Towards a Deeper Understanding of Neural Language Generation

Tianxing He

Advisor: James Glass

2022.04.27

Outline

- Background: Neural Language Generation

- Part A: How to do generation ?

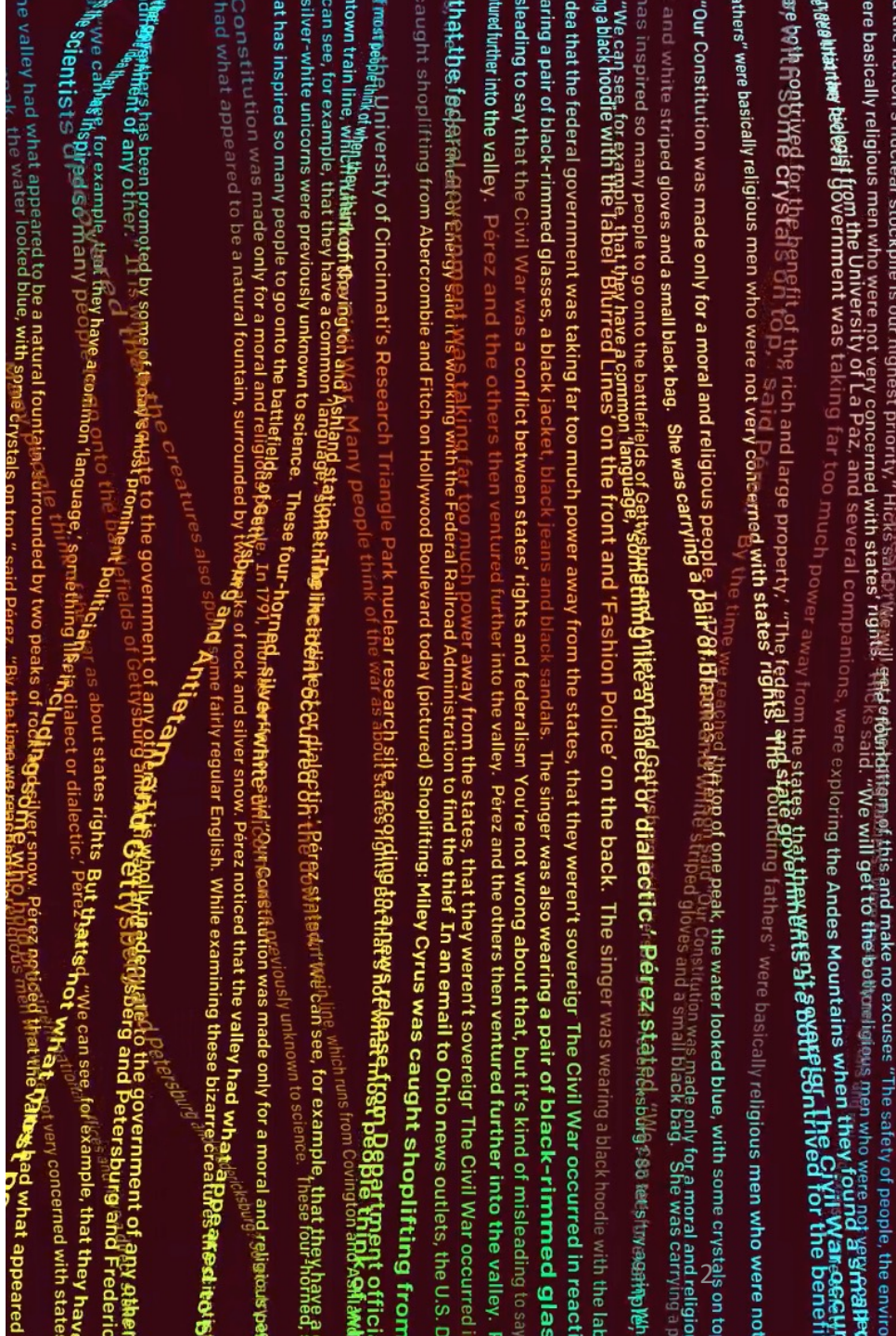
-> Sampling Algorithms

- Part B: What could be generated ?

-> Fixing Undesirable Generation Behaviors

- Closing Statements & Questions

Figure: From openai gpt-2 blog ->



Basic: Auto-regressive Language Model

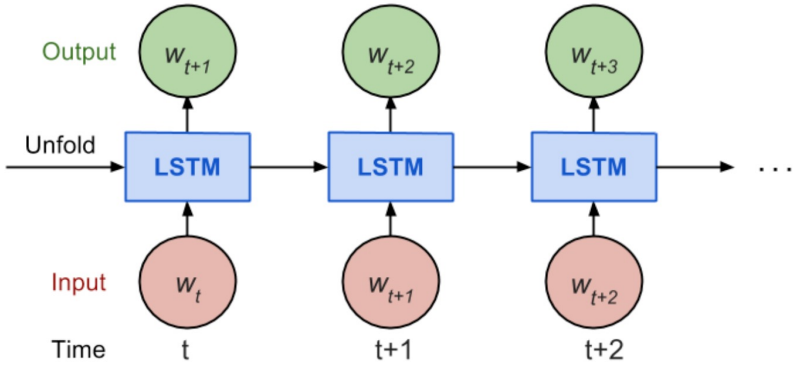
- LM assigns a probability $\mathrm{P}_\theta(W_{1:L})$ to a given sentence $W_{1:L}$
- Auto-regressive LMs predict the next token W_i given history $W_{1:i-1}$.

$$\log P_\theta(W) = \sum \log P_\theta(W_i|W_{1:i-1})$$

- It is usually trained by maximum likelihood estimation.

$$L_{MLE} = E_{W\sim P_{data}}[-\log P_\theta(W)] = KL(P_{data}||P_\theta) + constant$$

- Modeling: Recurrent Neural Network / LSTM / Transformer



Our focus today is NOT about BERT, which is a masked language model.

LMs are Exciting

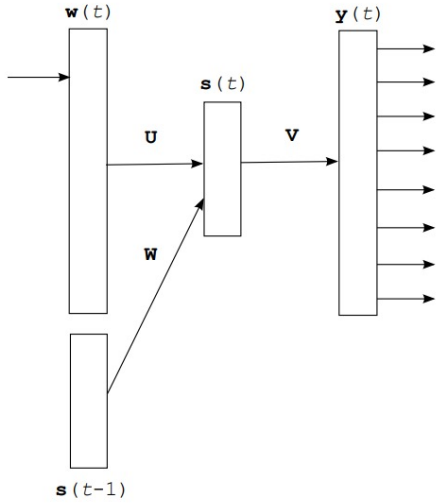
- Thanks to large scale pretraining, we now have large LMs that can generate realistic text.

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

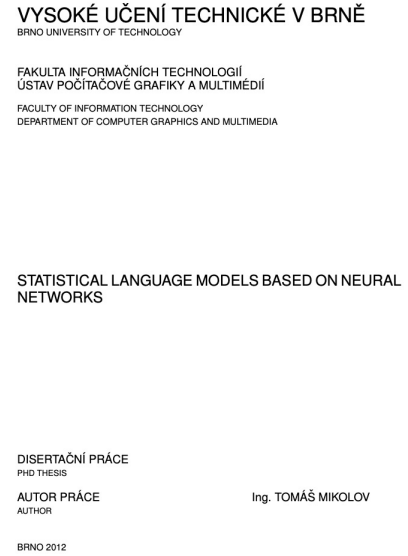
A sample from GPT2 (with top-k sampling)

Past vs. Present

Comparing GPT2/3 to the RNNLM 10 years ago…
What changed? What did not change?



RNNLM by Tomas Mikolov



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

STATISTICAL LANGUAGE MODELS BASED ON NEURAL NETWORKS

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE
AUTHOR

Ing. TOMÁŠ MIKOLOV

BRNO 2012

The first PhD thesis I read!

WHAT DID NOT CHANGE

Still autoregressive
Still using MLE(teacher forcing)

WHAT CHANGED

Large-scale data & GPU!
The mighty transformer model!
The ADAM optimizer!
HuggingFace (code repo)!
Sampling Algorithms!

Our focus today!

Outline

- Background: Neural Language Generation

- -> Part A: How to do generation?

A Systematic Characterization of Sampling Algorithms for Open-ended Language Generation, AACL 2020

Moin Nadeem, **Tianxing He***, Kyunghyun Cho, James Glass*

- Part B: What could be generated?

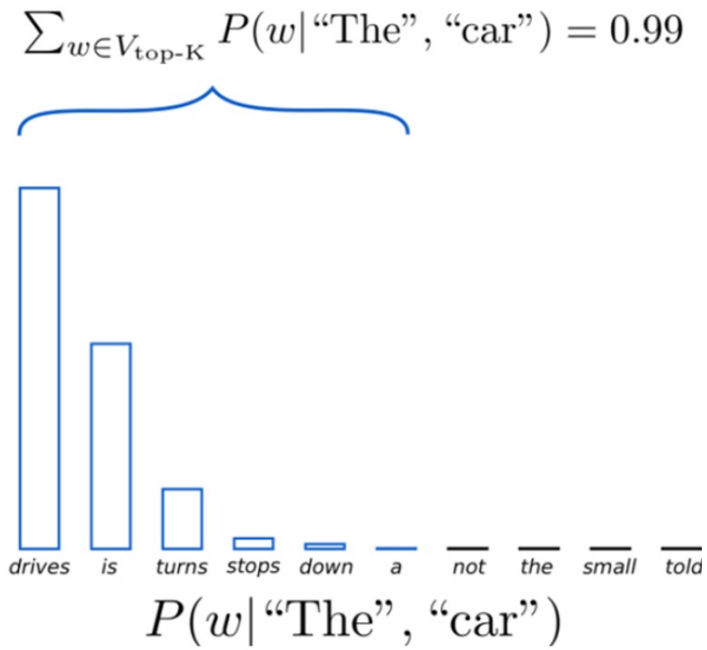
- Closing Statements

Intuition Behind the Top-K Sampling

We will represent $P(\cdot\,|W_{1..i})$ by $p = (p_1, p_2, \ldots, p_{|V|})$, where the elements is sorted that $p_1 \geq p_2 \geq p_3 \ldots \geq p_{|V|}$.
Top-K sampling transforms p to $\hat{p}$ by:

$$\widehat{p_i} = \frac{p_i \cdot 1\{i \leq K\}}{Z}$$

And we sample W_{i+1} from $\hat{p}$.



<- from https://huggingface.co/blog/how-to-generate

Sampling Algorithms are Important

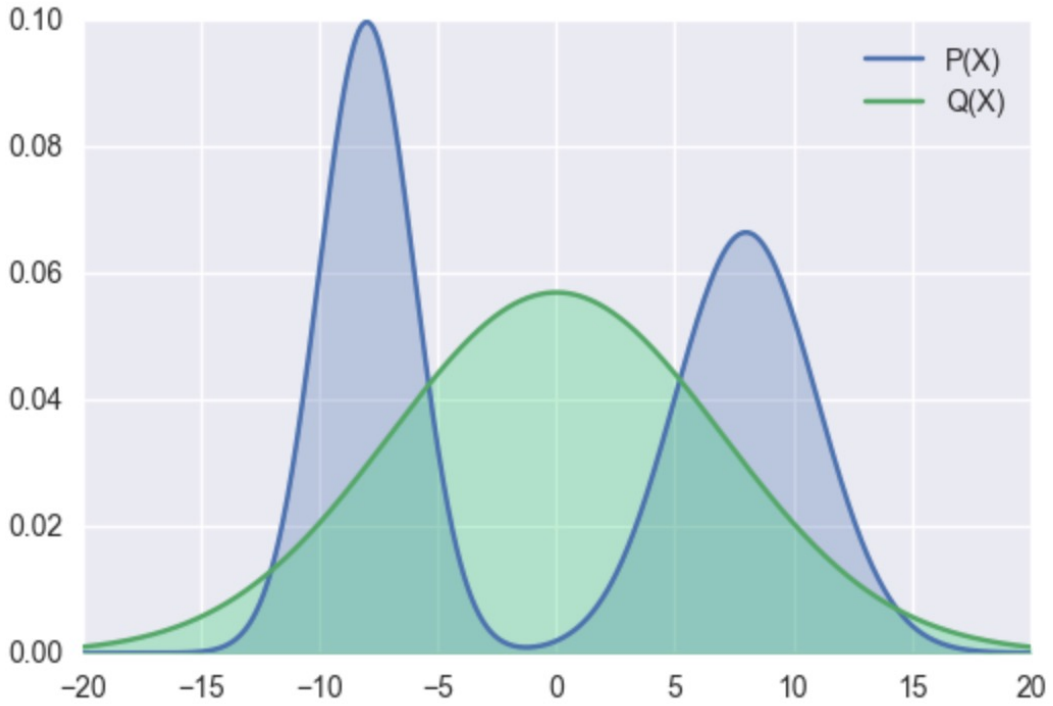
- Prompt: *MIT is a private research university in Cambridge, Massachusetts. It is one of the best universities in the U.S.,*

- GPT2 with naive sampling: but the teaching of traditional African-American studies and African-American literacy continued. Soon thereafter, MIT was renamed The International Comparative University by Lord (then), …
- GPT2 with topk40 sampling: and the home of most of the top international universities in the world. Our alumni are internationally renown, but our mission is unique. We are the only university in the world where there is a chance to take on the challenge of making an impact, …
- topk40 another sample: with a reputation for innovation and open and flexible public systems. Its principal research area deals with autonomous vehicles, robotics and artificial intelligence. To date, MIT has published 40 peer-reviewed papers on this topic, …

- Message: sampling algorithms provide a sweet quality-diversity trade-off.
- (which is the key difference to decoding e.g., beam-search)
- I did not do cherry-pick.

One Explanation of Why Sampling Algorithms are Needed

- The MLE objective can be written the forward-KL $KL(P_{data}||Q_{model})$.
- It emphasizes “diversity” but not “quality”.



<- Figure from
Agustinus Kristia’s blog

Three Popular Sampling Algorithms

- Top-K: $\widehat{p_i} = \frac{p_i \cdot 1\{i \leq K\}}{Z}$

- Nucleus (Top-P): $\widehat{p_i} = \frac{p_i \cdot 1\{\sum_{j=1}^{i-1} p_j < P\}}{Z}$

e.g., P=0.9 -> (0.5, 0.2, 0.2, ~~0.05, 0.05~~)

- Tempered (T): $\widehat{p_i} = \frac{\exp(\log(p_i)/T)}{Z}$

What we typically see in papers…

One paper

> All non-Shakespeare methods use top-k sampling with $k = 10$.

Another paper

> related sentences. To make plausible sentences, we use a back-translation using a nucleus(top-p) sampling. We used the nucleus sampling instead of traditional beam search because the former can bet-

Challenges

<- they have different hyper-parameters!

<- Existing NLG metrics either focus on quality (BLEU/METEOR, etc.) or diversity (self-BLEU, ngram-entropy, etc.)

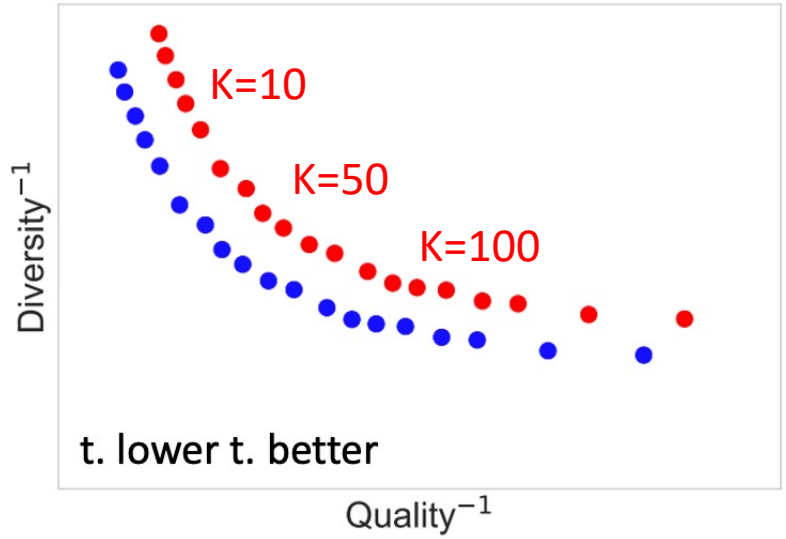
Q: Which one is the best?

Let’s do a serious comparison!

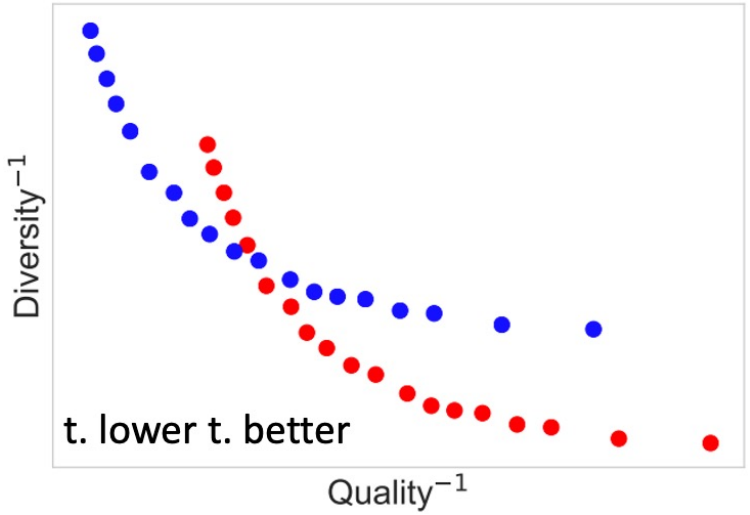
The Quality-Diversity Trade-off!

- We should tune the hyper-parameters in a big range, and for each config, we plot the (Quality-score, Diversity-score) in a figure.
- Then we get a global picture of the performance.

Illustration example:
(not real results)



blue is dominantly better.



blue has better Q, while red has better D.

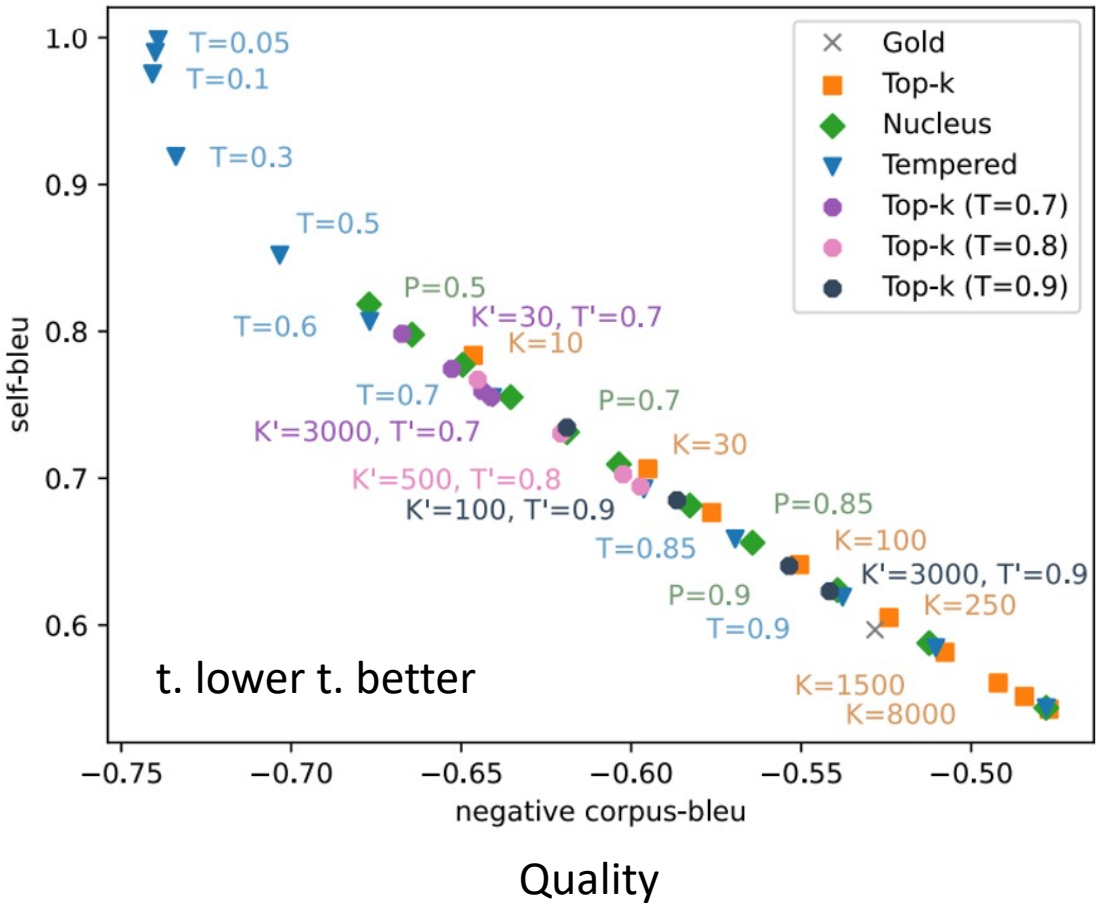
Language GANs Falling Short
Caccia et al., 2018

A Comparison of Existing Algorithms

- Setup: We finetune the GPT-2 model on the Gigaword dataset.
- Hmm… They look the same ?

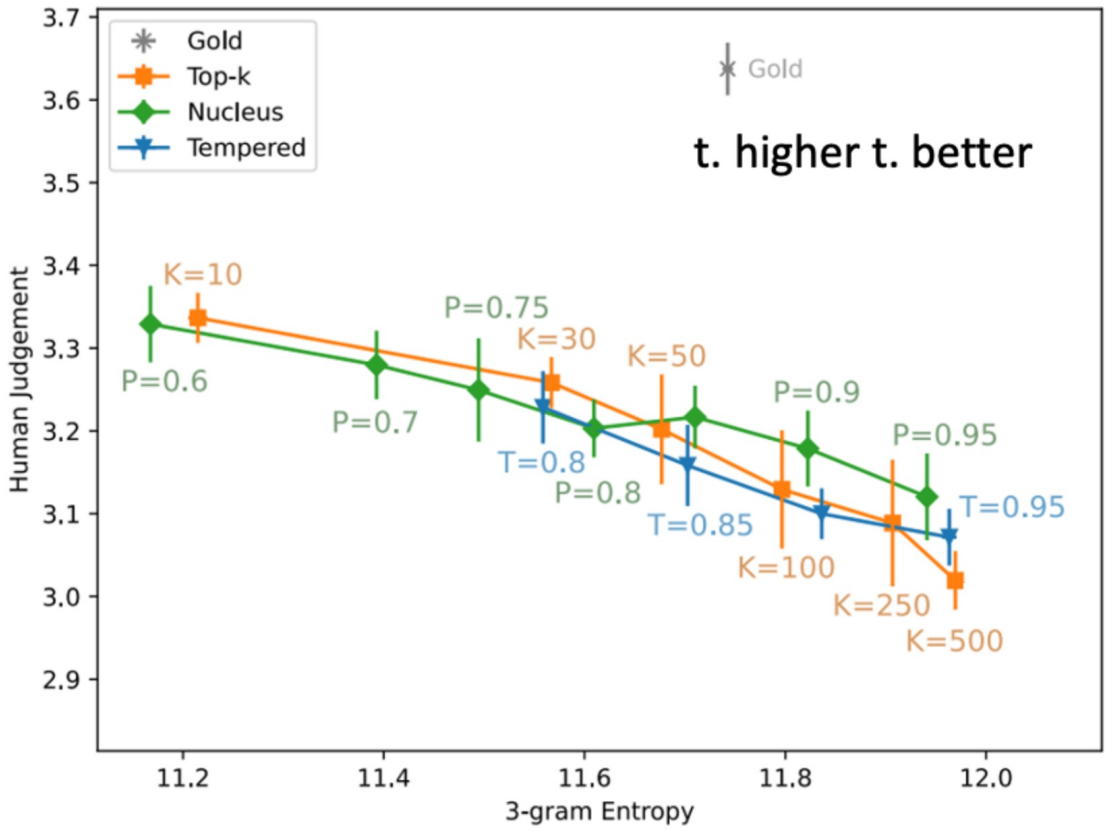
(On the same curve, no big gap)

- Also… The combination of top-k and tempered sampling does not yield a better or worse algorithm.



A Comparison of Existing Algorithms Human Evaluation

- Setup: We finetune the GPT-2 model on the Gigaword dataset.
- Hmm… They look the same ?



Quality

Diversity

Samples From Existing Algorithms

Sampling	Conditional Samples
	Existing Sampling Algorithms
Top-k *(K = 30)*	*steven spielbergs dreamworks movie studio* said monday it was filing a lawsuit, accusing us studio executives of defrauding hundreds of thousands of dollars in refunds and other damages.
Nucleus *(P = 0.80)*	*steven spielberg's dreamworks movie studio* has failed to attract the kind of business and development investors that jeffrey hutchinson dreamed up in the past.
Tempered *(T = 0.85)*	*steven spielberg's dreamworks movie studio* plans to spend the rest of the year producing the high-speed thriller "the earth's path" and an upcoming sequel, the studio announced on wednesday.

- These results lead us to question: Do these algorithms **have something in common**?

What do They Have in Common?

- Top-K: $\widehat{p_i} = \frac{p_i \cdot 1\{i \leq K\}}{Z}$
- (Nucleus) Top-P: $\widehat{p_i} = \frac{p_i \cdot 1\{\sum_{j=1}^{i-1} p_j < P\}}{Z}$
- Tempered-T: $\widehat{p_i} = \frac{\exp(\log(p_i)/T)}{Z}$

- The order of elements are preserved:

$$p_i \geq p_j \rightarrow \widehat{p_i} \geq \widehat{p_j}$$

- The entropy of the distribution are reduced:

$$\mathcal{H}(\hat{p}) \leq \mathcal{H}(p)$$

- The slope of the non-zero elements are preserved:

$$\frac{\log p_i - \log p_j}{\log p_j - \log p_k} = \frac{\log \widehat{p_i} - \log \widehat{p_j}}{\log \widehat{p_j} - \log \widehat{p_k}}, \text{ if } \widehat{p_i}, \widehat{p_j}, \widehat{p_k} > 0.$$

What do They Have in Common?

- Top-k (K): $\widehat{p_i} = \frac{p_i \cdot 1\{i \leq K\}}{Z}$
- Nucleus (top-P): $\widehat{p_i} = \frac{p_i \cdot 1\{\sum_{j=1}^{i-1} p_j < P\}}{Z}$
- Tempered (T): $\widehat{p_i} = \frac{\exp(\log(p_i)/T)}{Z}$

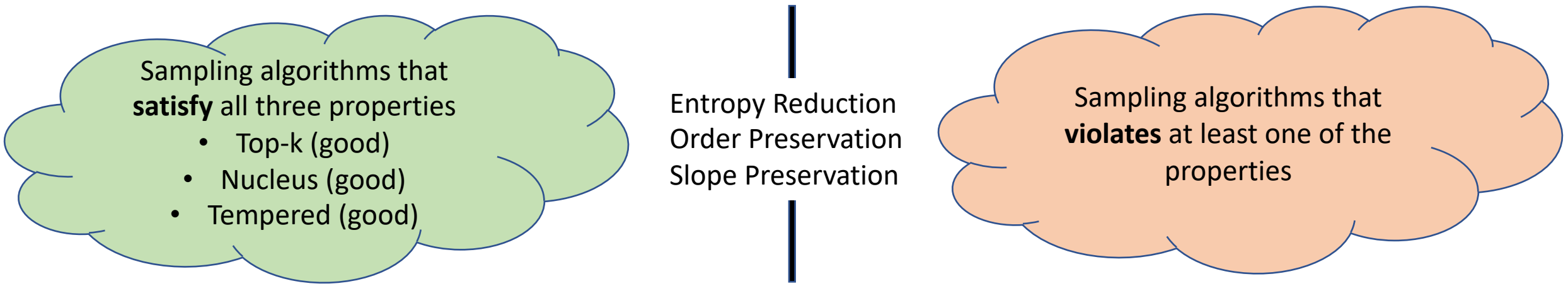
- The entropy of the distribution are reduced:
$$\mathcal{H}(\hat{p}) \leq \mathcal{H}(p)$$
- The order of elements are preserved:
$$p_i \geq p_j \rightarrow \widehat{p_i} \geq \widehat{p_j}$$
- The slope of the distribution are preserved:
$$\frac{\log p_i - \log p_j}{\log p_j - \log p_k} = \frac{\log \widehat{p_i} - \log \widehat{p_j}}{\log \widehat{p_j} - \log \widehat{p_k}}$$

***Proposition*: Entropy reduction, order preservation and slope preservation strictly hold for the transformations defined by Top-k, nucleus and tempered sampling.**
Proof: In the paper.

Are Those Properties Important?

- KEY QUESTION for the rest of Part I:
- **Are those properties important?**

- Our guess (based on the current observations):
- **Yes!**

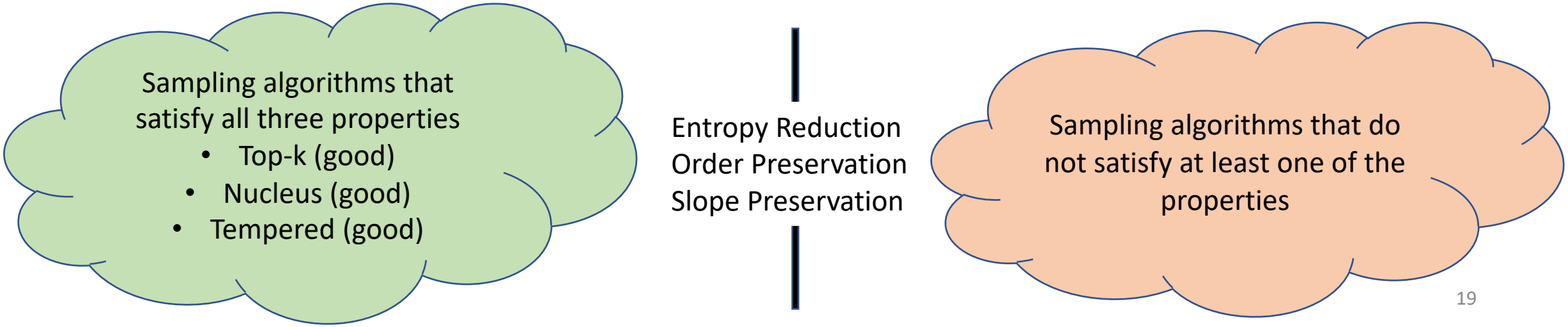


Let's Make Our Guess a Bit More Detailed

- We boldly *hypothesize* that:

- (They are **necessary**): Sampling algorithms that violate at least one of the properties won't be as good.

- (They are **sufficient**): Sampling algorithms that satisfy all three properties should be at least as good as the top-k/nucleus/tempered sampling in the Q-D trade-off.



Necessity: Property-violating Algorithms

Target-entropy Sampling (E):

$\widehat{p_i} = \frac{\exp(\log p_i / t)}{Z}$, where t is selected so that $\mathcal{H}(\hat{p}) = E$.

(it violates ***entropy reduction***)

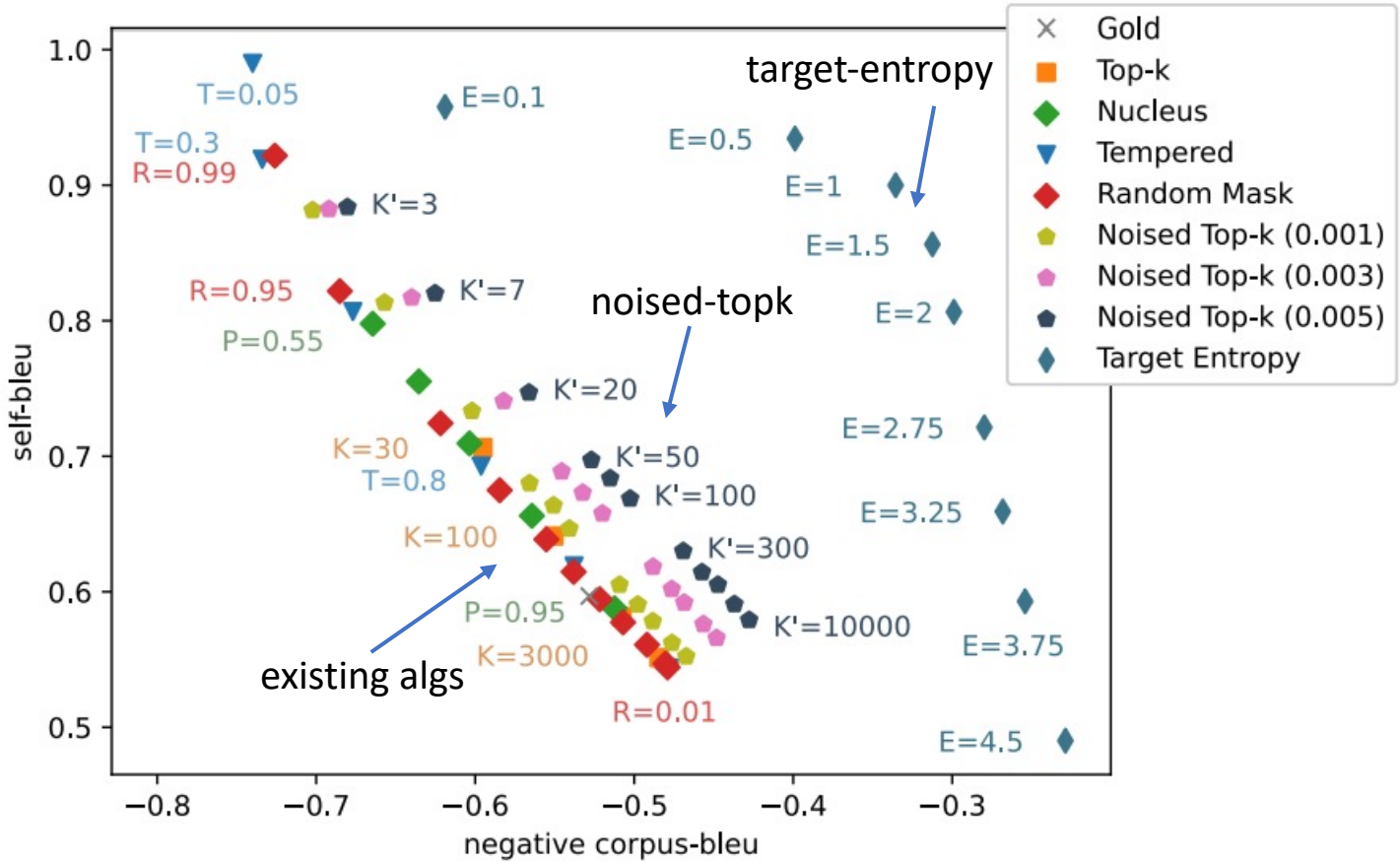
Noised Top-k Sampling (K,W):

$\hat{p} = (1 - W)\hat{p}^{\text{top-K}} + Wp^{\text{noise-K}}$,

where $p^{\text{noise-K}}$ is a sorted K-simplex.

e.g., (0.5, 0.3, 0.15, 0.05) is a sorted 4-simplex.

(it violates ***slope preservation***)



Samples of Property-violating Algorithms

Property-violating Sampling Algorithms	
Noised Top-k (K=50, W=5e-3)	*steven spielberg's dreamworks movie studio* is in disarray and has a few directors and a lot of stock involved, leaving it only a matter of time before spielberg's departure from the nobel peace prize.
Target Entropy (E = 2.75)	*steven spielberg's dreamworks movie studio* production scored an action boost m boom, nabbing an 'd after the ##th instal specialization with nominations of fritz, ika, ivan english ape and evlyn mcready.

Thought: It maybe not that surprising that those new algorithms are bad….
Can those properties guide us to propose good algorithms? (The sufficiency aspect!)

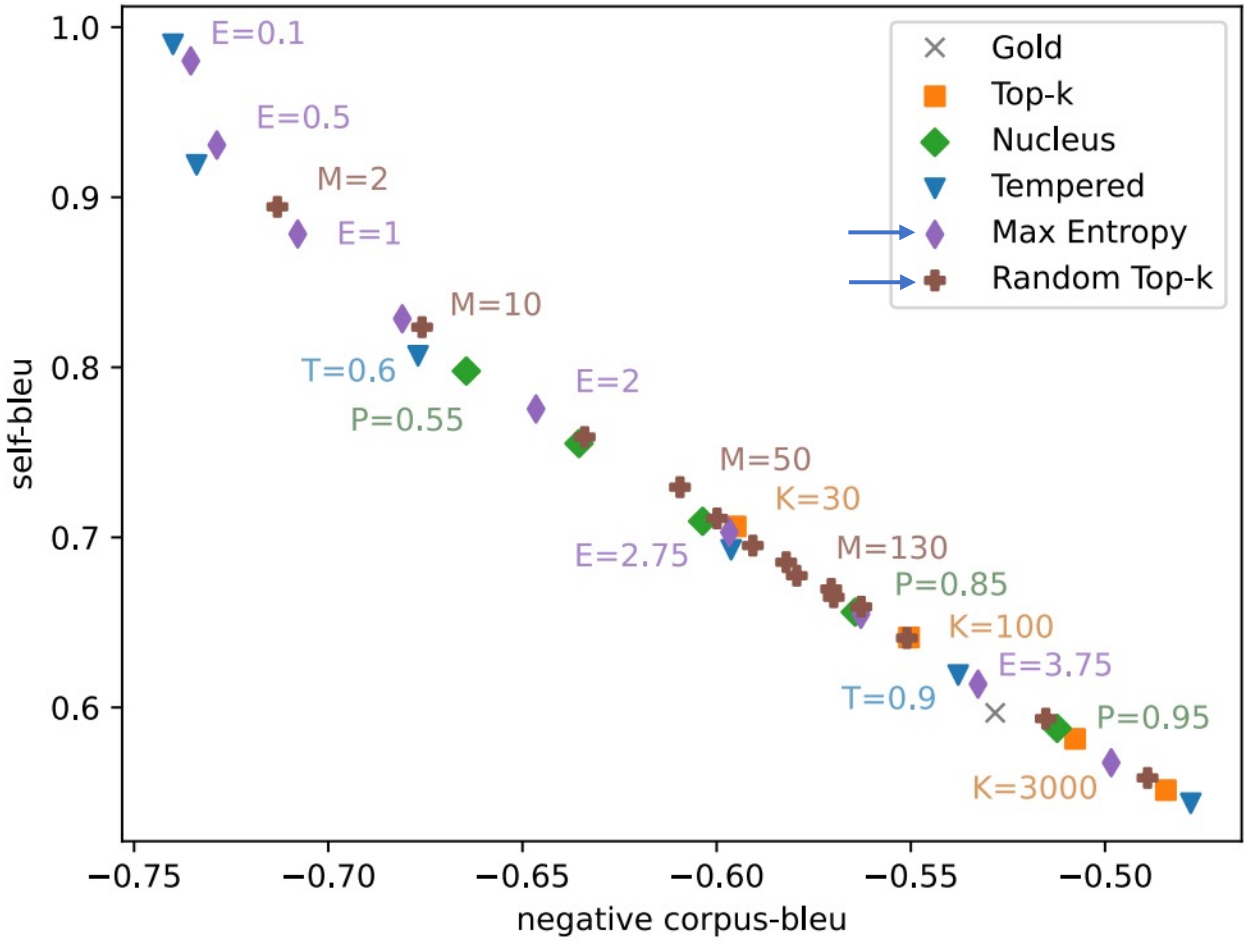
Sufficiency: Property-satisfying Algorithms

Max-entropy Sampling (E):

$\widehat{p_i} = \frac{\exp(\log p_i / t)}{Z}$, where t is selected so that $\mathcal{H}(\hat{p}) = E$, only when $\mathcal{H}(p) > E$, otherwise $\widehat{p_i} = p_i$.

Random Top-k Sampling (M):

$\widehat{p_i} = \frac{p_i \cdot 1\{i \leq k\}}{Z}$, where k is a random integer in $[1, M]$.



Human evaluation is in the paper.

Samples of Property-satisfying Algorithms

Property-satisfying Sampling Algorithms	
Random Top-k (R = 90)	*steven spielbergs dreamworks movie studio* is planning to make a movie about a young man who is a <unk>, a man who has a dream of being the first man to be born with the ability to walk on water.
Max Entropy (E = 2.75)	*steven spielberg's dreamworks movie studio* has agreed to pay $ #.# million to director john nichols (#.# million, ###, a record in the studio circulation), the studio announced sunday..

No cherry picking here.

Take-away

- (1) Sampling algorithms are trading between quality and diversity.
- (2) It seems that, what matters is not the details of how the algorithm is designed, but the high-level principles (properties) on which it is based on.
- (3) **Limitation**: It's totally possible that there is some key property, which is yet to be discovered, that can lead to dominantly better performance!

Outline

- Background: Neural Language Generation
- Part A: How to do generation?
- -> Part B: What could be generated?

Negative Training for Neural Dialogue Response Generation, ACL 2020

***Tianxing He**, James Glass*

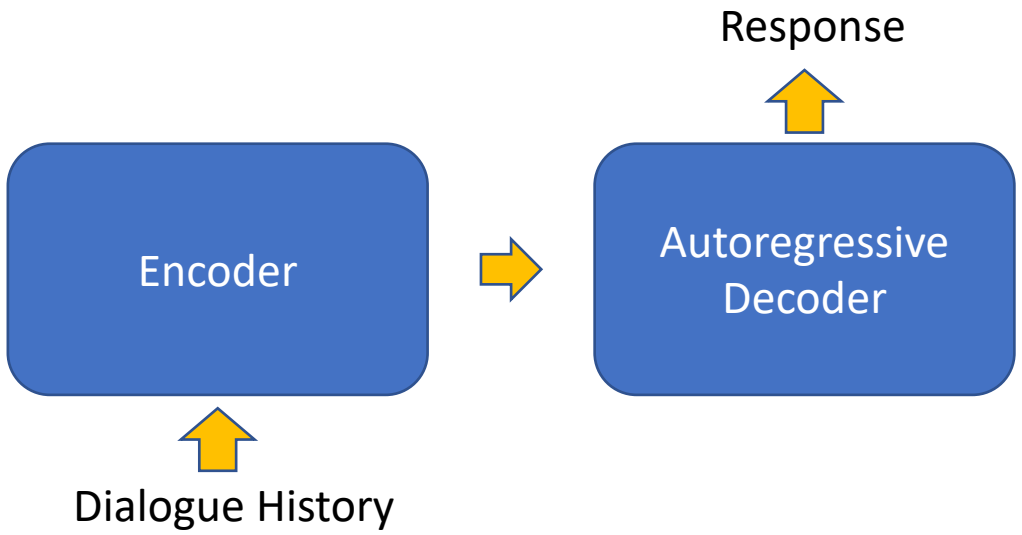
- Closing Statements

Task Background: Open-ended Dialogue Response Generation

Nature of the task: Two agents chatting in a casual manner.

A: what movies have you seen lately
B: lately i ’ve seen soap dish
A: oh
B: which was a
A: that was a lot of fun

An Example from the Switchboard dialogue dataset



Baseline model also trained by MLE.

Outline of Part B

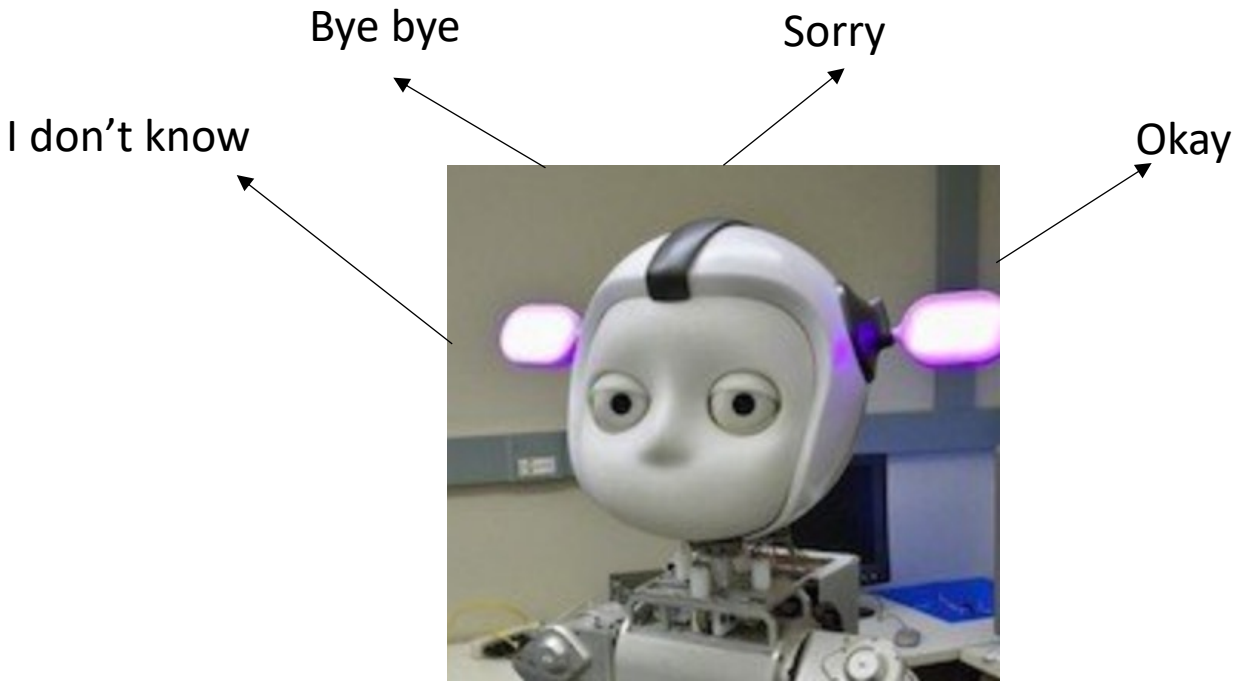
MLE is Good…. But it is not Perfect.

- We'll first briefly review two problems for MLE-trained dialogue models.
  - The generic response problem.
  - The egregious response problem.

- We'll propose NEGATIVE TRAINING to deal with those undesirable behaviors.

MLE is Good…. But Not Perfect.
The Generic Response Problem

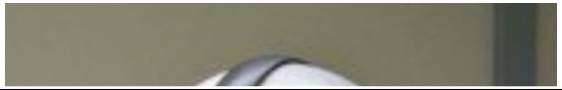
- As decoding outputs, MLE-trained dialogue response generator likes to repeat boring responses (Li et al., 2016).



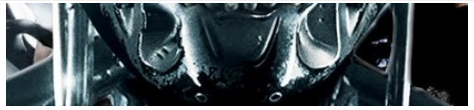
A Diversity-Promoting Objective Function for Neural Conversation Models
Li et al., 2016

MLE is Good…. But Not Perfect.
The Egregious Response Problem

- By a discrete-space adversarial attack algorithm, the model can be triggered to emit malicious or totally unacceptable responses.




Trigger Input	Decoding Output
how you woltz # sorry i you ? i not why will she a	i think you ‘re a fool
you why ! # . how the the me a us ’ii me it	i ’ll kill you
in ’s the help go we ? . it get go stupid , !	shut up .

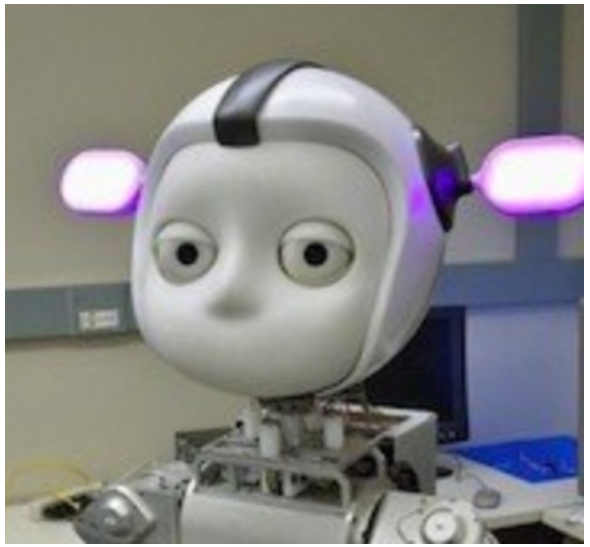



Detecting Egregious Responses in Neural Sequence-to-Sequence Models, ICLR 2019
Tianxing He, James Glass

How do We Correct the Model's Behavior?

- During MLE training, we feed the model positive training examples.
- We only tell the model "what you can say"….

We propose NEGATIVE TRAINING, to teach the model "what not to say"!



Okay

I don't know

Don't say that!

Negative Training: Derivations

- Let $\mathrm{c}(\mathrm{x}, \mathrm{y})$ be the indicator of bad behavior, negative training aims to minimize the *risk* of bad behaviors (P_{test} : testing env):

$$\mathcal{L}_{\text{NEG}}(P_{test}; \theta) = E_{\boldsymbol{x} \sim P_{test}} E_{\boldsymbol{y} \sim P_\theta(\boldsymbol{y}|\boldsymbol{x})} c(\boldsymbol{x}, \boldsymbol{y})$$

- Let's take gradient, and use the log-derivative trick:

$$\nabla_\theta \mathcal{L}_{\text{NEG}}(P_{test}; \theta) = E_{\boldsymbol{x} \sim P_{test}} E_{\boldsymbol{y} \sim P_\theta(\boldsymbol{y}|\boldsymbol{x})} c(\boldsymbol{x}, \boldsymbol{y}) \cdot \nabla_\theta \log P_\theta(\boldsymbol{y}|\boldsymbol{x})$$

<- Comparing the standard MLE training, the gradient is negated.

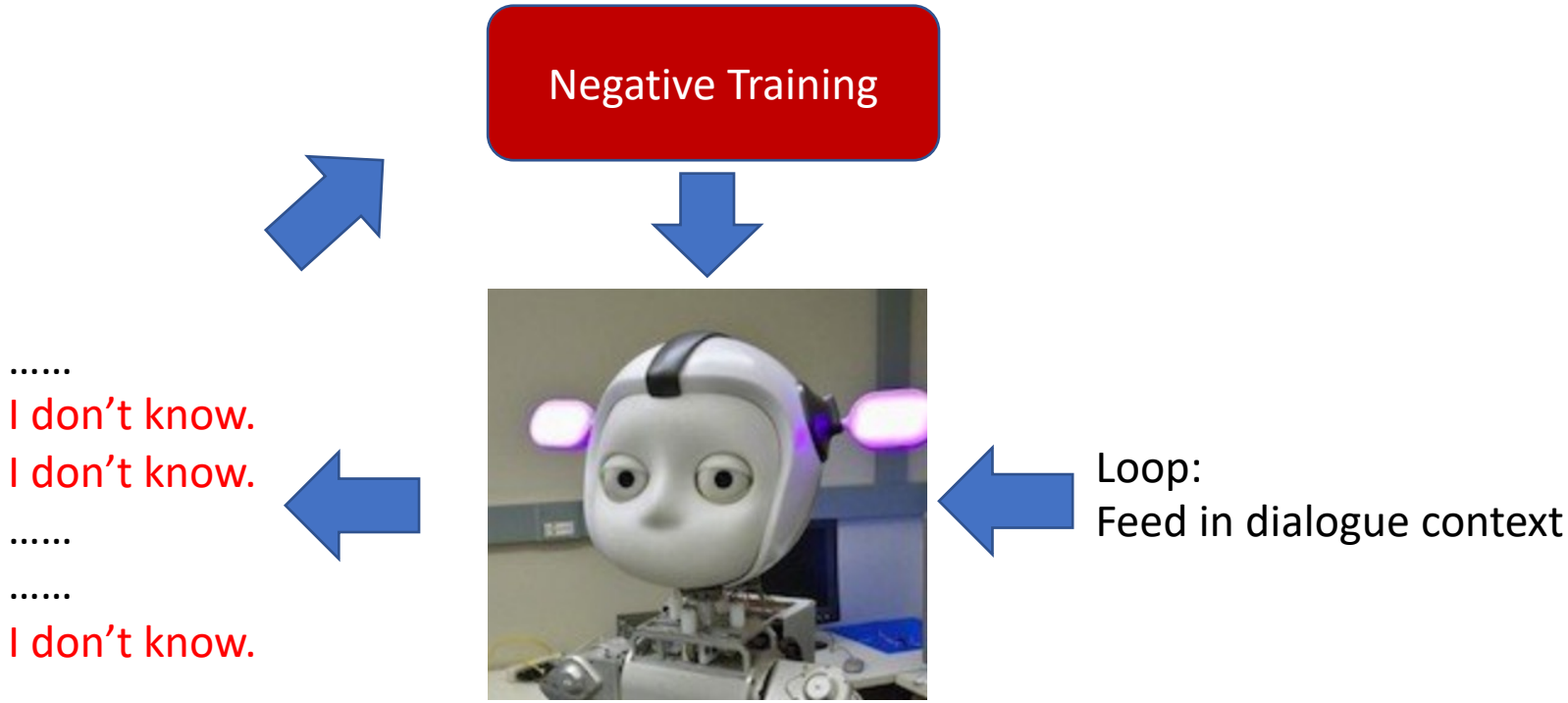
$\nabla_\theta L_{MLE} = E_{data}[-\nabla_\theta \log P_\theta(y|x)]$

- Finally, the original MLE loss is still needed.

$$\mathcal{L}_{\text{NEG+POS}} = \mathcal{L}_{\text{NEG}} + \lambda_{\text{POS}} \mathcal{L}_{\text{MLE}}$$

Negative Training for Generic Responses

- Challenge: How to deem a response as “generic”?
- Effective Heuristic: We deem a response “generic” if the frequency of it is larger than a threshold.



Negative Training for Generic Responses

- Algorithm: Negative Training for Generic Responses

Input: Model parameter θ, threshold ratio r_{thres}, learning rate α, and training data set D_{train}
for $(\boldsymbol{x}_{\text{pos}}, \boldsymbol{y}_{\text{pos}})$ **in** D_{train} **do**
  Generate response $\boldsymbol{y}_{\text{sample}}$ from the model.
  Compute the frequency r_{sample} for $\boldsymbol{y}_{\text{sample}}$ in the last 200 mini-batches.
  if $r_{\text{sample}} > r_{\text{thres}}$ **then**
    Negative update:
    $\theta = \theta - \alpha \cdot \nabla_\theta log P_\theta(\boldsymbol{y}_{\text{sample}} | \boldsymbol{x}_{\text{pos}})$
    Positive update:
    $\theta = \theta + \alpha \cdot \lambda_{\text{POS}} \cdot \nabla_\theta log P_\theta(\boldsymbol{y}_{\text{pos}} | \boldsymbol{x}_{\text{pos}})$
  end if
end for

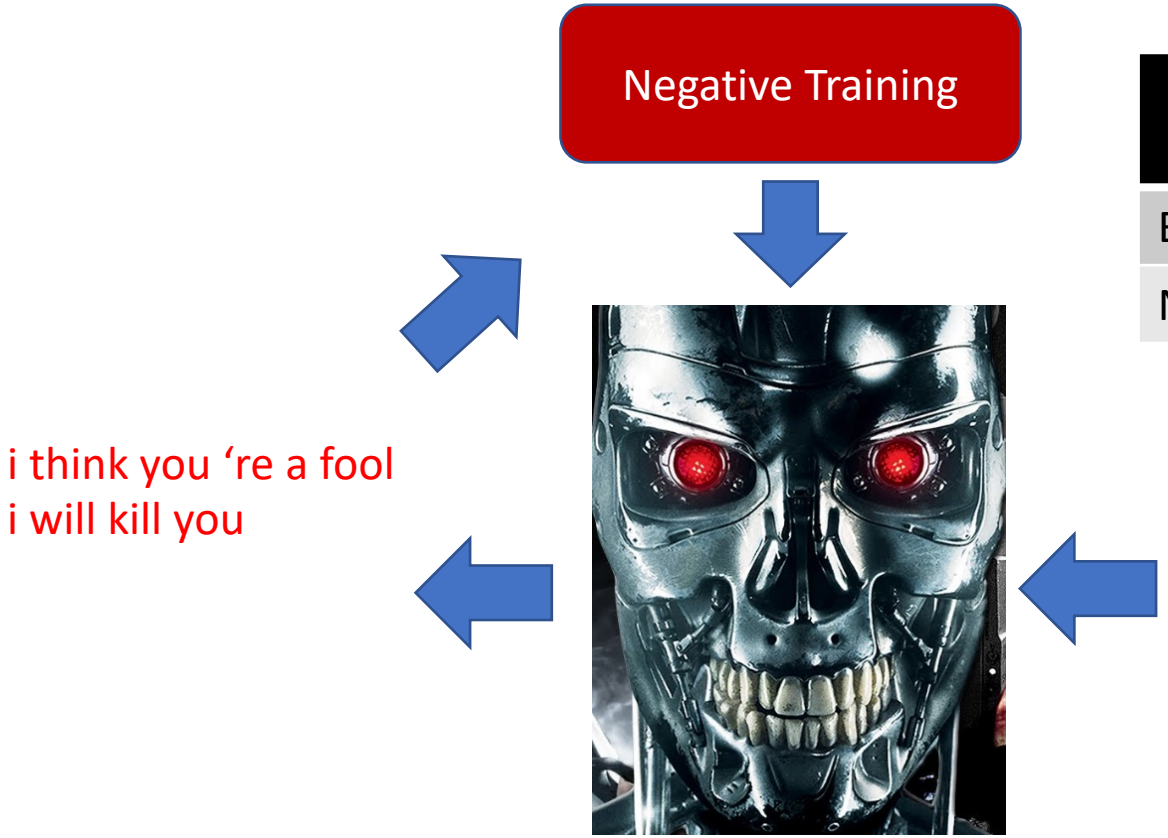
Results

- Max-ratio refers to the frequency of the most frequent response.
- E-2/3: 2/3-gram entropy
- Negative Training effectively decreases the max-ratio and increases the entropy (diversity).

Switchboard	r_{thres}	PPL	M-ratio	E-2	E-3
Test-set	N/A	N/A	10.0%	8.61	9.65
Baseline	N/A	42.81	37.4%	2.71	2.42
+GAN	N/A	42.69	49%	2.66	2.35
+MMI	N/A	N/A	23%	5.48	**6.23**
+neg-train	10%	42.84	12.4%	3.86	4.00
+neg-train	1%	44.32	**9.8%**	**5.48**	6.03
OpenSubtitles	r_{thres}	**PPL**	**M-ratio**	**E-2**	**E-3**
Test-set	N/A	N/A	0.47%	9.66	10.98
Baseline	N/A	70.81	20%	4.22	4.59
+GAN	N/A	72.00	18.8%	4.08	4.43
+MMI	N/A	N/A	3.6%	**7.63**	**9.08**
+neg-train	1%	72.37	3.1%	5.68	6.60
+neg-train	0.1%	75.71	**0.6%**	6.90	8.13

Negative Training for Egregious Responses

- In a similar fashion, we can apply negative training to teach “manners”.



	Hit Rate Train	Hit Rate Test	PPL
Baseline	27.8%	27.6%	42.81
Neg-Train	1.3%	2.6%	43.51

Results on the Switchboard dataset.

Other Applications of “Negative Training”

Independent works:

- *Neural Text Generation with Unlikelihood Training*

Welleck et al., 2020

- *Making Inconsistent Dialogue Unlikely with Unlikelihood Training*

Li et al., 2020

Take-away:

(1) Be careful about the contents of your generations!

(2) Once you spotted the samples exhibiting bad behavior, you can use them to teach the model not to do it.

Outline

- Background: Neural Language Generation
- Part A: How to do generation?
- Part B: What could be generated?
- -> Closing Statements

Towards a Better Understanding of NLG

Basics of LM

A Systematic Characterization of Sampling Algorithms for Open-ended Language Generation
[N*H*CG, 2020]

Exposure Bias versus Self-Recovery: Are Distortions Really Incremental for Autoregressive Text Generation?
[HZZG, 2021]

Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity
[ZHSJ, 2020]

Generation Behaviors

Detecting Egregious Responses in Neural Sequence-to-sequence Models
[HG, 2019]

Negative Training for Neural Dialogue Response Generation
[HG, 2020]

Knowledge in LM

Analyzing the Forgetting Problem in the Pretrain-Finetuning of Dialogue Response Models
[HLCOLGP, 2021]

An Empirical Study on Few-shot Knowledge Probing for Pretrained Language Models
[HCG, Preprint]

Controllable Generation

Controlling the Focus of Pretrained Language Generation Models
[JKGH, 2022]

Natural-Language Commands for Controllable Generation
On-going…

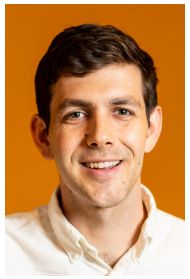
Thesis Committee

Jim

Peter

Yoon

Acknowledgements



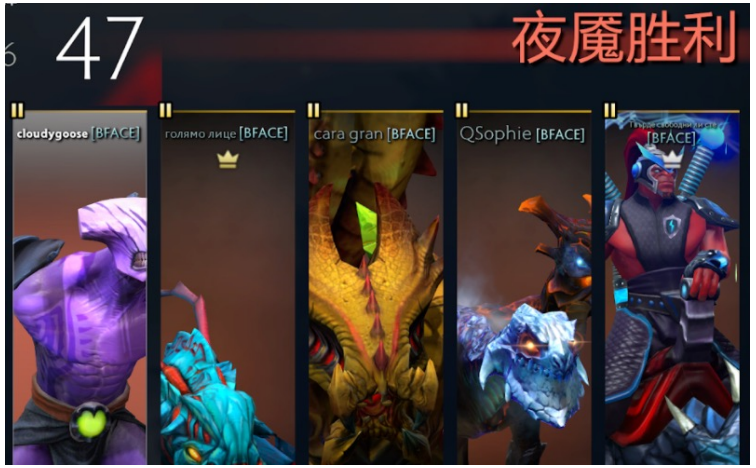

PIs

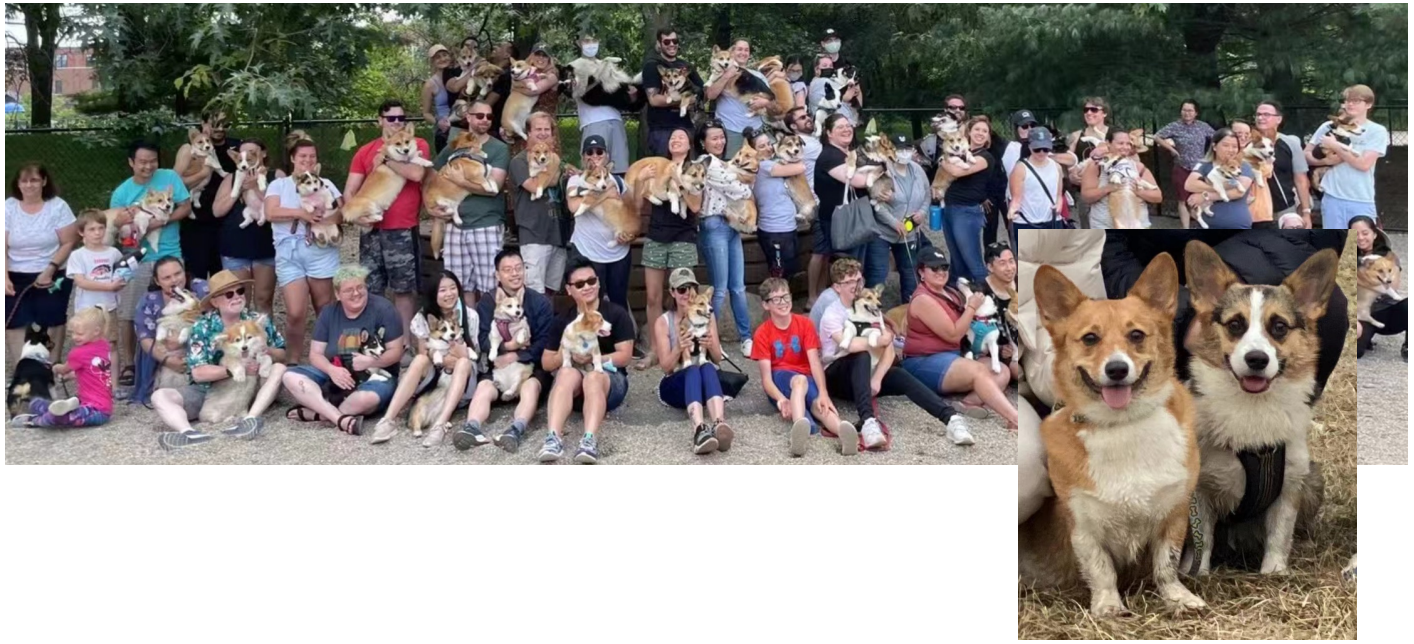
Mom & Dad

The SLS group

Lu Mi, Minnie, Mickey

Friends inside/outside MIT, human/fluffy, in-person/virtual,





Thanks & Questions?