CGMH: Constrained Sentence Generation by Metropolis-Hastings Sampling

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, Lei Li

ByteDance 字节跳动

Content

- Motivation
  - Constrained generation is useful
  - Constrained generation is difficult for current methods
- Introduction
  - Metropolis-Hastings sampling
- Method
  - Stationary distribution
  - Proposal
  - Accept/Reject
- Experiment
  - Keywords2Sentence generation
  - Unsupervised paraphrase generation
  - Sentence correction

Motivation

- We often need to add constrains to sentence generation.
  - Hard constrains (eg. keyword2sentence)

    Juice -> Brand natural juice, specially made for you

Motivation

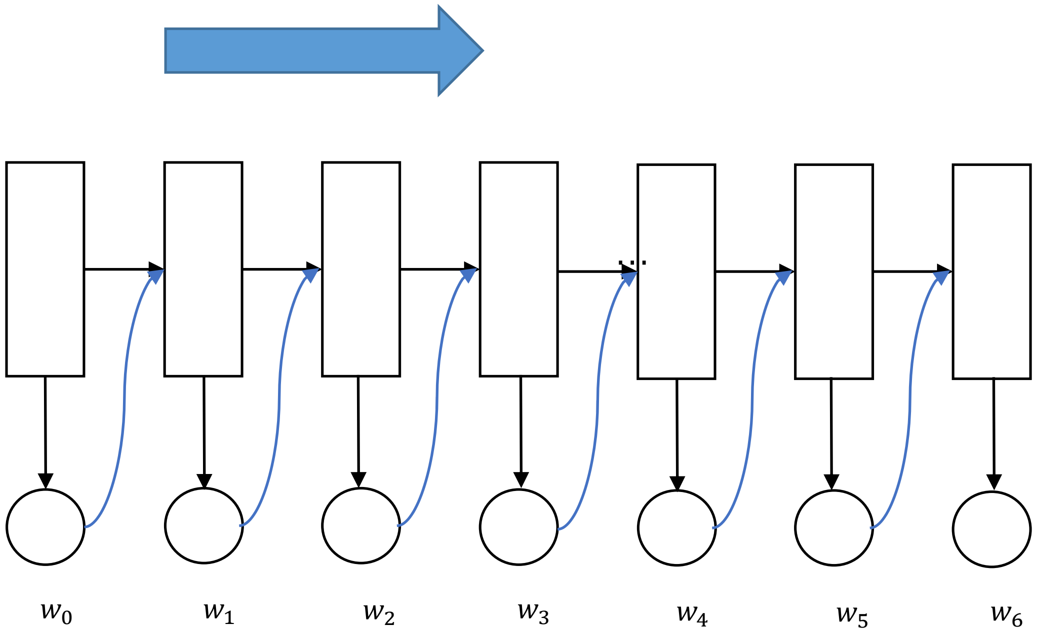
- We often need to add constrains to sentence generation.
  - Hard constrains (eg. keyword2sentence)

    Juice -> Brand natural juice, specially made for you
  - Soft constrains (eg. paraphrase)

    The movie is a great success -> It is one of my favorite movies

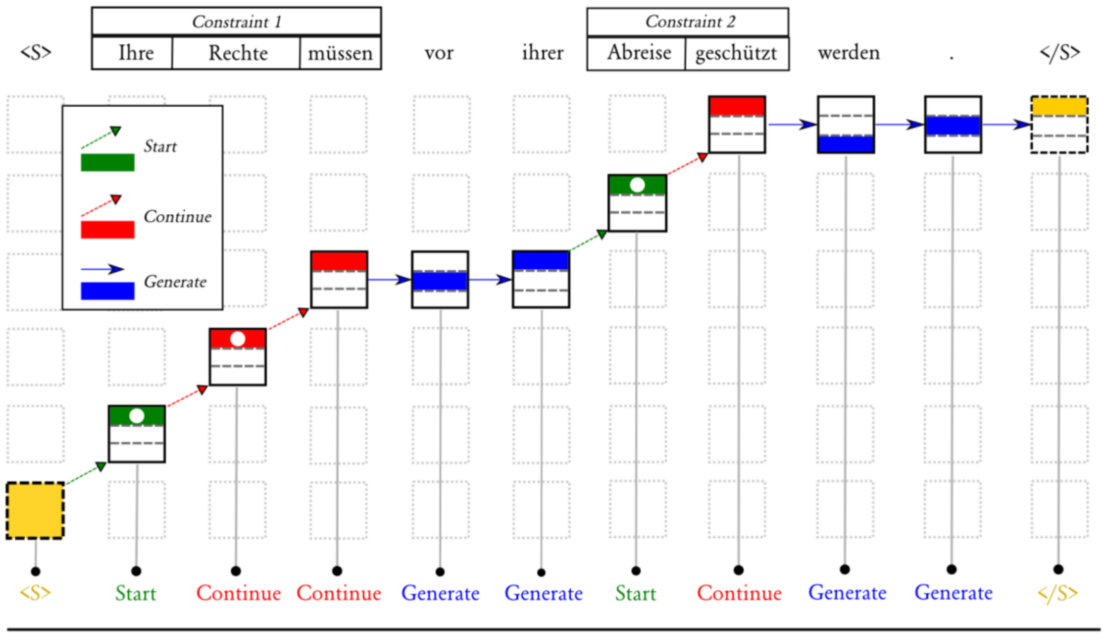
Motivation

- It's difficult to add constraints to sequential language models.



Motivation

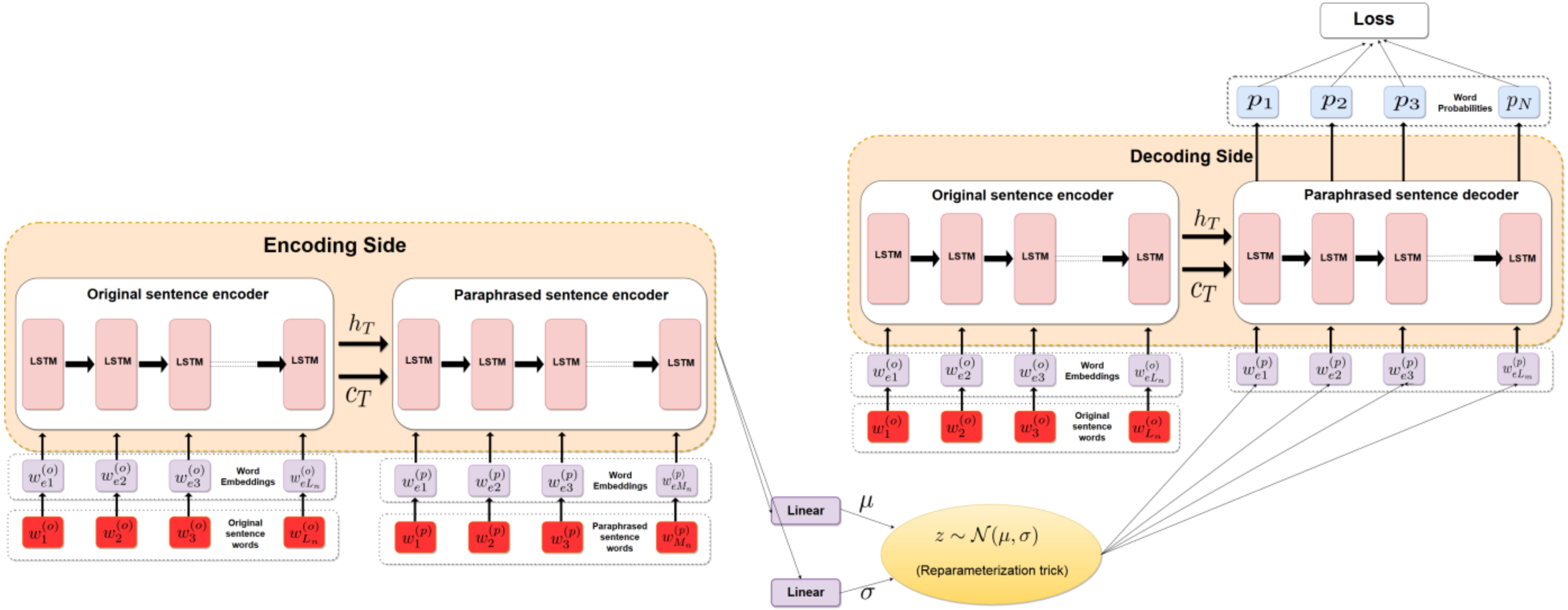
- Methods dedicated for constrained sentence generation can only handle a specific kind of constraints.
  - Grid Beam Search(GBS)[1]
  - Constrained Beam Search(CBS)[2]



Input: Rights protection should begin before their departure .

Motivation

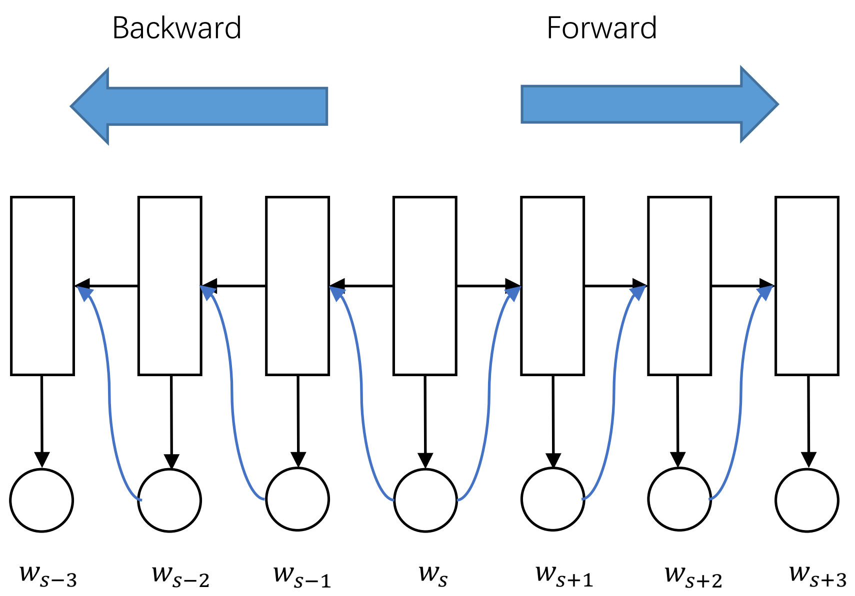
➢ Methods dedicated for constrained sentence generation can only handle a specific kind of constraints.
  - VAE-SVG[3]



Motivation

➢ Current models don’t perform well

- LSTM w/ sep-B/F[4] generates **independent backward and forward** sequences from the given word.



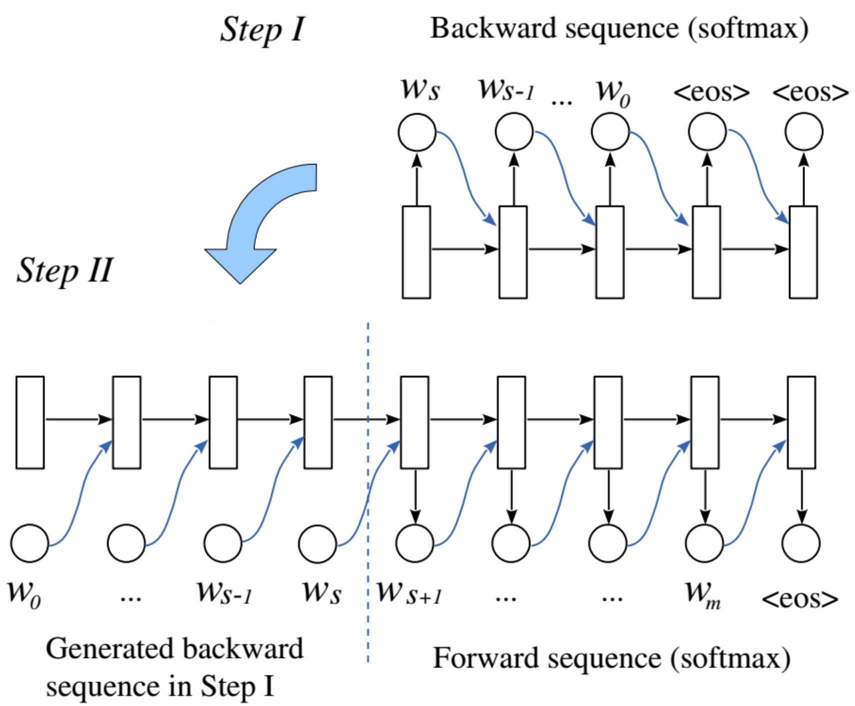
Eg: demand -> this is what it does in <u>demand</u> is very necessary .

Motivation

➢ Current models don’t perform well

- LSTM w/ asyn-B/F[4] **first generates the first half of a sentence and then generates another half** conditioned on the first half.

Eg: player -> The best name of the <u>player</u> is not making a year .

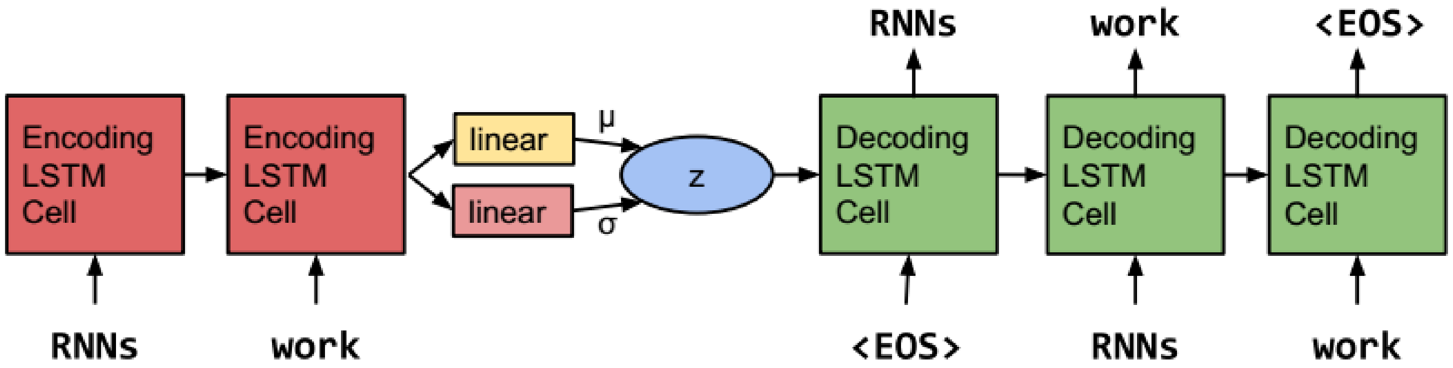


Motivation

➢ Current models don't perform well

- VAE[5] - We can perform paraphrasing by
  1. Encode a sentence into a distributional representation
  2. Add some noise to the representation
  3. then decoding the disturbed representation.

Unfortunately, the generated sentences of this method is of low quality.

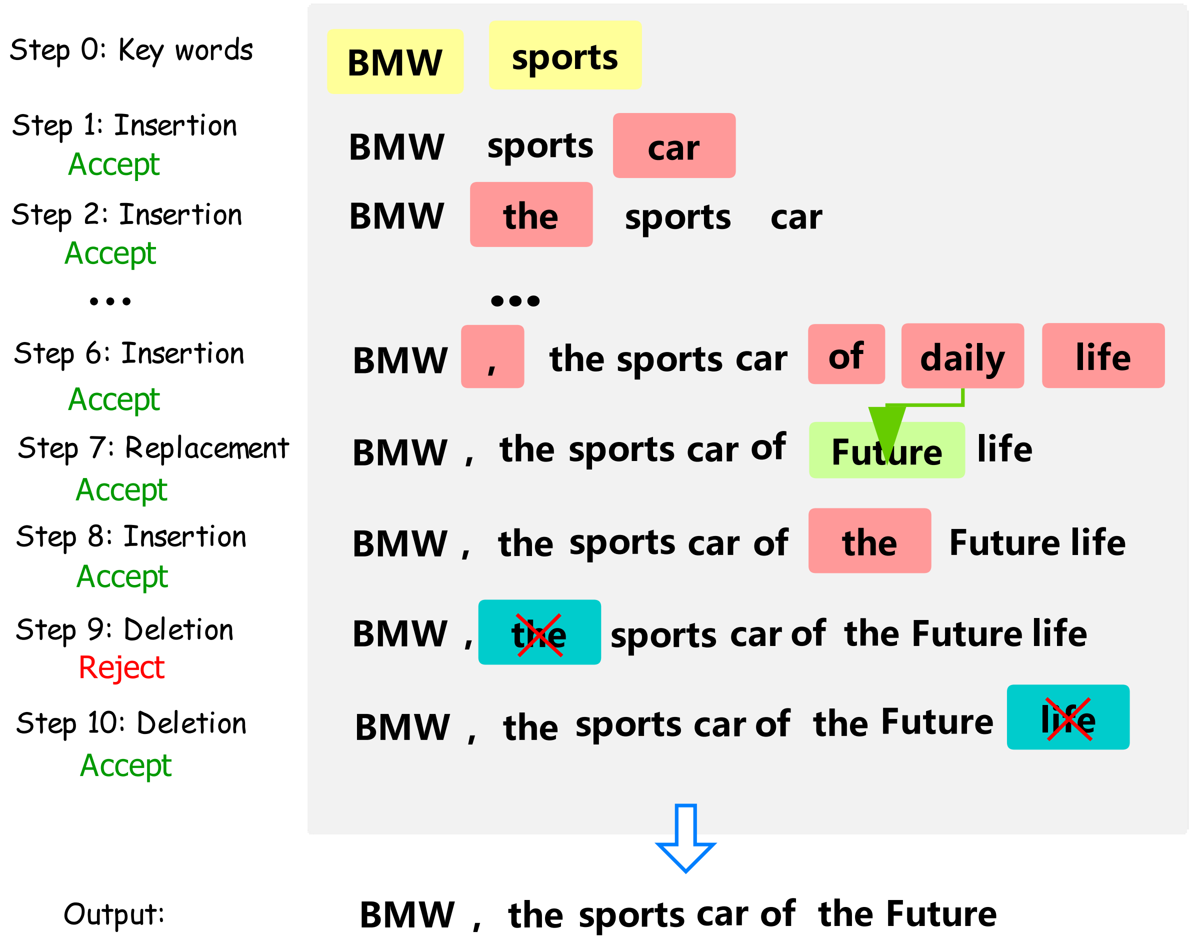


Motivation

- We need a practical method for sentence generation under both hard and soft constraints! So we propose **Constrained Generation by Metropolis-Hastings sampling (CGMH).**

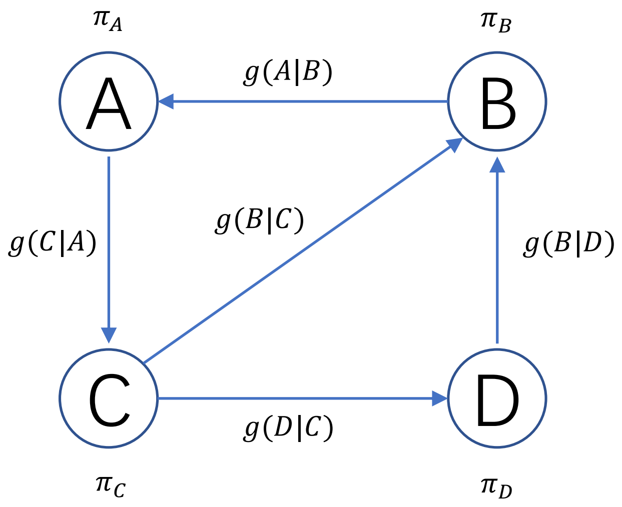
Introduction

- The main idea of CGMH is performing Metropolis-Hastings **sampling directly in sentence space**. The figure on the right illustrates CGMH by an example of generating advertisement from keywords.



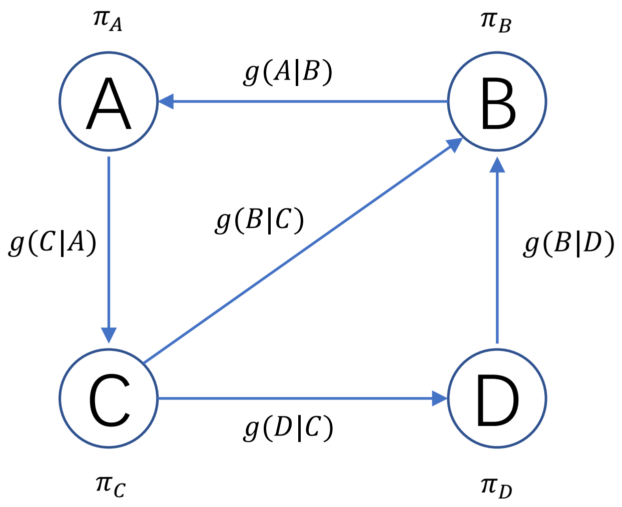
Introduction - Metropolis-Hastings Sampling

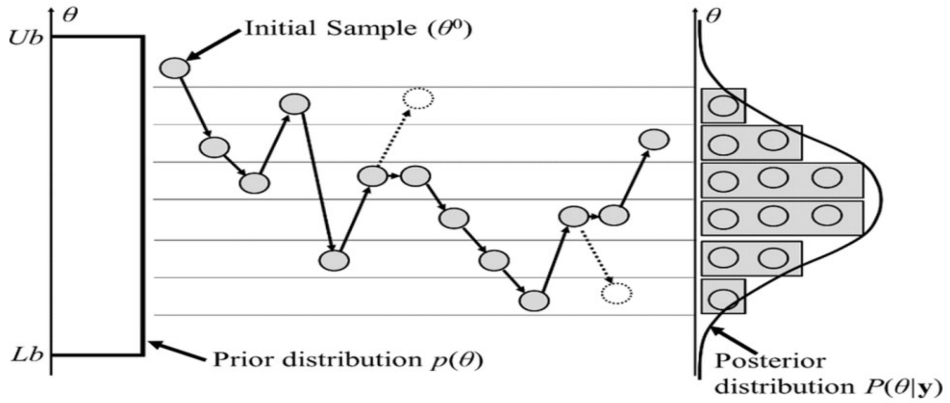
- Metropolis-Hastings(MH) sampling is a 2-step **Markov Chain Monte Carlo** (MCMC) algorithm



Introduction - Metropolis-Hastings Sampling

- Metropolis-Hastings(MH) sampling is a 2-step **Markov Chain Monte-Carlo** (MCMC) algorithm
- MH first **proposes** a transition, and then **accepts or rejects** the transition. (Gibbs sampling is a special case of MH sampling which always accepts transitions.)





Method – Stationary Distribution

- We set the stationary distribution as:

$$\pi(x) = P(x) \cdot P_C(x)$$

Method – Stationary Distribution

➢ We set the stationary distribution as:

$$\pi(x) \ = P(x) \cdot P_C(x)$$

- $P\ (x) = \prod_t P(x_t|x_{0:t-1})$ is the probability of sentence in a general-purpose language model.

Method – Stationary Distribution

➢ We set the stationary distribution as:

$$\pi(x) = P(x) \cdot P_C(x)$$

- $P(x) = \prod_t P(x_t|x_{0:t-1})$ is the probability of sentence in a general-purpose language model.
- $P_C(x) = \prod_{\mathrm{i}} P_C^i(x)$ is the indicator function showing whether constraints are satisfied.

Method – Stationary Distribution

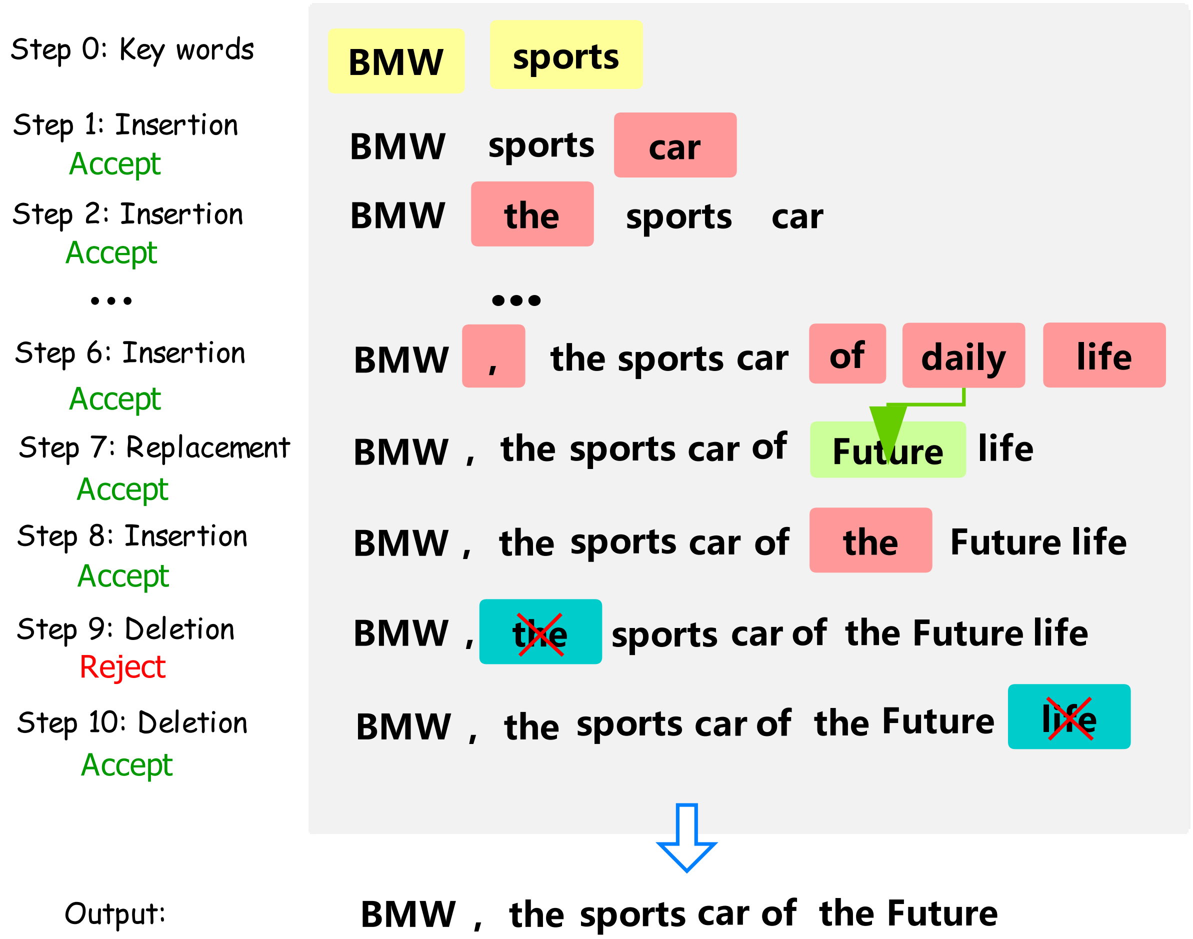
- We set the stationary distribution as:

$$\pi(x) = P(x) \cdot P_C(x)$$

  - $P(x) = \prod_t P(x_t | x_{0:t-1})$ is the probability of sentence in a general-purpose language model.
  - $P_C(x) = \prod_i P_C^i(x)$ is the indicator function showing whether constraints are satisfied.
  - For different tasks, we use different $P_C(x)$:
    - Keywords2Sentence: $P_C(x) = 1_{\{x \text{ contains the keywords}\}}$
    - Paraphrase: $P_C(x) = 1 \,/\, P_C^{KW}(x) \,/\, P_C^{KW}(x)\, P_C^{SIM}(x)$
    - Correction: $P_C(x) = 1 \,/\, P_C^{WMA}(x)$

Method – Proposal

- We use MH algorithm to sample from $\pi(x)$
  - From a sentence x_{t-1}, we propose a new sentence x' by **replacement / insertion / deletion** of a position from x_{t-1}



Method –Accept/Reject

- Calculate the acceptance rate:

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

Method –Accept/Reject

- Calculate the acceptance rate:

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

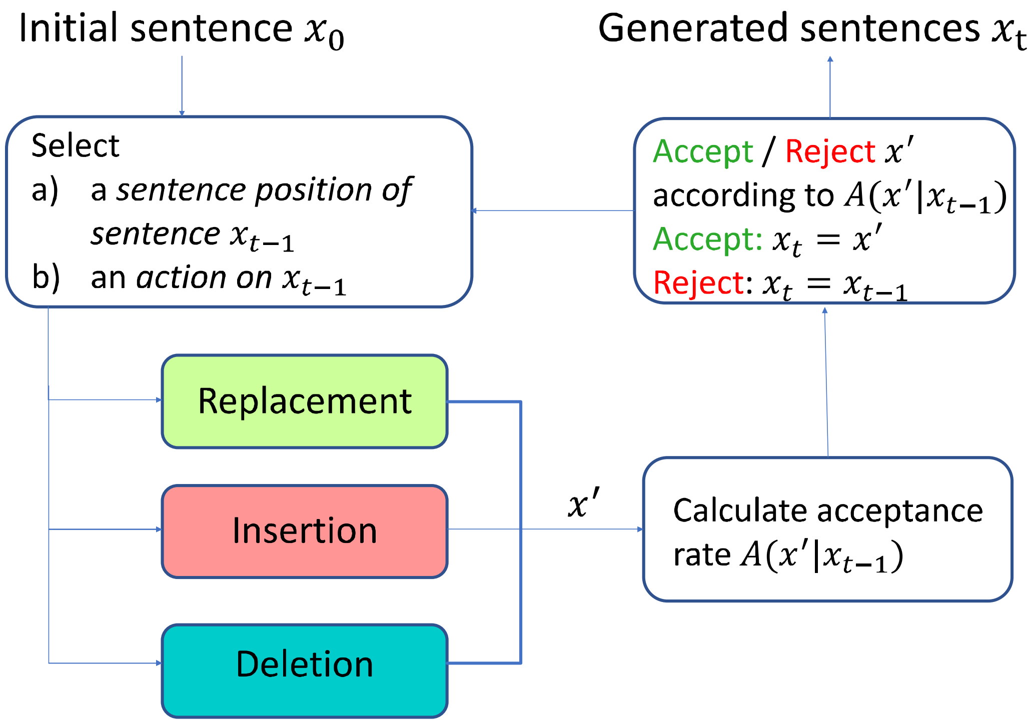
- Accept x' with probability $A(x'|x_{t-1})$

Method –Accept/Reject

- Calculate the acceptance rate:

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

- Accept x' with probability $A(x'|x_{t-1})$



Experiment

We test CGMH on three different tasks.

Experiment

We test CGMH on three different tasks.

- Keywords2Sentence Generation
  - Aim: To generate fluent sentences containing the given set of words.
  - Dataset: A subset of One-Billion-Word Corpus (5M)

Experiment

We test CGMH on three different tasks.

- Keywords2Sentence Generation
  - Aim: To generate fluent sentences containing the given set of words.
  - Dataset: A subset of One-Billion-Word Corpus (5M)
- Unsupervised Paraphrase Generation
  - Aim: To generate sentences with similar meaning of the given one.
  - Dataset: Quora(140k pairs of paraphrase sentences)

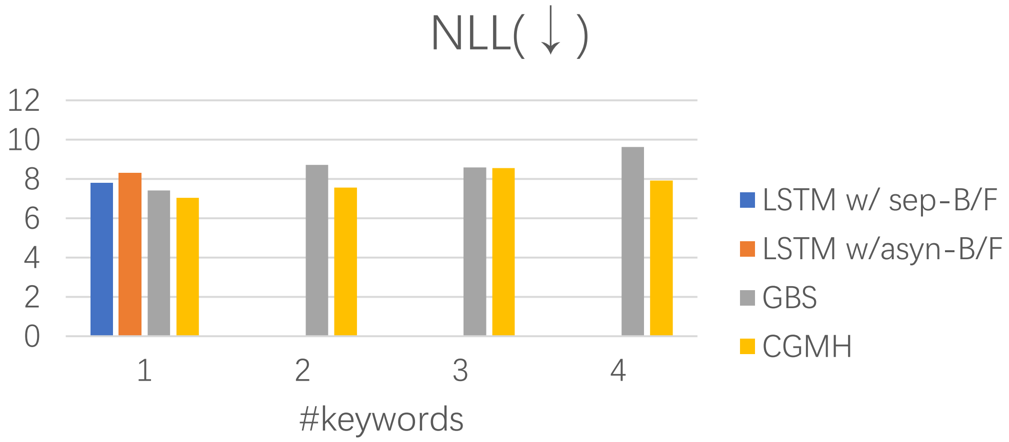
Experiment

We test CGMH on three different tasks.

- **Keywords2Sentence Generation**
  - Aim: To generate fluent sentences containing the given set of words.
  - Dataset: A subset of One-Billion-Word Corpus (5M)
- **Unsupervised Paraphrase Generation**
  - Aim: To generate sentences with similar meaning of the given one.
  - Dataset: Quora(140k pairs of paraphrase sentences)
- **Sentence Correction**
  - Aim: To correct the errors in the given sentence.
  - Dataset: A subset of One-Billion-Word Corpus (5M, base language model) and JFLEG(1501 sentences, for test only)

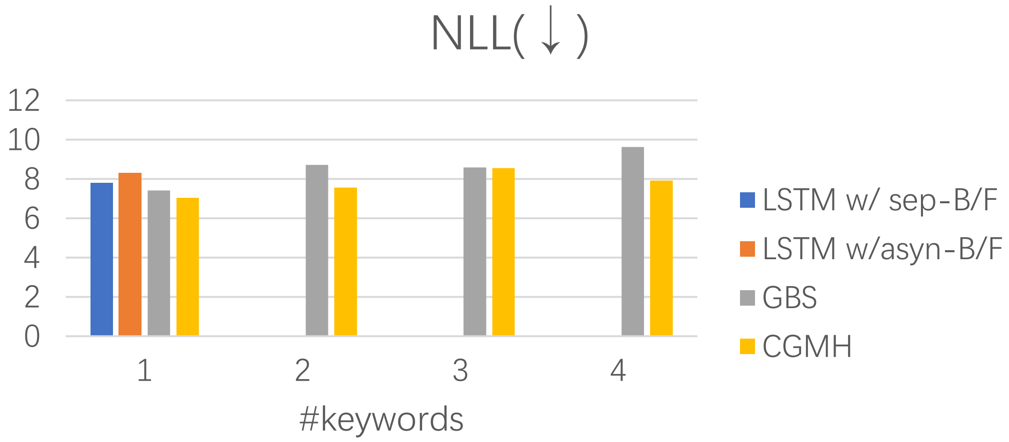
Experiment - Keywords2Sentence Generation

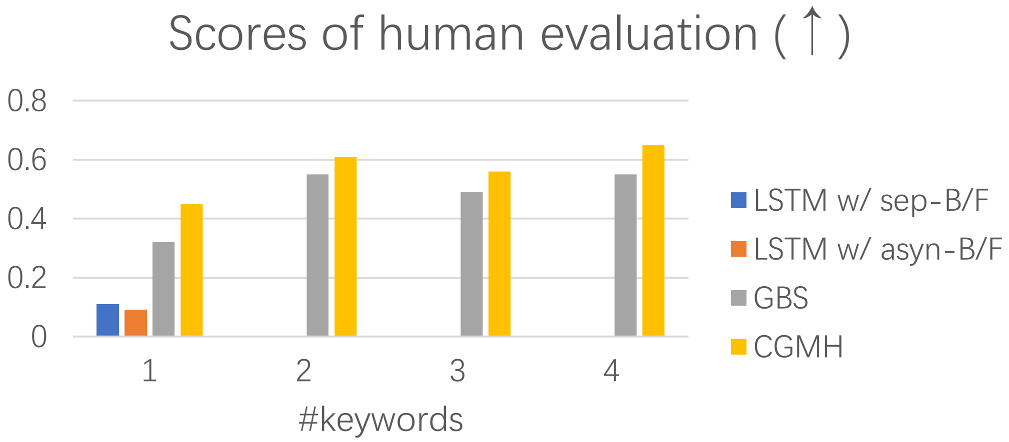
- To generate sentences from a variable number of keywords, we simply start sampling from the given keywords. Experimental results show that CGMH outperforms previous work in both NLL and human evaluations.



Experiment - Keywords2Sentence Generation

- To generate sentence from a variable number of keywords, we simply start sampling from the given keywords. Experimental results show that CGMH outperforms previous work in both NLL and human evaluations.

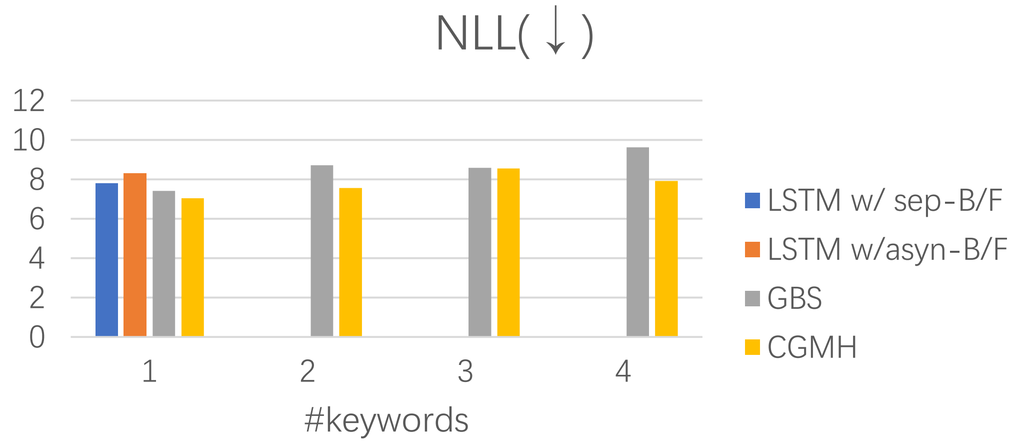


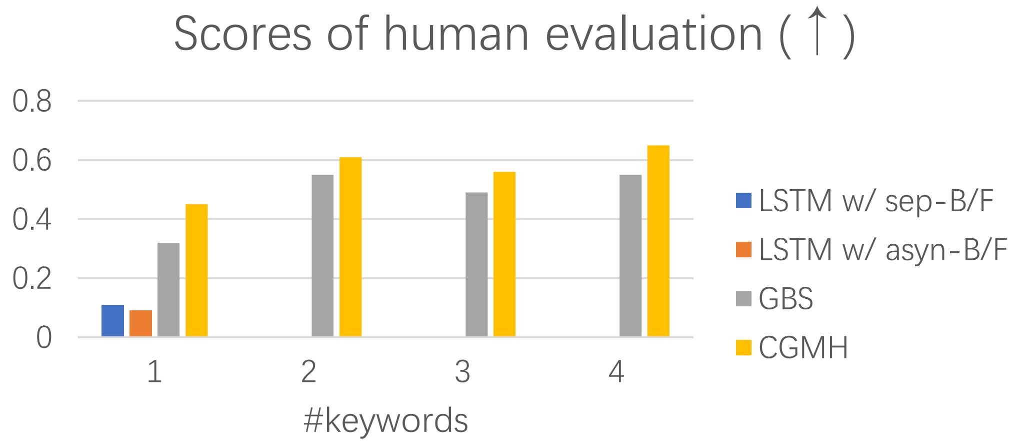


ByteDance字节跳动

Experiment - Keywords2Sentence Generation

➢ To generate sentence from a variable number of keywords, we simply start sampling from the given keywords. Experimental results show that CGMH outperforms previous work in both NLL and human evaluations.





Keyword(s)	Generated Sentences
friends	My good friends were in danger .
project	The first project of the scheme .
have, trip	But many people have never made the trip .
lottery, scholarships	But the lottery has provided scholarships.
decision, build, home	The decision is to build a new home.
attempt, copy, painting, denounced	The first attempt to copy the painting was denounced.

Experiment - Unsupervised Paraphrase Generation

- In order to generate paraphrases, we set $P_C(x)$ to be a measure of semantical similarity between generated sentences x and the given one x_0. We tried several kinds of similarity measures.

Experiment - Unsupervised Paraphrase Generation

- In order to generate paraphrases, we set $P_C(x)$ to be a measure of semantical similarity between generated sentences x and the given one x_0. We tried several kinds of similarity measures.
  - **CGMH w/o matching**, $P_C\ (x)$ always equals 1. We use it for comparison..

Experiment - Unsupervised Paraphrase Generation

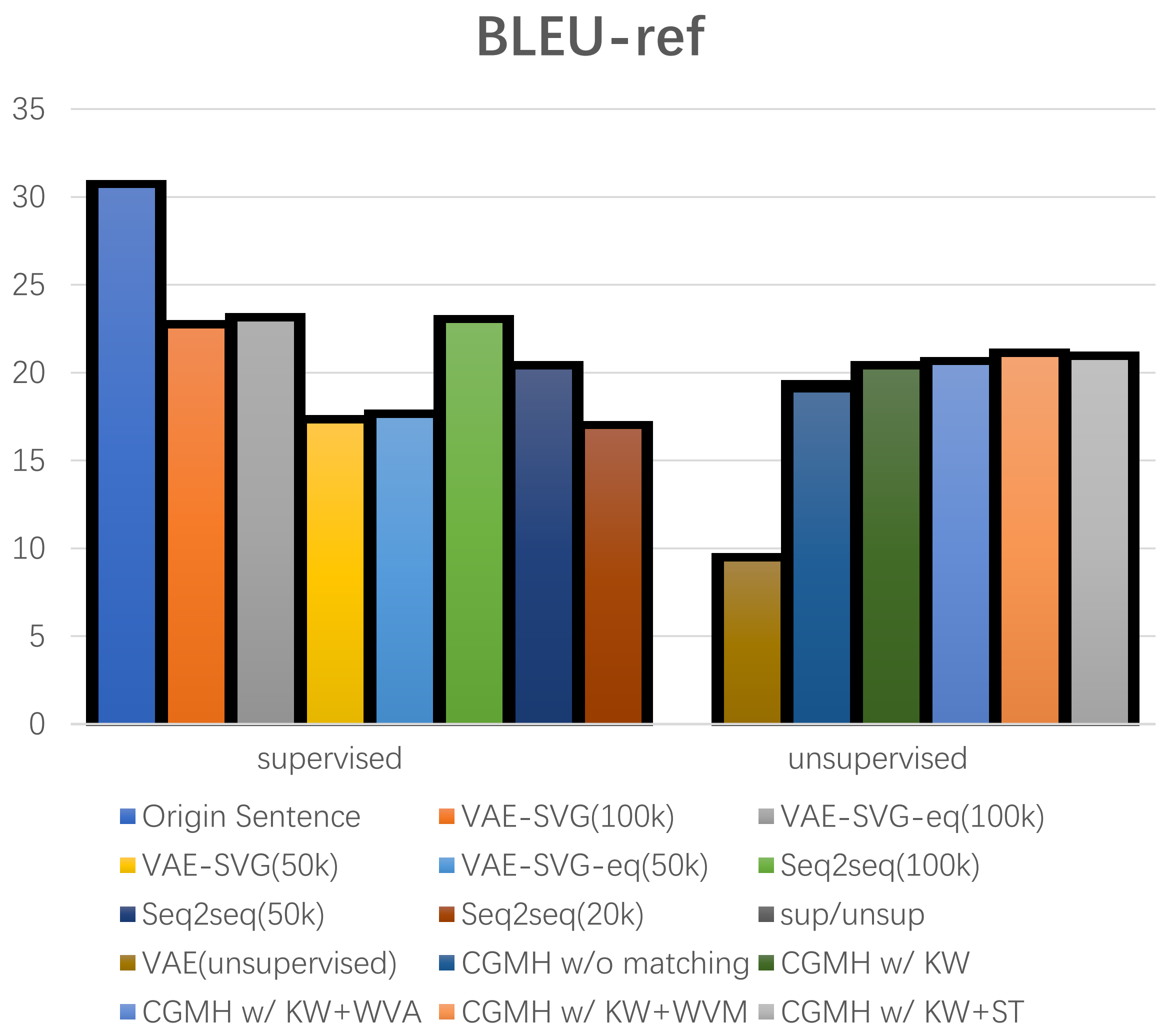
- In order to generate paraphrases, we set $P_C(x)$ to be a measure of semantical similarity between generated sentences x and the given one x_0. We tried several kinds of similarity measures.
  - **CGMH w/o matching**, $P_C\ (x)$ always equals 1. We use it for comparison.
  - **CGMH w/ KW**, $P_C\ (x) = P_C^{KW}(x),\ P_C^{KW}(x)$=1 if x still contains the keywords of x_0, which ensures that important information of x_0 won't be forgotten.

Experiment - Unsupervised Paraphrase Generation

- In order to generate paraphrases, we set $P_C(x)$ to be a measure of semantical similarity between generated sentences x and the given one x_0. We tried several kinds of similarity measures.
  - **CGMH w/o matching**, $P_C\ (x)$ always equals 1. We use it as a baseline.
  - **CGMH w/ KW**, $P_C\ (x) = P_C^{KW}(x),\ P_C^{KW}(x)$=1 if x still contains the keywords of x_0, which ensures that important information of x_0 won't be forgotten.
  - **CGMH w/ KW+**SIM, $P_C\ (x) = P_C^{KW}(x)\ P_C^{SIM}(x),\ P_C^{SIM}\ (x)$ is the cosine similarity of sentences embeddings. If SIM=**WVA**, sentence embeddings are calculated as the mean vectors of word embeddings. If SIM=**ST**, we get sentence embeddings by SkipThoughts. And if SIM=**WVM**, we calculate maximal word similarities between each word in x with words in x_0, and use their average value as $P_C^{WVM}(x)$.

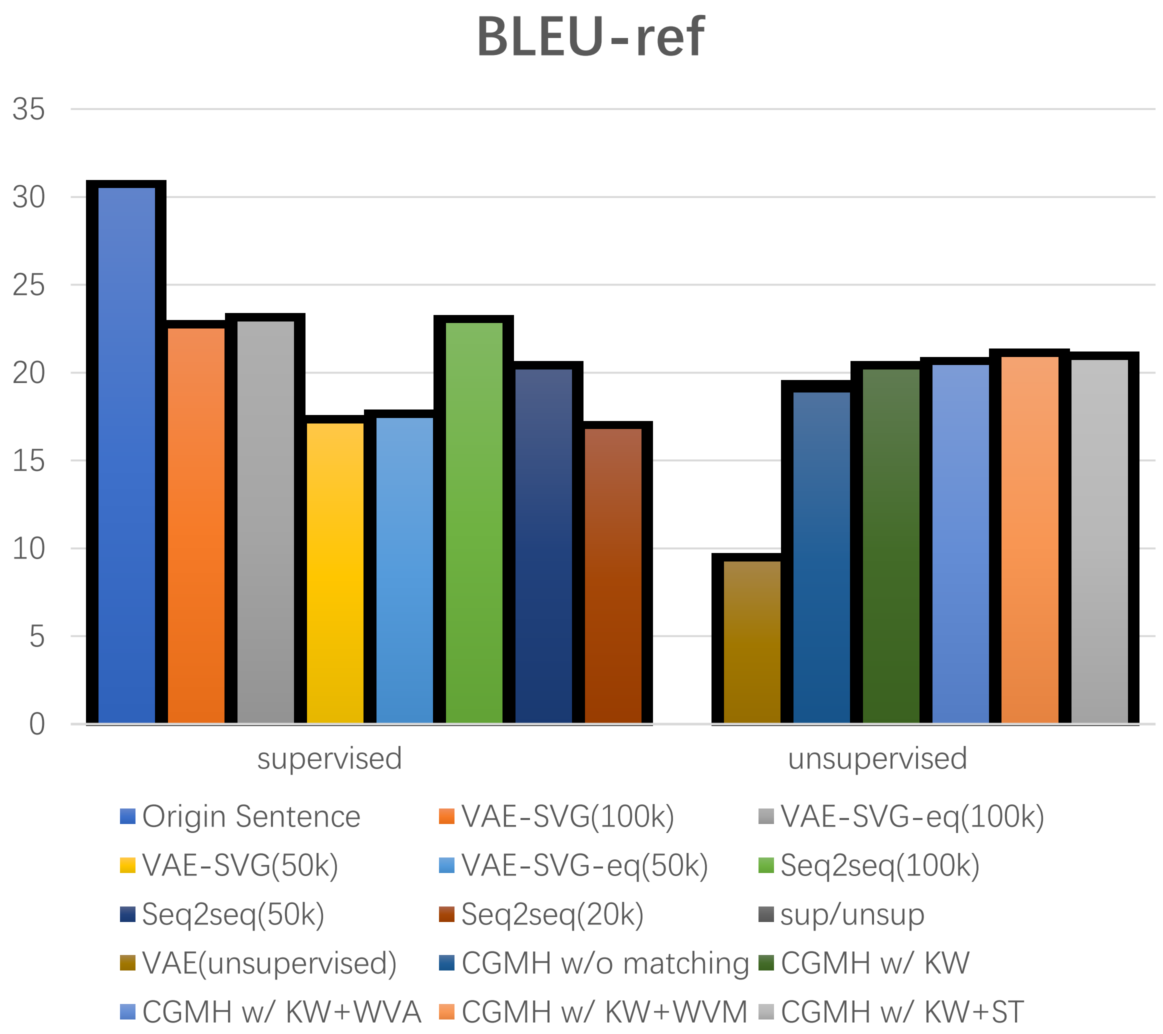
Experiment - Unsupervised Paraphrase Generation

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



Experiment - Unsupervised Paraphrase Generation

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



Examples

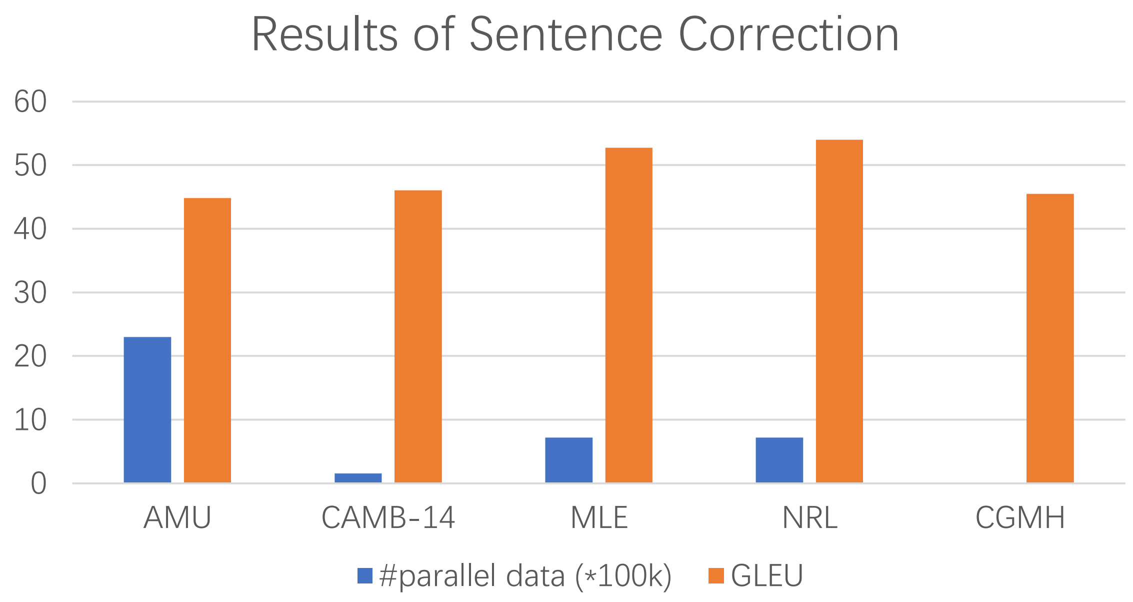
1,what ‘s the best plan to lose weight ->
what ‘s the best way to slim down quickly

2. how should i control my emotion ->
how do i control my anger

3. why do my dogs love to eat tuna fish ->
why do my dogs like to eat raw tuna and raw fish

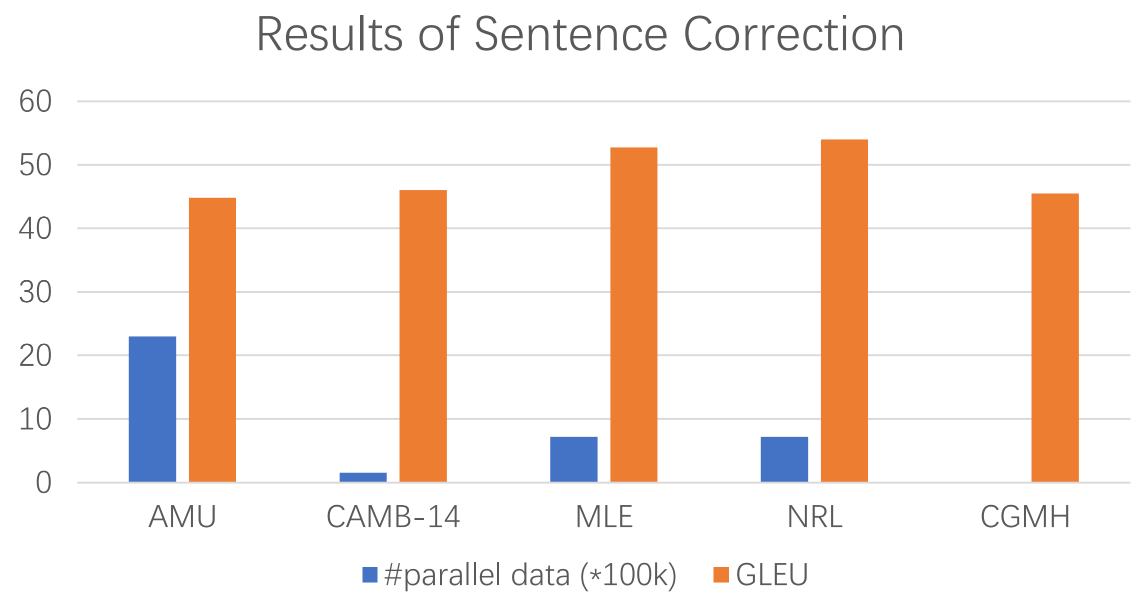
Experiment - Unsupervised Error Correction

- CGMH outperforms some of the supervised models trained on large parallel corpus.



Experiment - Unsupervised Error Correction

- CGMH outperforms some of the supervised models trained on large parallel corpus.

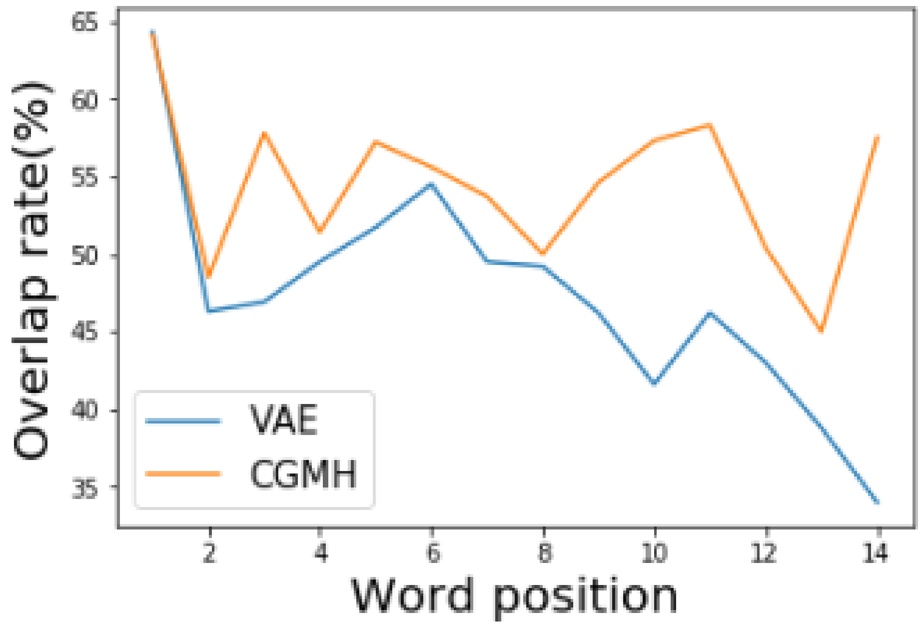


Erroneous sen1	**Even if we are failed , we have to try to get a new things .**
Reference sen1	Even if we all failed , we have to try to get new things .
Output sen1	Even if we are failing , we have to try to get some new things ..
Erroneous sen2	In the world oil price very high right now .
Reference sen2	In today 's world , oil prices are very high right now .
Output sen2	In the world , oil prices are very high right now .

Analysis

➢ Why CGMH outperforms sequential models?

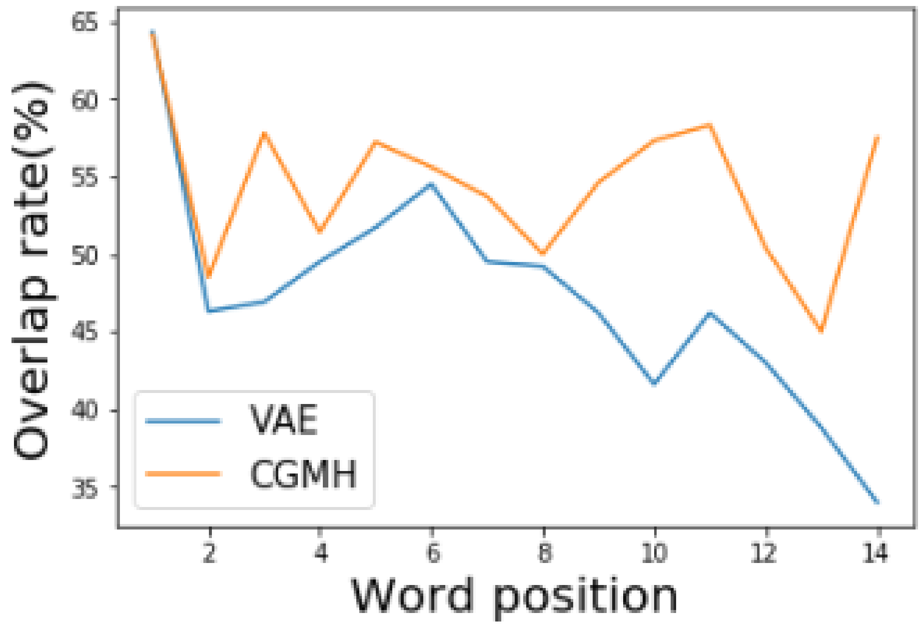
- RNN can be thought of as an autoregressive Bayesian network generating words conditioned on previous ones. Hence **error will accumulate** during generation.



Analysis

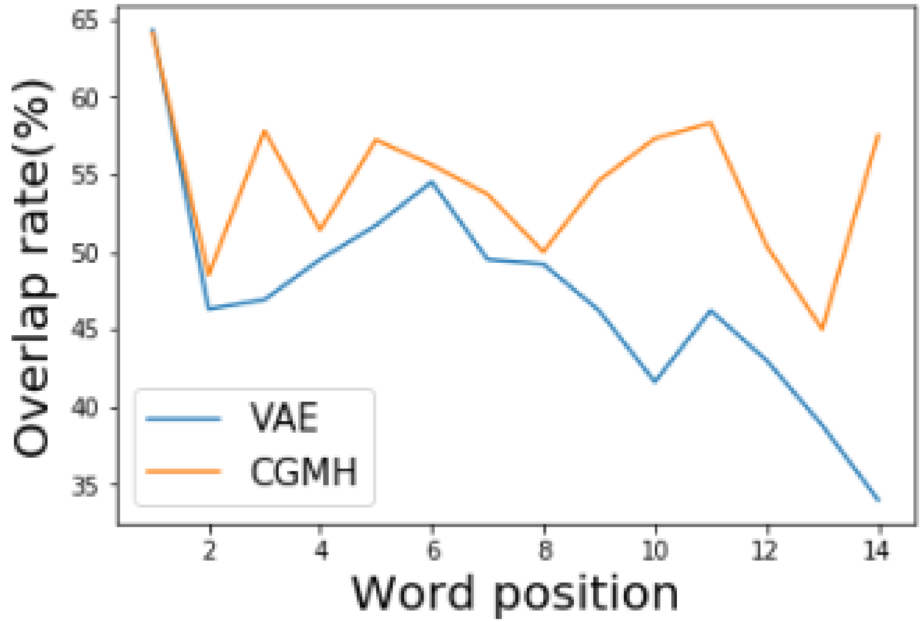
➢ Why CGMH outperforms sequential models?

- RNN can be thought of as an autoregressive Bayesian network generating words conditioned on previous ones. Hence **error will accumulate** during generation.
- CGMH doesn't generate sequentially, so error won't accumulate.



Analysis

- Why CGMH outperforms sequential models?
  - RNN can be thought of as an autoregressive Bayesian network generating words conditioned on previous ones. Hence **error will accumulate** during generation.
  - CGMH doesn't generate sequentially, so error won't accumulate.
  - At the same time, CGMH has the ability of **self-correction**. Please refer to the part of sentence correction.



Reference

[1] Hokamp, C., and Liu, Q. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *ACL*.

[2] Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2017. Guided open vocabulary image captioning with constrained beam search. In EMNLP.

[3] Gupta, A.; Agarwal, A.; Singh, P.; and Rai, P. 2017. A deep generative framework for paraphrase generation. *arXiv preprint arXiv:1709.05074*.

[4] Mou, L.; Yan, R.; Li, G.; Zhang, L.; and Jin, Z. 2015. Backward and forward language modeling for constrained sentence generation. *arXiv preprint arXiv:1512.06612*.

[5] Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In CoNLL.

[6] Li, Z.; Jiang, X.; Shang, L.; and Li, H. 2017. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

[7] Junczys-Dowmunt, M., and Grundkiewicz, R. 2016. Phrasebased machine translation is state-of-the-art for automatic grammatical error correction. *arXiv preprint arXiv:1605.06353*.

[8] Felice, M.; Yuan, Z.; Andersen, Ø. E.; Yannakoudakis, H.; and Kochmar, E. 2014. Grammatical error correction using hybrid systems and type filtering. In *CoNLL*.

[9] Napoles, C.; Sakaguchi, K.; Post, M.; and Tetreault, J. 2015. Ground truth for grammatical error correction metrics. In *ACL*.

THANKS.

ByteDance 字节跳动