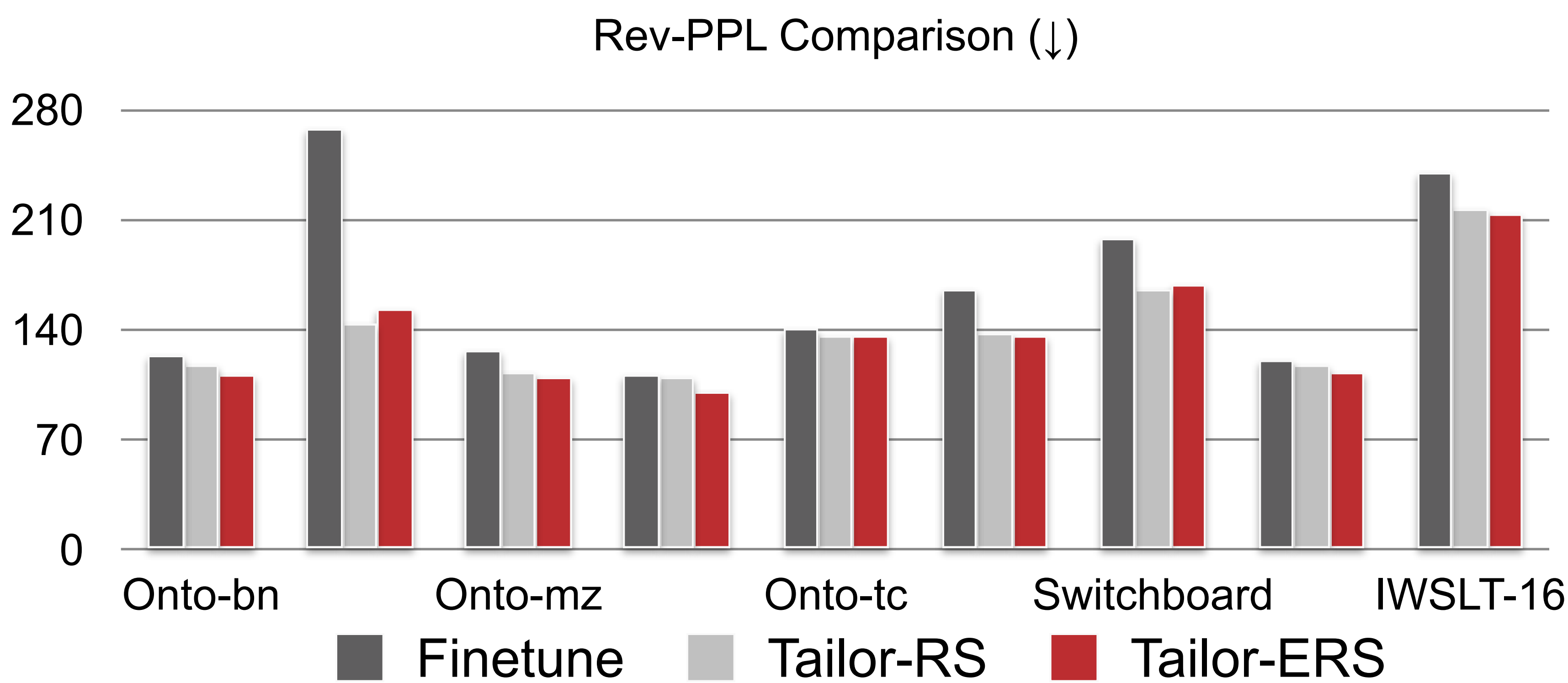
MCTailor - ERS



Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

Experiment – Results

Tailor performs better than baseline on all metrics including generation quality.



Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

Cases Generated by MC-Tailor

MC-Tailor reallocates probabilities of simple utterances or disfluent sentences to complex and natural ones.

	Direct-Finetune	MCTailor-ERS
1	In the case if you think of this -	And do you still feel that way every day ?
2	Oh well .	But it would be tough .
3	I ’ve been there n’t said anything wrong .	He knew about the attack at the Paris offices .

Do you have the right scissor? MC Tailor [N. Miao, Y. Song, H. Zhou, **Lei Li**, ACL20a]

Recap

1. Natural Language Generation Problem
2. Generic Monte-Carlo Framework for Constrained NLG
3. Generating Adversarial Sentences with Semantic Category Constraint
4. Tailoring the Generation Density

Thanks

- Joint w/ Ning Miao, Hao Zhou, Huangzhao Zhang, Yuxuan Song, Lili Mou, Rui Yan, Maosen Zhang, Yexiang Xue, Nan Jiang
- Contact: lileilab@bytedance.com

Reference

1. Ning Miao, Hao Zhou, Lili Mou, Rui Yan, Lei Li. “CGMH: Constrained Sentence Generation by Metropolis-Hastings Sampling”. In: the 33rd AAAI Conference on Artificial Intelligence (AAAI). Jan. 2019.
2. Huangzhao Zhang, Ning Miao, Hao Zhou, Lei Li. “Generating Fluent Adversarial Examples for Natural Languages”. In: the 57th Annual Meeting of the Association for Computational Linguistics (ACL) - short papers. July 2019.
3. Ning Miao, Hao Zhou, Chengqi Zhao, Wenxian Shi, Lei Li. “Kernelized Bayesian Softmax for Text Generation”. In: the 33rd Conference on Neural Information Processing Systems (NeurIPS). Dec. 2019.
4. Ning Miao, Yuxuan Song, Hao Zhou, Lei Li. “Do you have the right scissors? Tailoring Pre-trained Language Models via Monte-Carlo Methods”. In: the 58th Annual Meeting of the Association for Computational Linguistics (ACL) - short papers. July 2020.
5. Maosen Zhang, Nan Jiang, Lei Li, Yexiang Xue. “Constraint Satisfaction Driven Natural Language Generation: A Tree Search Embedded MCMC Approach”. In: the Conference on Empirical Methods in Natural Language Processing (EMNLP) - Findings. Nov. 2020.