A Contrastive Framework for Neural Text Generation

Yixuan Su* Tian Lan** Yan Wang** Dani Yogatama+ Lingpeng Kong++ Nigel Collier*
*Language Technology Lab, University of Cambridge
**Tencent AI Lab +DeepMind
++Department of Computer Science, The University of Hong Kong

221005 (수)
Joosung Yoon

Overview

Contrastive approach for
Finetuning training loss & decoding strategy

문제제기:
maximization-based decoding methods (e.g., beam search) of neural language models often lead to degenerate solutions
ex) undesirable repetitions

특정 토큰의 확률을 줄이는 unlikelihood training등의 대안이 문제를 완화하긴하지만 coherence가 부족

본 논문에서 보여준 것:
an underlying reason for model degeneration is the anisotropic distribution of token representations.

해결방법:
(i) SimCTG, a contrastive training objective to calibrate the model's representation space.
-> anisotropic 해소하겠다
(ii) (ii) a decoding method—contrastive search—to encourage diversity while maintaining coherence in the generated text.
-> 다양하게 뽑지만 coherence 유지해보겠다

```
import torch
# load the language model
from simctg.simctggpt import SimCTGGPT model_name
 = r'cambridgeltl/simctg_wikitext103' model = SimCTGGPT(model_na
me)
model.eval()
tokenizer = model.tokenizer
# prepare input
prefix_text = # The prefix text in Table 4
print ('Prefix is: {}'.format(prefix_text))
tokens = tokenizer.tokenize(prefix_text)
input_ids = tokenizer.convert_tokens_to_ids(tokens) input_ids =
 torch.LongTensor(input_ids).view(
1,-1)
# generate result with contrastive search
beam_width, alpha, decoding_len = 8, 0.6, 128
output = model.fast_contrastive_search(input_ids=input_ids,
                        beam_width=beam_width, alpha=alpha,
decoding_len=decoding_len) print("Output:\n" + 100 * '-')
print(tokenizer.decode(output))
```

```
def ranking_fast(context_hidden, next_hidden, next_top_k_probs,
alpha, beam_width):
    '''
        context_hidden: bsz*beam x seqlen x embed_dim
        next_hidden: bsz*beam x 1 x embed_dim
        next_top_k_probs: bsz x beam
    '''
    _, context_len, embed_dim = context_hidden.size()
    norm_context_hidden = context_hidden / context_hidden.norm(
dim=2, keepdim=True)
    norm_next_hidden = next_hidden / next_hidden.norm(dim=2,
keepdim=True)
    cosine_matrix = torch.matmul(norm_context_hidden, norm_next_
hidden.transpose(
1,2)).squeeze(-1)    # [B*K, S]
    scores, _ = torch.max(cosine_matrix, dim=-1)    # [B*K]
    next_top_k_probs = next_top_k_probs.view(-1)    # [B*K]
    scores = (1.0 - alpha) * next_top_k_probs - alpha * scores
    scores = torch.stack(torch.split(scores, beam_width))
# [B, K]
    selected_idx = scores.max(dim=-1)[1]    # [B]
    return selected_idx
```

https://github.com/yxuansu/SimCTG/blob/fa52955131d6533fc8388277eb068b9e2de2f987/document_generation/utlis.py

01
Introduction

token representation
distribution의 비대칭이 문제

MLE 기반의 학습과 most likely sequence로
디코딩하는건 충분하지 않을 수 있다
-> Degeneration
-> token, phrase, sent level의 중복문장 생성

---

less likely vocab에서 샘플링하는 방법이 제안되었지만 의미적으로 안
맞거나 반대 텝이 나오기도함 (unlikelihood training도 비슷한 케이스)

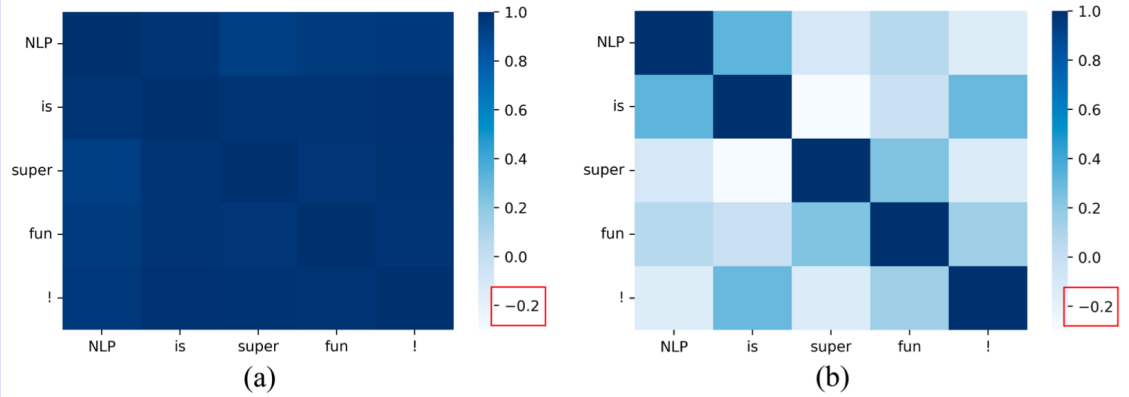

Figure 1: Token cosine similarity matrix of (a) GPT-2 and (b) SimCTG. (best viewed in color)

모두 다 token representation distribution이 불균등(anisotropic)하기 때문임

본문에서 제안하는 모델

SimCTG (a simple contrastive framework for neural text generation)

(i) at each decoding step, the output should be selected from the set of most probable candidates predicted by the model to better maintain the semantic coherence between the generated text and the human-written prefix

(ii) the sparseness of the token similarity matrix of the generated text should be preserved to avoid degeneration.

Background

- MLE로 학습하는 LM은 transformer-based model 구조에서 모델의 표현이 anisotropic distribution을 갖게 됨
- Deterministic Sampling은 greedy, beam이고 highest probability에 의존해서 degeneration을 야기함
- Stochastic Sampling은 top-k, nucleus sampling류임. 가끔 의미적으로 반대되는 단어까지 생성하기도 하는 부작용 있음

2.1 Language Modelling

The goal of language modelling is to learn a probability distribution $p_\theta(\boldsymbol{x})$ over a variable-length text sequence $\boldsymbol{x} = \{x_1, ..., x_{|\boldsymbol{x}|}\}$, where θ denotes model parameters. Typically, the maximum likelihood estimation (MLE) objective is used to train the language model which is defined as

$$\mathcal{L}_{\text{MLE}} = -\frac{1}{|\boldsymbol{x}|}\sum_{i=1}^{|\boldsymbol{x}|}\log p_\theta(x_i|\boldsymbol{x}_{<i}). \quad (1)$$

However, as observed in many recent studies [10, 9, 44], training with likelihood maximization objective often yields an anisotropic distribution of model representations (especially for Transformer-based models) that undermines the model's capacity.

2.2 Open-ended Text Generation

In this work, we focus on studying the task of open-ended text generation due to its generality in various applications, such as story generation [11, 43], contextual text completion [36], poetry generation [23], and dialogue systems [48]. Formally, conditioned on a human-written prefix (i.e., context) $\boldsymbol{x}$, the task is to decode a continuation $\hat{\boldsymbol{x}}$ from the language model and the resulting text is $\{x_1, .., x_{|\boldsymbol{x}|}, \hat{x}_{|\boldsymbol{x}|+1}, ..., \hat{x}_{|\boldsymbol{x}|+|\hat{\boldsymbol{x}}|}\}$. Typically, there are two classes of methods used for decoding, which are (1) deterministic methods and (2) stochastic methods.

Deteriminstic Methods. Two widely used deterministic approaches are greedy and beam search which aim to select the text continuation with highest probability based on the model's probability distribution p_θ. However, solely maximizing the output probability often leads to dullness [22] and degeneration [11, 14] in the generated text.

Stochastic Methods. To remedy the issues of deterministic decoding, several approaches have been proposed to sample from p_θ. To avoid sampling from the unreliable tail of distribution, Fan *et al.* [11] proposed top-k sampling which draws sample from the vocabulary subset $V^{(k)}$ that maximizes $\sum_{v\in V^{(k)}} p_\theta(v|\boldsymbol{x})$. Here, $|V^{(k)}| = k$ and $\boldsymbol{x}$ is the prefix context. Differently, the current state-of-the-art nucleus sampling [14] draws sample from the smallest vocabulary subset U with total probability mass above a threshold $p \in [0, 1]$; i.e., U is the smallest vocabulary subset such that $\sum_{v\in U} p_\theta(v|\boldsymbol{x}) \geq p$. While the sampling approaches help to alleviate model degeneration, the intrinsic stochasticity in these methods could cause the semantic meaning of the sampled text to diverge from or even contradict to the human-written prefix [3].

02
Methodology

Training: Contrastive Training
Decoding: Contrastive Search

3.1 Contrastive Training

Our goal is to encourage the language model to learn discriminative and isotropic token representations. To this end, we introduce a contrastive objective $\mathcal{L}_{\text{CL}}$ into the training of the language model. Specifically, given a variable-length sequence $\boldsymbol{x} = \{x_1, ..., x_{|\boldsymbol{x}|}\}$, the $\mathcal{L}_{\text{CL}}$ is defined as

$$\mathcal{L}_{\text{CL}} = \frac{1}{|\boldsymbol{x}|\times(|\boldsymbol{x}|-1)} \sum_{i=1}^{|\boldsymbol{x}|} \sum_{j=1, j\neq i}^{|\boldsymbol{x}|} \max\{0, \rho - s(h_{x_i}, h_{x_i}) + s(h_{x_i}, h_{x_j})\}, \tag{2}$$

where $\rho \in [-1, 1]$ is a pre-defined margin and h_{x_i} is the representation of token x_i produced by the model. The similarity function s computes the cosine similarity between token representations as

$$s(h_{x_i}, h_{x_j}) = \frac{h_{x_i}^{\top} h_{x_j}}{\|h_{x_i}\| \cdot \|h_{x_j}\|}. \tag{3}$$

Intuitively, by training with $\mathcal{L}_{\text{CL}}$, the model learns to pull away the distances between representations of distinct tokens.[2] Therefore, a discriminative and isotropic model representation space can be obtained. The overall training objective $\mathcal{L}_{\text{SimCTG}}$ is then defined as

$$\mathcal{L}_{\text{SimCTG}} = \mathcal{L}_{\text{MLE}} + \mathcal{L}_{\text{CL}}, \tag{4}$$

where the maximum likelihood estimation (MLE) objective $\mathcal{L}_{\text{MLE}}$ is described in Eq. (1). Note that, when the margin ρ in $\mathcal{L}_{\text{CL}}$ equals to 0, the $\mathcal{L}_{\text{SimCTG}}$ degenerates to the vanilla MLE objective $\mathcal{L}_{\text{MLE}}$.

3.2 Contrastive Search

We propose a novel decoding method, *contrastive search*. At each decoding step, the key ideas of contrastive search are (i) the generated output should be selected from the set of most probable candidates predicted by the model; and (ii) the generated output should be discriminative enough with respect to the previous context. In this way, the generated text can (i) better maintain the semantic coherence with respect to the prefix while (ii) avoiding model degeneration.

Formally, given the previous context $\boldsymbol{x}_{<t}$, at time step t, the selection of the output x_t follows

$$x_t = \underset{v\in V^{(k)}}{\arg\max} \Big\{ (1-\alpha) \times \underbrace{p_\theta(v|\boldsymbol{x}_{<t})}_{\text{model confidence}} - \alpha \times \underbrace{(\max\{s(h_v, h_{x_j}) : 1 \leq j \leq t-1\})}_{\text{degeneration penalty}} \Big\}, \tag{5}$$

[2] By definition, the cosine similarity $s(h_{x_i}, h_{x_i})$ of the identical token x_i is 1.0.

Contrastive Training

- Cosine sim 기준으로 유사한 토큰들이 더 큰 loss를 받는 구조
- 붙어 있는 토큰들을 더 멀리 떨어뜨리게 하는 효과
- ρ 값이 0 이면 적용 안 하는거나 마찬가지 (MLE만 쓰는 구조)

Contrastive Search

- 모델이 예측한 셋 안에서 확률 높은 후보들이되, 이전 문맥과 구분이 될 수 있어야함
- 토큰 생성에 대한 확률값에 해당 토큰의 hidden states와 이전 토큰들의 hidden states의 유사도중 max값을 뽑아서 penalty term으로 줌
- Q) token들이 많으면 이거 계산시간 오래 걸리지 않을까? Matmul로 해결

```
def ranking_fast(context_hidden, next_hidden, next_top_k_probs,
alpha, beam_width):
    '''
        context_hidden: bsz*beam x seqlen x embed_dim
        next_hidden: bsz*beam x 1 x embed_dim
        next_top_k_probs: bsz x beam
    '''
    _, context_len, embed_dim = context_hidden.size()
    norm_context_hidden = context_hidden / context_hidden.norm(
dim=2, keepdim=True)
    norm_next_hidden = next_hidden / next_hidden.norm(dim=2,
keepdim=True)
    cosine_matrix = torch.matmul(norm_context_hidden, norm_next_
hidden.transpose(
1,2)).squeeze(-1)    # [B*K, S]
    scores, _ = torch.max(cosine_matrix, dim=-1)    # [B*K]
    next_top_k_probs = next_top_k_probs.view(-1)    # [B*K]
    scores = (1.0 - alpha) * next_top_k_probs - alpha * scores
    scores = torch.stack(torch.split(scores, beam_width))
# [B, K]
    selected_idx = scores.max(dim=-1)[1]    # [B]
    return selected_idx
```

03
Document Generation

Training & Decoding 방법들에 대한 ablation studies

Experiment settings

Baseline

Finetuning GPT-2
on evaluated benchmark

[1] MLE GPT2
[2] unlikelihood GPT2

Evaluation benchmark

- Eval Dataset:Wikitext-103
- SimCTG, MLE finetuning Wikitext-103 (40k training steps)
- UL finetuning (38.5K steps token-level, 1.5K steps sentence-level)
- bs: 128, max_seq_len: 256, optim: adam, lr: 2e-5
- Decoding은 prefix를 32~128 length정도 되는 정보를 주고 시작함
- deterministic method: greedy, beam (10)
- search stochastic method: p=0.95
- proposed contrastive search: k and α in Eq. (5) are set as 8 (top_k 8개 보고) and 0.6. (degeneration penalty에 점수를 좀 더 줬음)

Hyper params

Evaluation Metrics

[1]

Language Modelling Quality

(1) Perplexity on the test set of Wikitext-103.
(2) Prediction Accuracy (토큰 맞추기)
(3) Prediction Repetition (next token의 top-1 예측이 prefix(이전입력)에 있으면 카운팅 됨). 낮은 게 좋음

Model	Language Modelling Quality				Generation Quality							
	ppl↓	acc↑	rep↓	wrep↓	Method	rep-2↓	rep-3↓	rep-4↓	diversity↑	MAUVE↑	coherence↑	gen-ppl
MLE	24.32	39.63	52.82	29.97	greedy	69.21	65.18	62.05	0.04	0.03	0.587	7.32
					beam	71.94	68.97	66.62	0.03	0.03	0.585	6.42
					nucleus	4.45	0.81	0.43	0.94	0.90	0.577	49.71
					contrastive	44.20	37.07	32.44	0.24	0.18	0.599	9.90
Unlike.	28.57	38.41	**51.23**	**28.57**	greedy	24.12	13.35	8.04	0.61	0.69	0.568	37.82
					beam	11.83	5.11	2.86	0.81	0.75	0.524	34.73
					nucleus	4.01	0.80	0.42	**0.95**	0.87	0.563	72.03
					contrastive	7.48	3.23	1.40	0.88	0.83	0.574	43.61
SimCTG	**23.82**	**40.91**	51.66	28.65	greedy	67.36	63.33	60.17	0.05	0.05	0.596	7.16
					beam	70.32	67.17	64.64	0.04	0.06	0.591	6.36
					nucleus	4.05	0.79	0.37	0.94	0.92	0.584	47.19
					contrastive	**3.93**	**0.78**	**0.31**	**0.95**	**0.94**	**0.610**	**18.26**
Human	-	-	36.19	-	-	3.92	0.88	0.28	0.95	1.00	0.644	24.01

Table 1: Evaluation results on Wikitext-103 test set. "Unlike." denotes the model trained with unlikelihood objective. ↑ means higher is better and ↓ means lower is better.

Prediction Repetition. The fraction of next-token (top-1) predictions that occur in the prefix which is defined as: **rep** $= \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\boldsymbol{x}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{t=1}^{|\boldsymbol{x}|}\mathbb{1}[\arg\max p_\theta(x|\boldsymbol{x}_{<t}) \in \boldsymbol{x}_{<t}]$.

In addition, the next token repetitions that do not equal to the ground truth token: **wrep** $= \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\boldsymbol{x}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{t=1}^{|\boldsymbol{x}|}\mathbb{1}[\arg\max p_\theta(x|\boldsymbol{x}_{<t}) \in \boldsymbol{x}_{<t} \wedge \neq x_t]$ is also reported.

4.1.2 Generation Quality

Generation Repetition. This metric measures the sequence-level repetition as the portion of duplicate n-grams in the generated text [54]. For a generated text continuation $\hat{\boldsymbol{x}}$, the repetion at n-gram level is defined as: **rep-n** $= 100 \times (1.0 - \frac{|\text{unique n-grams}(\hat{\boldsymbol{x}})|}{|\text{total n-grams}(\hat{\boldsymbol{x}})|})$.

Diversity. This metric takes into account the generation repetition at different n-gram levels and it is defined as: **diversity** $= \prod_{n=2}^{4}(1.0 - \frac{\text{rep-n}}{100})$. It can be deemed as an overall assessment of model degeneration. A lower diversity means a more severe degeneration of the model.

MAUVE [34] is a metric that measures the token distribution closeness between the generated text and human-written text. A higher MAUVE score means the model generates more human-like texts.

Semantic Coherence. To automatically measure the semantic coherence (i.e., consistency) between the prefix and the generated text, we employ the advanced sentence embedding method, SimCSE [13]. Specifically, given the prefix $\boldsymbol{x}$ and the generated text $\hat{\boldsymbol{x}}$, the coherence score is defined as: **coherence** $= v_{\boldsymbol{x}}^\top v_{\hat{\boldsymbol{x}}}/(\|v_{\boldsymbol{x}}\|\cdot\|v_{\hat{\boldsymbol{x}}}\|)$, where $v_{\boldsymbol{x}} = \text{SimCSE}(\boldsymbol{x})$ and $v_{\hat{\boldsymbol{x}}} = \text{SimCSE}(\hat{\boldsymbol{x}})$.

Perplexity of Generated Text. Lastly, we evaluate the perplexity of the generated text $\hat{\boldsymbol{x}}$ given the prefix $\boldsymbol{x}$, which is defined as: **gen-ppl** $= 2^{f(\mathcal{D},\theta)}$ and $f(\mathcal{D},\theta) = \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\hat{\boldsymbol{x}}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\log_2 p_\theta(\hat{\boldsymbol{x}}|\boldsymbol{x})$.
Importantly, the optimal approach should produce text which has a perplexity *close* to that of the human-written text [14]. A high gen-ppl means the generated text is very *unlikely* given the prefix, therefore being low quality. In contrastive, a low gen-ppl means the generated text has a low diversity and gets stuck in repetitive loops [14]. We use the model θ trained with $\mathcal{L}_{\text{SimCTG}}$ to measure the gen-ppl of different approaches, therefore making sure the numbers are comparable with each other.[6]

[2] Generation Quality

(1) Generation Repetition
(sentence-level에서 n-grams의 반복을 카운팅)
(2) Diversity (n-gram levels에서 repetition을 계산함)
(3) MAUVE (생성한거랑 human-written text와 token distribution closeness를 계산함)
(4) Semantic Coherence
(simCSE로 prefix와 generated text의 representation을 구해서 coherence score를 계산함)
(5) Perplexity of Generated Text

Model	Language Modelling Quality				Generation Quality							
	ppl↓	acc↑	rep↓	wrep↓	Method	rep-2↓	rep-3↓	rep-4↓	diversity↑	MAUVE↑	coherence↑	gen-ppl
MLE	24.32	39.63	52.82	29.97	greedy	69.21	65.18	62.05	0.04	0.03	0.587	7.32
					beam	71.94	68.97	66.62	0.03	0.03	0.585	6.42
					nucleus	4.45	0.81	0.43	0.94	0.90	0.577	49.71
					contrastive	44.20	37.07	32.44	0.24	0.18	0.599	9.90
Unlike.	28.57	38.41	**51.23**	**28.57**	greedy	24.12	13.35	8.04	0.61	0.69	0.568	37.82
					beam	11.83	5.11	2.86	0.81	0.75	0.524	34.73
					nucleus	4.01	0.80	0.42	**0.95**	0.87	0.563	72.03
					contrastive	7.48	3.23	1.40	0.88	0.83	0.574	43.61
SimCTG	**23.82**	**40.91**	51.66	28.65	greedy	67.36	63.33	60.17	0.05	0.05	0.596	7.16
					beam	70.32	67.17	64.64	0.04	0.06	0.591	6.36
					nucleus	4.05	0.79	0.37	0.94	0.92	0.584	47.19
					contrastive	**3.93**	**0.78**	**0.31**	**0.95**	**0.94**	**0.610**	**18.26**
Human	-	-	36.19	-	-	3.92	0.88	0.28	0.95	1.00	0.644	24.01

Table 1: Evaluation results on Wikitext-103 test set. "Unlike." denotes the model trained with unlikelihood objective. ↑ means higher is better and ↓ means lower is better.

Prediction Repetition. The fraction of next-token (top-1) predictions that occur in the prefix which is defined as: **rep** $= \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\boldsymbol{x}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{t=1}^{|\boldsymbol{x}|}\mathbb{1}[\arg\max p_\theta(x|\boldsymbol{x}_{<t}) \in \boldsymbol{x}_{<t}]$.

In addition, the next token repetitions that do not equal to the ground truth token: **wrep** $= \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\boldsymbol{x}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{t=1}^{|\boldsymbol{x}|}\mathbb{1}[\arg\max p_\theta(x|\boldsymbol{x}_{<t}) \in \boldsymbol{x}_{<t} \wedge \neq x_t]$ is also reported.

4.1.2 Generation Quality

Generation Repetition. This metric measures the sequence-level repetition as the portion of duplicate n-grams in the generated text [54]. For a generated text continuation $\hat{\boldsymbol{x}}$, the repetion at n-gram level is defined as: **rep-n** $= 100 \times (1.0 - \frac{|\text{unique n-grams}(\hat{\boldsymbol{x}})|}{|\text{total n-grams}(\hat{\boldsymbol{x}})|})$.

Diversity. This metric takes into account the generation repetition at different n-gram levels and it is defined as: **diversity** $= \prod_{n=2}^{4}(1.0 - \frac{\text{rep-n}}{100})$. It can be deemed as an overall assessment of model degeneration. A lower diversity means a more severe degeneration of the model.

MAUVE [34] is a metric that measures the token distribution closeness between the generated text and human-written text. A higher MAUVE score means the model generates more human-like texts.

Semantic Coherence. To automatically measure the semantic coherence (i.e., consistency) between the prefix and the generated text, we employ the advanced sentence embedding method, SimCSE [13]. Specifically, given the prefix $\boldsymbol{x}$ and the generated text $\hat{\boldsymbol{x}}$, the coherence score is defined as: **coherence** $= v_{\boldsymbol{x}}^\top v_{\hat{\boldsymbol{x}}}/(\|v_{\boldsymbol{x}}\|\cdot\|v_{\hat{\boldsymbol{x}}}\|)$, where $v_{\boldsymbol{x}} = \text{SimCSE}(\boldsymbol{x})$ and $v_{\hat{\boldsymbol{x}}} = \text{SimCSE}(\hat{\boldsymbol{x}})$.

Perplexity of Generated Text. Lastly, we evaluate the perplexity of the generated text $\hat{\boldsymbol{x}}$ given the prefix $\boldsymbol{x}$, which is defined as: **gen-ppl** $= 2^{f(\mathcal{D},\theta)}$ and $f(\mathcal{D},\theta) = \frac{1}{\sum_{\boldsymbol{x}\in\mathcal{D}}|\hat{\boldsymbol{x}}|}\sum_{\boldsymbol{x}\in\mathcal{D}}\log_2 p_\theta(\hat{\boldsymbol{x}}|\boldsymbol{x})$. Importantly, the optimal approach should produce text which has a perplexity *close* to that of the human-written text [14]. A high gen-ppl means the generated text is very *unlikely* given the prefix, therefore being low quality. In contrastive, a low gen-ppl means the generated text has a low diversity and gets stuck in repetitive loops [14]. We use the model θ trained with $\mathcal{L}_{\text{SimCTG}}$ to measure the gen-ppl of different approaches, therefore making sure the numbers are comparable with each other.[6]

Human Evaluation

Model	Decoding Method	Coherence	Fluency	Informativeness
Agreement	-	0.51	0.64	0.70
MLE	nucleus	2.92	3.32	3.91
	contrastive	2.78	2.29	2.56
Unlikelihood	nucleus	2.59	3.02	3.58
	contrastive	2.76	2.90	3.35
SimCTG	nucleus	2.96	3.34	3.96
	contrastive	3.25★	3.57★	3.96
SimCTG-large	nucleus	3.01	3.37	**3.98**
	contrastive	**3.33**★	**3.66**★	**3.98**
Human	-	3.70	3.71	4.21

Table 2: Human evaluation results. ★ results significantly outperforms the results of nucleus sampling with different models (Sign Test with p-value < 0.05).

5점 척도 평가

(1) Coherence: Whether the generated text is semantically consistent with the prefix.
(2) Fluency: Whether the generated text is fluent and easy to understand.
(3) Informativeness: Whether the generated text is diverse and contains interesting content.

Wikitext-103 testset에서 32 길이로 200개 랜덤추출
큰 모델에서 성능 개선 확인 -> Future work: GPT3

Open-domain Dialogue Generation

Model	Method	LCCC			DailyDialog		
		Coherence	Fluency	Informativeness	Coherence	Fluency	Informativeness
Agreement	-	0.73	0.61	0.57	0.64	0.60	0.55
MLE	greedy	3.01	3.27	1.97	3.28	3.51	2.92
	beam	2.60	2.90	1.55	3.16	3.43	2.78
	nucleus	2.78	3.55	2.64	2.67	3.58	3.42
	contrastive	3.28★	3.84★	3.06★	3.27	3.41	2.82
SimCTG	greedy	3.04	3.32	2.01	3.31	3.50	2.94
	beam	2.57	2.93	1.59	3.19	3.45	2.79
	nucleus	2.84	3.58	2.72	2.75	3.59	3.39
	contrastive	**3.32**★	**3.96**★	**3.13**★	**3.73**★	**3.85**★	**3.46**
Human	-	3.42	3.76	3.20	4.11	3.98	3.74

Table 3: Human evaluation results. ★ results significantly outperforms the results of greedy search, beam search, and nucleus sampling with different models. (Sign Test with p-value < 0.05).

Benchmark and Baselines

(1) 영어: DailyDialog
(2) 중국어: LCCC

중국어가 MLE에서도 잘되는 이유는?
ㄴThis is due to the intrinsic property of Chinese language model for which the MLE objective can already yield a representation space that displays a high level of isotropy. 🤔

04
Further Analysis

Self-similarty, Contrastive Loss Margin, Decoding Latency, Case Studies, Token Similarity Matrix, ···

Token Representation Self-similarity

$$\text{self-similarity}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{x}|\times(|\boldsymbol{x}|-1)} \sum_{i=1}^{|\boldsymbol{x}|} \sum_{j=1, j\neq i}^{|\boldsymbol{x}|} \frac{h_{x_i}^{\top} h_{x_j}}{\|h_{x_i}\| \cdot \|h_{x_j}\|}, \quad (6)$$

- Self-similarity?
  Token 끼리의 sim 평균
- Output layer에서 차이가 많이 남

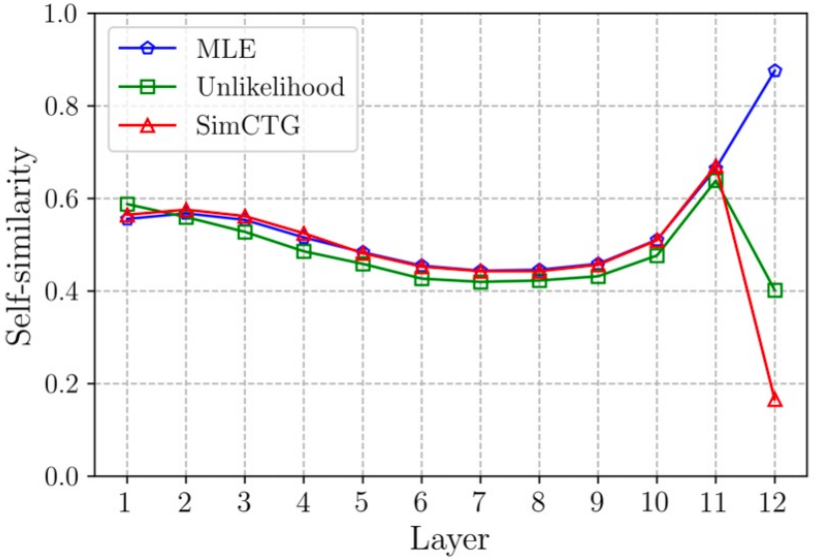

Figure 2: Layer-wise representation self-similarity.

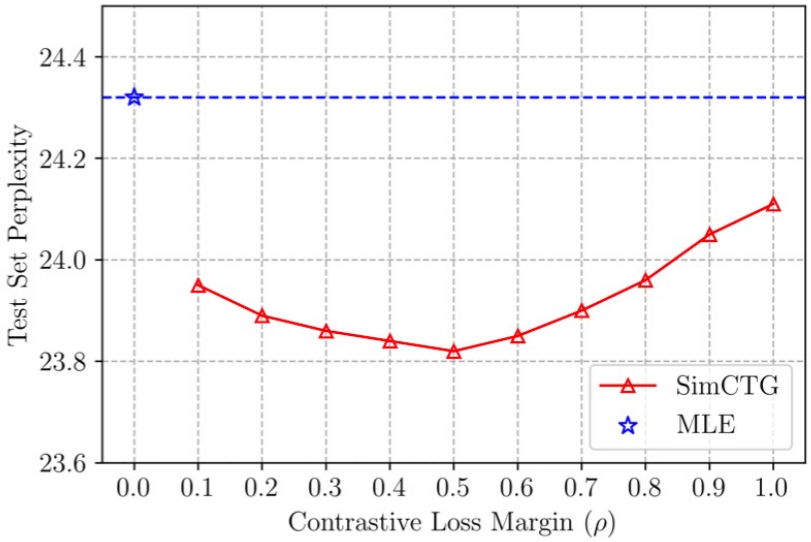

Figure 3: The effect of contrastive margin ρ.

The Effect of Contrastive Loss Margin

- contrastive loss margin ρ (Eq. (2))에 대해서 분석해보면
  perplexity on the Wikitext-103 test set기준에서는 0.5값이 가장 적당한 마진임을 알 수 있음

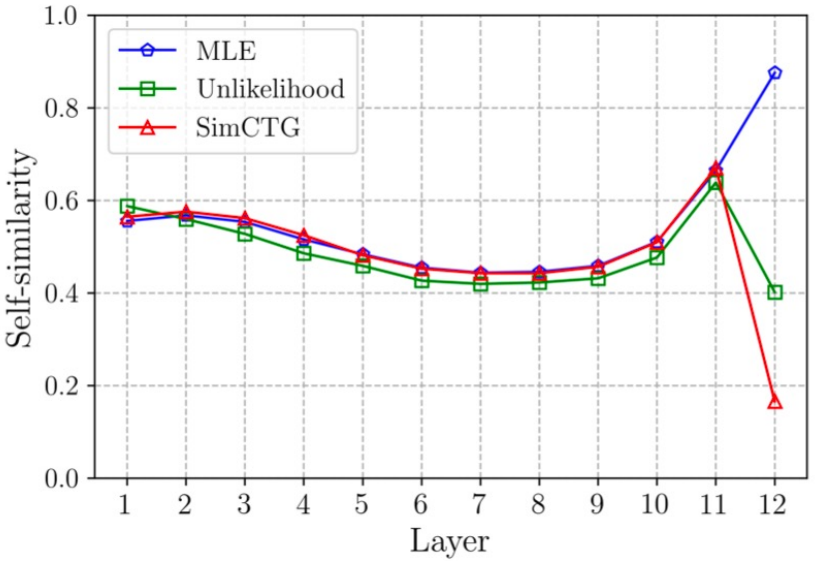

Figure 2: Layer-wise representation self-similarity.

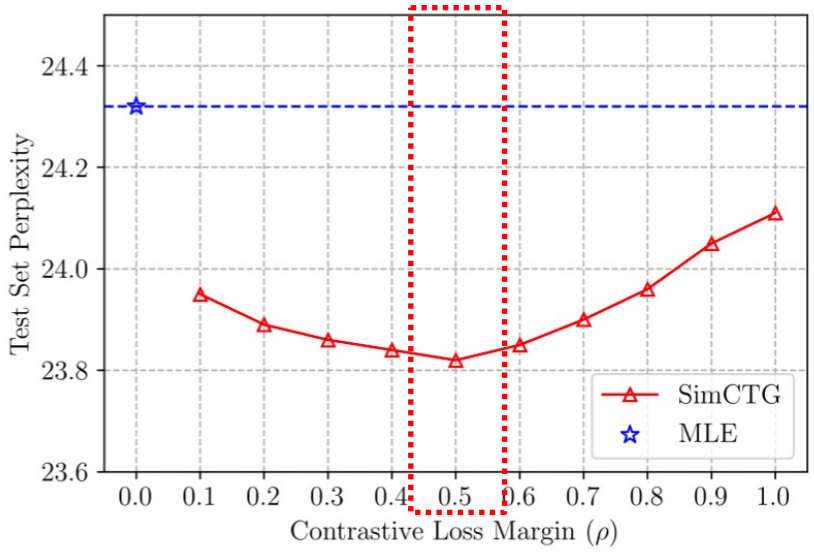

Figure 3: The effect of contrastive margin ρ.

Contrastive Search versus Nucleus Sampling
Decoding Latency Comparison

- 두가지 관점에서 분석함 (1) generation diversity (2) perplexity of the generated text (gen-ppl): 다양성은 높고 ppl은 낮고
- 생각보다 latency차이가 많이 안 남

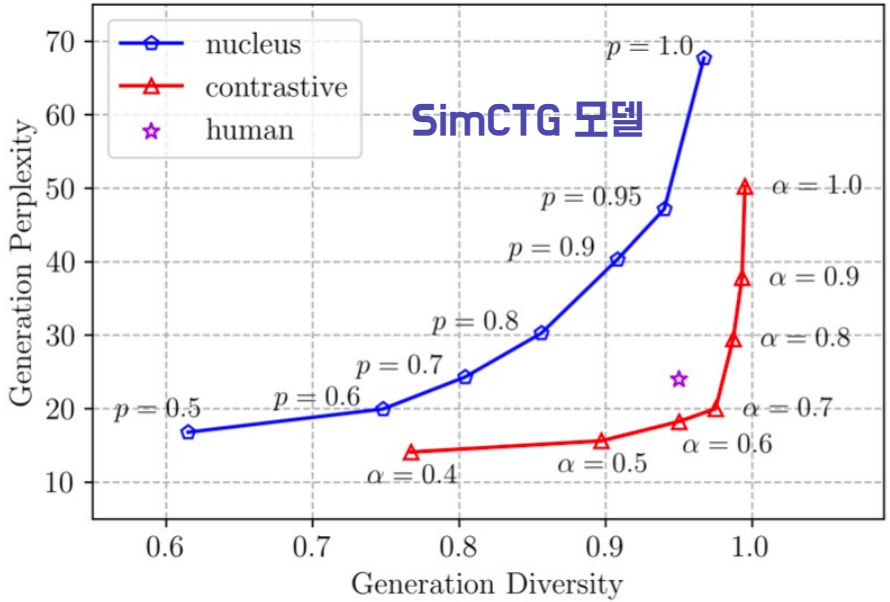

Figure 4: Contrastive search vs nucleus sampling.

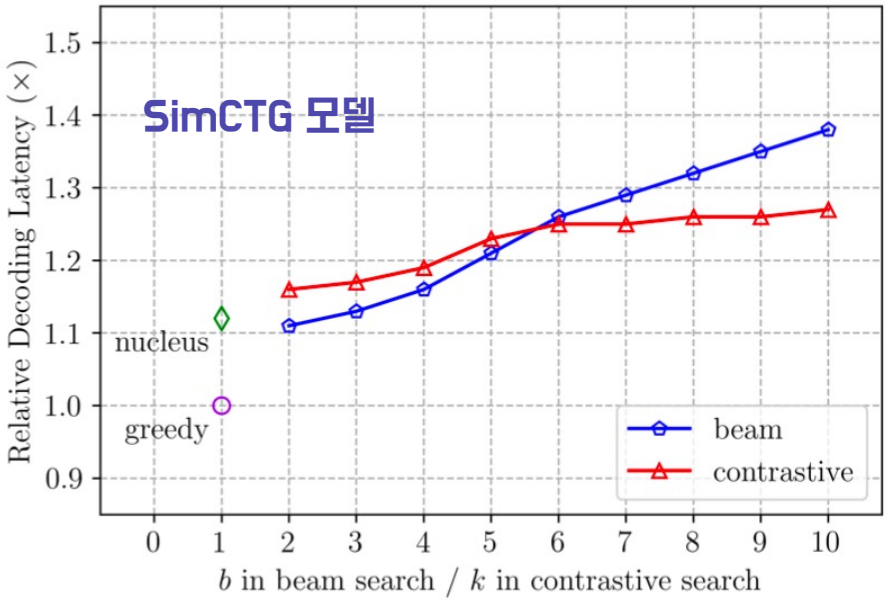

Figure 5: Inference latency comparison.

Case Study

prefix	Butt criticized Donald's controls in certain situations in the game, as well as the difficulty of some levels and puzzles. Buchanan also criticized the controls, calling	div.	coh.	gen-ppl
beam	them " too easy " and " too difficult ". He also criticized the game's graphics, stating that the game's graphics were " too simplistic " and " too simplistic ". He also criticized the game's graphics, stating that the game's graphics were " too simplistic " and " too simplistic ". He also criticized the graphics, stating that the game's graphics were " too simplistic " and " too simplistic ". He also criticized the graphics, stating that the game's graphics were " too simplistic " and " too simplistic ". He also criticized the graphics, stating that the game's graphics were" ...	0.02	0.605	7.91
nucleus	them " offensively rudimentary ". After the game was released, Buchanan wrote a number of game reviews about his review of the game, primarily summarizing what he thought the reviewers had enjoyed about it. Buchanan was pleased with how consistently these reviews awarded the game. For one, he praised the puzzles, compared them to Nos. 1 and 2 and enjoyed the construction of the artificial intelligence in the game's forms. He found no fault with the puzzles, even in those situations where they seemed repetitive. According to an Xbox Magazine review, " every time ...	0.98	0.597	37.83
contrastive	them " unimpressive " and a " nightmare " of an experience to play with players unfamiliar with Tetris. On the other hand, his opinion was shared by other reviewers, and some were critical of the game's technical design for the Wii version of Tetris. In addition, Tintin's review included a quote from Roger Ebert, who said that Tetris was better than the original game due to its simplicity and ease of play. Ebert's comments were included in the game's DVD commentary, released on March 22, 2010. It is unclear if any of the video commentary was taken from ...	0.98	0.626	19.64

Table 4: **Case Study**: The beam search produces degeneration repetitions (highlighted in red) and the nucleus sampling produces text that has incoherent semantics with respect to the prefix (highlighted in blue). The reasonable repetitions produced by contrastive search are highlighted in green. The "div." and "coh." stand for diversity and coherence metrics. (best viewed in color)

Comparison of Token Similarity Matrix

- 각 기법별 token similarity matrix보면 제안 기법의 align이 잘 되어 있는걸 볼 수 있음

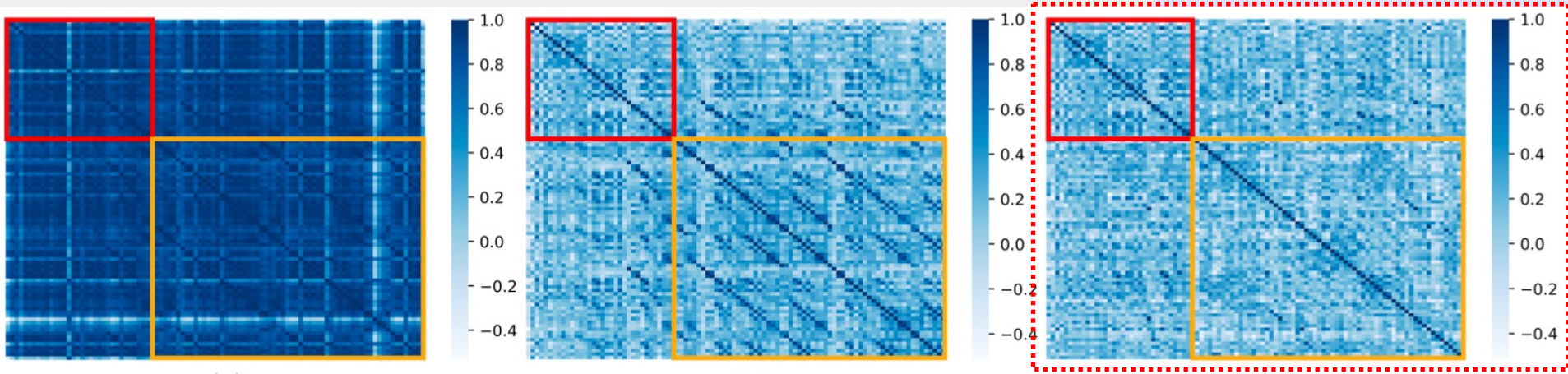

Figure 6: (a) MLE + beam search; (b) SimCTG + beam search; (c) SimCTG + contrastive search. The token similarity matrix of the prefix and the generated text are highlighted in red and yellow.

Conclusion

- Neural LM의 degeneation의 문제는 token representation의 anisotropic distribution 문제임을 보임
- SimCTG 제안함, isotropic, discriminative representation space 만들어줌
- contrastive search이라는 디코딩 방식도 제안함
- automatic and human evaluations에서 모두 가장 좋은 점수 얻고 SOTA보다 높은 점수 기록함

Appendix
Gen-ppl Results Measured by Different Models

- 다른 모델들에 대한 결과를 보면 ppl 자체는 낮은 모델들도 있지만 human-written text와 가장 유사한 건 역시 제안하는 모델
- ppl이 낮은 것 보다 사람이랑 유사한 것이 제일 좋은 것이라 주장 (논문에 있던 모델별 평가척도는 human evaluation 이었음)

F Gen-ppl Results Measured by Different Models

	greedy	beam	nucleus	contrastive	human
MLE	7.77	6.48	48.82	9.43	
Unlike.	39.02	37.38	76.22	46.03	24.86
SimCTG	8.01	6.87	47.64	**20.53**	

Table 6: The results of gen-ppl measured by the model trained with MLE.

	greedy	beam	nucleus	contrastive	human
MLE	13.18	11.67	58.01	15.94	
Unlike.	44.13	42.67	71.13	47.82	29.62
SimCTG	12.34	10.98	55.24	**23.47**	

Table 7: The results of gen-ppl measured by the model trained with Unlikelihood.

In Table 6 and 7, we show the gen-ppl (detailed in §4.1.2) results of different methods as measured by the model trained with MLE and Unlikelihood, respectively. As we use different models to measure gen-ppl, the results in Table 6 and 7 are slightly different from the ones in Table 1. Nontheless, we can draw the same conclusion as in Section §4.2 that SimCTG + contrastive search is the best performing method as it obtains the generation perplexity that is closest to the human-written text.

Thanks!