Residual Energy-Based Models for Text Generation

Yuntian Deng[1], Anton Bakhtin[2], Myle Ott[2], Arthur Szlam[2], Marc'Aurelio Ranzato[2]

[1]Harvard University (work done while interning at FAIR)

[2]Facebook AI Research

Apr 15, 2021

Neural Text Generation

- Numerous Applications: Machine Translation, Document Summarization, News Generation, etc.
- Great Progress: GPT-2 (Radford et al., 2019a), GPT-3 (Brown et al., 2020), Switch Transformer (Fedus et al., 2021)

Neural Text Generation

- Numerous Applications: Machine Translation, Document Summarization, News Generation, etc.
- Great Progress: GPT-2 (Radford et al., 2019a), GPT-3 (Brown et al., 2020), Switch Transformer (Fedus et al., 2021)

Neural Text Generation

- Numerous Applications: Machine Translation, Document Summarization, News Generation, etc.
- Great Progress: GPT-2 (Radford et al., 2019a), GPT-3 (Brown et al., 2020), Switch Transformer (Fedus et al., 2021)

  I am having lunch with computer scientist John Schulman. He is perhaps best known for his work on reinforcement learning, or RL. In layman's terms, RL

Neural Text Generation

- Numerous Applications: Machine Translation, Document Summarization, News Generation, etc.

- Great Progress: GPT-2 (Radford et al., 2019a), GPT-3 (Brown et al., 2020), Switch Transformer (Fedus et al., 2021)

  I am having lunch with computer scientist John Schulman. He is perhaps best known for his work on reinforcement learning, or RL. In layman's terms, RL is an AI that tries to learn how to solve problems by observing the behavior of its environment and making adjustments based on the inputs received. John Schulman is a pioneer in this area, and his approach is to look at what's actually going on in an RL system and then to build up a model of how the system will actually solve the problem.

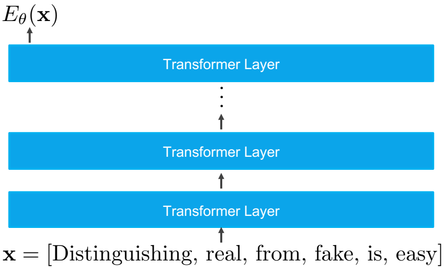
Generative Models of Text

Sentence: $\mathbf{x} = [x_1, \ldots, x_T]$

Model: $P_\phi(\mathbf{x}) = \prod_{t=1}^{T} P_\phi(x_t | x_{<t})$

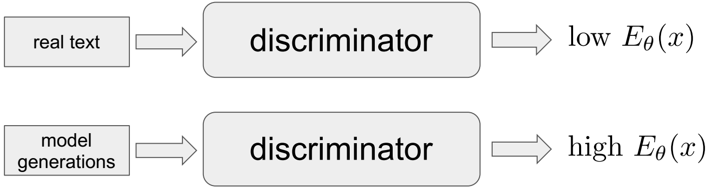
- Parameterization: uses a transformer with casual attention mask
- Training: optimizes $\max_\phi \log P_\phi(\mathbf{x})$ (MLE)
- Generation: sequentially samples x_t^* from $P_\phi(x_t | x_{<t}^*)$

Prior Work: discriminating real text from generations



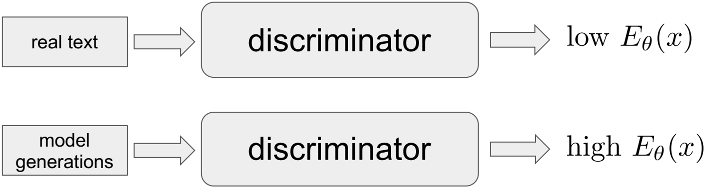
- Discriminators can reliably distinguish real text from model generations (Gehrmann et al., 2019; Bakhtin et al., 2019; Radford et al., 2019b; Zellers et al., 2019; Ippolito et al., 2019)
- Can we use such discriminators to improve text generation (this work)?

Prior Work: discriminating real text from generations



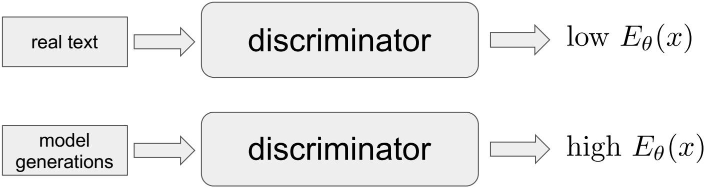
- Discriminators can reliably distinguish real text from model generations (Gehrmann et al., 2019; Bakhtin et al., 2019; Radford et al., 2019b; Zellers et al., 2019; Ippolito et al., 2019)
- Can we use such discriminators to improve text generation (this work)?

Prior Work: discriminating real text from generations



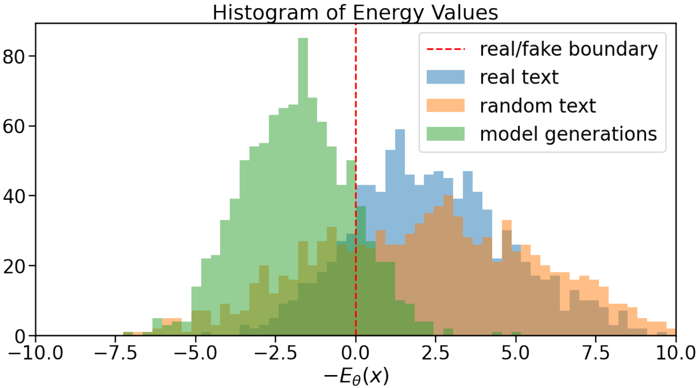
- Discriminators can reliably distinguish real text from model generations (Gehrmann et al., 2019; Bakhtin et al., 2019; Radford et al., 2019b; Zellers et al., 2019; Ippolito et al., 2019)
- Can we use such discriminators to improve text generation (this work)?

Prior Work: discriminating real text from generations



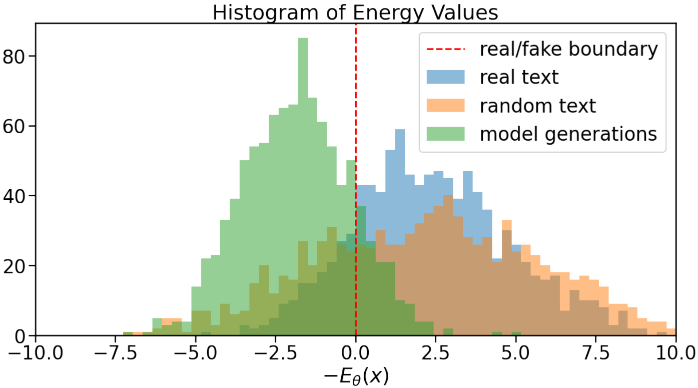
- Discriminators can reliably distinguish real text from model generations (Gehrmann et al., 2019; Bakhtin et al., 2019; Radford et al., 2019b; Zellers et al., 2019; Ippolito et al., 2019)
- Can we use such discriminators to improve text generation (this work)?

Can we simply search for text considered "real" by the discriminator?



- The discriminator was trained only on model generations and real text, it's unable to score out-of-distribution inputs (such as random text) properly

Can we simply search for text considered “real” by the discriminator?



- The discriminator was trained only on model generations and real text, it’s unable to score out-of-distribution inputs (such as random text) properly

Residual Energy-Based Models for Text

- Combine the discriminator score with the generator score to penalize out-of-distribution inputs

$$\log P_{\text{joint}}(\mathbf{x}) = -E_\theta(\mathbf{x}) + \log P_\phi(\mathbf{x}) - \log Z(\theta, \phi)^1$$

- Alternatively, can be understood as adjusting the log likelihood of the generator $\log P_\phi(\mathbf{x})$ by $-E_\theta(\mathbf{x})$

[1] The same form has been proposed in Wang and Ou (2018) for speech recognition

Residual Energy-Based Models for Text

- Combine the discriminator score with the generator score to penalize out-of-distribution inputs

$$\log P_{\text{joint}}(\mathbf{x}) = -E_\theta(\mathbf{x}) + \log P_\phi(\mathbf{x}) - \log Z(\theta, \phi)^1$$

- Alternatively, can be understood as adjusting the log likelihood of the generator $\log P_\phi(\mathbf{x})$ by $-E_\theta(\mathbf{x})$

[1] The same form has been proposed in Wang and Ou (2018) for speech recognition

Training

$$\log P_{\text{joint}}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) - \log Z(\theta, \phi)$$

- We pretrain $P_\phi(\mathbf{x})$ and only consider learning θ
- MLE is intractable: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$
- Algorithms requiring sampling from $P_{\text{joint}}(\mathbf{x})$ (such as CD-k) are computationally expensive
- Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010; Ma and Collins, 2018) provides an elegant solution

Training

$$\log P_{\text{joint}}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) - \log Z(\theta, \phi)$$

- We pretrain $P_\phi(\mathbf{x})$ and only consider learning θ
- MLE is intractable: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$
- Algorithms requiring sampling from $P_{\text{joint}}(\mathbf{x})$ (such as CD-k) are computationally expensive
- Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010; Ma and Collins, 2018) provides an elegant solution

Training

$$\log P_{\text{joint}}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) - \log Z(\theta, \phi)$$

- We pretrain $P_\phi(\mathbf{x})$ and only consider learning θ
- MLE is intractable: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$
- Algorithms requiring sampling from $P_{\text{joint}}(\mathbf{x})$ (such as CD-k) are computationally expensive
- Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010; Ma and Collins, 2018) provides an elegant solution

Training

$$\log P_{\text{joint}}(\mathbf{x}) = \log P_{\phi}(\mathbf{x}) - E_{\theta}(\mathbf{x}) - \log Z(\theta, \phi)$$

- We pretrain $P_{\phi}(\mathbf{x})$ and only consider learning θ
- MLE is intractable: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_{\phi}(\mathbf{x}) \exp(-E_{\theta}(\mathbf{x}))$
- Algorithms requiring sampling from $P_{\text{joint}}(\mathbf{x})$ (such as CD-k) are computationally expensive
- Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010; Ma and Collins, 2018) provides an elegant solution

Noise Contrastive Estimation (NCE)

Data distribution: $P_{\text{data}}(x)$

Noise distribution: $P_\phi(x)$

- The NCE objective:

$$\max_\theta \mathop{\mathbb{E}}_{\mathbf{x}_+\sim P_{\text{data}}} \log \frac{1}{1+\exp(E_\theta(\mathbf{x}_+))} + \mathop{\mathbb{E}}_{\mathbf{x}_-\sim P_\phi} \log \frac{1}{1+\exp(-E_\theta(\mathbf{x}_-))}$$

- Happens to be the same as training a discriminator
  - $\mathbf{x}_+$: $y(\mathbf{x}_+) = 1$, $\mathbf{x}_-$: $y(\mathbf{x}_-) = 0$
  - $P(y(\mathbf{x}) = 1) = \text{sigmoid}(-E_\theta(\mathbf{x})), P(y(\mathbf{x}) = 0) = 1 - P(y(\mathbf{x}) = 1)$
  - Train with MLE

Noise Contrastive Estimation (NCE)

Data distribution: $P_{\text{data}}(x)$

Noise distribution: $P_\phi(x)$

- The NCE objective:

$$\max_\theta \mathop{\mathbb{E}}_{\mathbf{x}_+ \sim P_{\text{data}}} \log \frac{1}{1+\exp(E_\theta(\mathbf{x}_+))} + \mathop{\mathbb{E}}_{\mathbf{x}_- \sim P_\phi} \log \frac{1}{1+\exp(-E_\theta(\mathbf{x}_-))}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}_+ \sim P_{\text{data}}} \log \text{sigmoid}(-E_\theta(\mathbf{x}_+)) + \mathop{\mathbb{E}}_{\mathbf{x}_- \sim P_\phi} \log(1 - \text{sigmoid}(-E_\theta(\mathbf{x}_-)))$$

- Happens to be the same as training a discriminator
  - $\mathbf{x}_+$: $y(\mathbf{x}_+) = 1$, $\mathbf{x}_-$: $y(\mathbf{x}_-) = 0$
  - $P(y(\mathbf{x}) = 1) = \text{sigmoid}(-E_\theta(\mathbf{x})), P(y(\mathbf{x}) = 0) = 1 - P(y(\mathbf{x}) = 1)$
  - Train with MLE

Noise Contrastive Estimation (NCE)

Data distribution: $P_{\text{data}}(x)$

Noise distribution: $P_\phi(x)$

- The NCE objective:

$$\max_\theta \mathop{\mathbb{E}}_{\mathbf{x}_+ \sim P_{\text{data}}} \log \frac{1}{1+\exp(E_\theta(\mathbf{x}_+))} + \mathop{\mathbb{E}}_{\mathbf{x}_- \sim P_\phi} \log \frac{1}{1+\exp(-E_\theta(\mathbf{x}_-))}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}_+ \sim P_{\text{data}}} \log \text{sigmoid}(-E_\theta(\mathbf{x}_+)) + \mathop{\mathbb{E}}_{\mathbf{x}_- \sim P_\phi} \log(1 - \text{sigmoid}(-E_\theta(\mathbf{x}_-))$$

- Happens to be the same as training a discriminator
  - $\mathbf{x}_+$: $y(\mathbf{x}_+) = 1$, $\mathbf{x}_-$: $y(\mathbf{x}_-) = 0$
  - $P(y(\mathbf{x}) = 1) = \text{sigmoid}(-E_\theta(\mathbf{x})), P(y(\mathbf{x}) = 0) = 1 - P(y(\mathbf{x}) = 1)$
  - Train with MLE

Theoretical Guarantee

Theorem (1)

If P_ϕ has the same support as P_{data}, then the NCE objective reaches its maximum at $\log P_{joint}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) = \log P_{data}$, if there exists such θ.

- At optimum, $P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x})) = P_{\text{data}}(\mathbf{x})$ is self-normalizing: $\log Z(\theta, \phi) = 0$[2].
- The optimum might not be reached due to E_θ only having finite capacity, and also due to optimization errors

[2] During evaluation, we still need to estimate $\log Z(\theta, \phi)$ since we cannot guarantee that the optimum is reached

Theoretical Guarantee

Theorem (1)

If P_ϕ has the same support as P_{data}, then the NCE objective reaches its maximum at $\log P_{joint}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) = \log P_{data}$, if there exists such θ.

- At optimum, $P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x})) = P_{\text{data}}(\mathbf{x})$ is self-normalizing: $\log Z(\theta, \phi) = 0$[2].
- The optimum might not be reached due to E_θ only having finite capacity, and also due to optimization errors

[2] During evaluation, we still need to estimate $\log Z(\theta, \phi)$ since we cannot guarantee that the optimum is reached

Theoretical Guarantee

Theorem (1)

If P_ϕ has the same support as P_{data}, then the NCE objective reaches its maximum at $\log P_{joint}(\mathbf{x}) = \log P_\phi(\mathbf{x}) - E_\theta(\mathbf{x}) = \log P_{data}$, if there exists such θ.

- At optimum, $P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x})) = P_{\text{data}}(\mathbf{x})$ is self-normalizing: $\log Z(\theta, \phi) = 0$[2].
- The optimum might not be reached due to E_θ only having finite capacity, and also due to optimization errors

[2] During evaluation, we still need to estimate $\log Z(\theta, \phi)$ since we cannot guarantee that the optimum is reached

Estimating Log Likelihood

- Generative models can be evaluated with the log likelihood of data
- Text generation usually uses perplexity (PPL, the lower the better):

$$\text{PPL} = \exp(-\frac{1}{\#\text{words}} \log P_{\text{joint}}(\mathbf{x}))$$

- Challenge: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$ is intractable
- Solution: We proposed asymptotic lower- and upper-bounds (based on Nowozin (2018))
- Those bounds also allow us to estimate $P_{\text{joint}}(x_t|x_{<t})$ through marginalization

Estimating Log Likelihood

- Generative models can be evaluated with the log likelihood of data
- Text generation usually uses perplexity (PPL, the lower the better):

$$\text{PPL} = \exp(-\frac{1}{\#\text{words}} \log P_{\text{joint}}(\mathbf{x}))$$

- Challenge: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$ is intractable
- Solution: We proposed asymptotic lower- and upper-bounds (based on Nowozin (2018))
- Those bounds also allow us to estimate $P_{\text{joint}}(x_t|x_{<t})$ through marginalization

Estimating Log Likelihood

- Generative models can be evaluated with the log likelihood of data
- Text generation usually uses perplexity (PPL, the lower the better):

$$\text{PPL} = \exp(-\frac{1}{\#\text{words}} \log P_{\text{joint}}(\mathbf{x}))$$

- Challenge: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_{\phi}(\mathbf{x}) \exp(-E_{\theta}(\mathbf{x}))$ is intractable
- Solution: We proposed asymptotic lower- and upper-bounds (based on Nowozin (2018))
- Those bounds also allow us to estimate $P_{\text{joint}}(x_t | x_{<t})$ through marginalization

Estimating Log Likelihood

- Generative models can be evaluated with the log likelihood of data
- Text generation usually uses perplexity (PPL, the lower the better):

$$\text{PPL} = \exp(-\frac{1}{\#\text{words}} \log P_{\text{joint}}(\mathbf{x}))$$

- Challenge: $Z(\theta, \phi) = \sum_{\mathbf{x}} P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$ is intractable
- Solution: We proposed asymptotic lower- and upper-bounds (based on Nowozin (2018))
- Those bounds also allow us to estimate $P_{\text{joint}}(x_t|x_{<t})$ through marginalization

Generation

$P_{\text{joint}}(\mathbf{x}) = P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))/Z$

- Challenge: P_{joint} does not allow sequential sampling of $x_t | x_{<t}$
- Our solution: sample from P_ϕ and then resample
- $n \to \infty$ recovers exact sampling from P_{joint}

Generation

$P_{\text{joint}}(\mathbf{x}) = P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))/Z$

- Challenge: P_{joint} does not allow sequential sampling of $x_t | x_{<t}$
- Our solution: sample from P_ϕ and then resample
- $n \to \infty$ recovers exact sampling from P_{joint}

Generation

$$P_{\text{joint}}(\mathbf{x}) = P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))/Z$$

- Challenge: P_{joint} does not allow sequential sampling of $x_t | x_{<t}$
- Our solution: sample from P_ϕ and then resample
- $n \to \infty$ recovers exact sampling from P_{joint}

Algorithm 1: Top-k Joint Sampling

Input: number of samples n drawn from P_ϕ, value of k in top-k

`// Get a set of samples from` P_ϕ

sample n samples $\{x^1, \cdots, x^n\}$ from P_ϕ with top-k sampling

calculate energies $s^i = E_\theta(x^i)$ for each $x^i \in \{x^1, \cdots, x^n\}$

`// Resample from the set of samples`

sample $x = x^i$ with probability $\frac{\exp(-s^i)}{\sum_{j=1}^n \exp(-s^j)}$

return x

Generation

$$P_{\text{joint}}(\mathbf{x}) = P_\phi(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))/Z$$

- Challenge: P_{joint} does not allow sequential sampling of $x_t | x_{<t}$
- Our solution: sample from P_ϕ and then resample
- $n \to \infty$ recovers exact sampling from P_{joint}

Algorithm 1: Top-k Joint Sampling

Input: number of samples n drawn from P_ϕ, value of k in top-k

`// Get a set of samples from` P_ϕ

sample n samples $\{x^1, \cdots, x^n\}$ from P_ϕ with top-k sampling

calculate energies $s^i = E_\theta(x^i)$ for each $x^i \in \{x^1, \cdots, x^n\}$

`// Resample from the set of samples`

sample $x = x^i$ with probability $\frac{\exp(-s^i)}{\sum_{j=1}^{n} \exp(-s^j)}$

return x

Experimental Setup

- Datasets
  - CC-News: 16 billion words (Nagel, 2016; Bakhtin et al., 2019)
  - Toronto Book Corpus: 0.5 billion words (Zhu et al., 2015; Kiros et al., 2015)
- Splits into 160-token chunks, models the last 40 tokens conditioned on the first 120 tokens
- Sub-samples 1k/1k chunks for validation/test

Baselines & Models

- Baselines
  - Generator P_ϕ (BASE LM)
  - Locally normalized residual model (RALM)

$$\log P_{\mathsf{RALM}}(x_t|x_{<t}) = \log P_\phi(x_t|x_{<t}) + \log P_\theta(x_t|x_{<t}) + const$$

  - Big language model matching #parameters (BALM)
- P_{joint} (P_ϕ is always BASE LM, only energy function E_θ varies)
  - Unidirectional transformer (UNIT)
  - Bidirectional transformer (BIT)
  - Pretrained Bidirectional transformers (BIT with *)

Baselines & Models

- Baselines
  - Generator P_ϕ (BASE LM)
  - Locally normalized residual model (RALM)

$$\log P_{\mathsf{RALM}}(x_t|x_{<t}) = \log P_\phi(x_t|x_{<t}) + \log P_\theta(x_t|x_{<t}) + const$$

  - Big language model matching #parameters (BALM)
- P_{joint} (P_ϕ is always BASE LM, only energy function E_θ varies)
  - Unidirectional transformer (UniT)
  - Bidirectional transformer (BiT)
  - Pretrained Bidirectional transformers (BiT with *)

Results - Estimated Perplexities

Model (#parameters)	CC-News	Toronto Book Corpus
Without External Data		
Base LM (203M)	14.89	18.14
RALM (LM+203M)	14.89	18.17
BALM (408M)	13.92	18.24
joint UniT (LM+203M)	13.81-13.82	**17.46-17.48**
joint BiT-Base (LM+125M)	13.01-13.03	-
joint BiT-Med (LM+203M)	**12.38-12.42**	-
With External Data		
joint BiT-Base* (LM+125M)	12.93-12.95	16.17-16.18
joint BiT-Large* (LM+355M)	**12.10-12.16**	**15.17-15.22**

Limitation of Perplexity Estimation

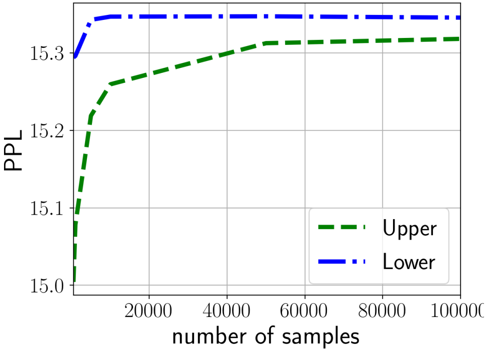
- We used asymptotic bounds and used 100k samples to estimate $Z(\theta, \phi)$
- There is no guarantee that 100k is enough

Limitation of Perplexity Estimation

- We used asymptotic bounds and used 100k samples to estimate $Z(\theta, \phi)$
- There is no guarantee that 100k is enough

Limitation of Perplexity Estimation

- We used asymptotic bounds and used 100k samples to estimate $Z(\theta, \phi)$
- There is no guarantee that 100k is enough



Per-Step Perplexity

$$P_{\text{joint}}(x_t|x_{<t}) = P_\phi(x_t|x_{<t}) \frac{\mathbb{E}_{x'_{t+1},\cdots,x'_T \sim P_\phi(\cdot|x_{\leq t})}[\exp(-E_\theta(x_{\leq t}, x'_{t+1:T}))]}{\mathbb{E}_{x'_t,\cdots,x'_T \sim P_\phi(\cdot|x_{\leq t-1})}[\exp(-E_\theta(x_{\leq t-1}, x'_{t:T}))]}$$

- We marginalize $P_{\text{joint}}(\mathbf{x})$ to get $P_{\text{joint}}(x_t|x_{<t})$
- We use the asymptotic bounds to estimate those expectations
- For the last step, we can also compute the expectation analytically

Per-Step Perplexity

$$P_{\text{joint}}(x_t|x_{<t}) = P_\phi(x_t|x_{<t}) \frac{\mathbb{E}_{x'_{t+1},\cdots,x'_T \sim P_\phi(\cdot|x_{\leq t})}[\exp(-E_\theta(x_{\leq t}, x'_{t+1:T}))]}{\mathbb{E}_{x'_t,\cdots,x'_T \sim P_\phi(\cdot|x_{\leq t-1})}[\exp(-E_\theta(x_{\leq t-1}, x'_{t:T}))]}$$

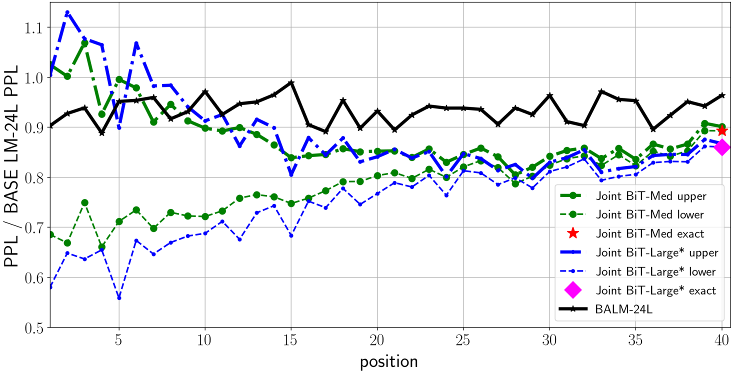
- We marginalize $P_{\text{joint}}(\mathbf{x})$ to get $P_{\text{joint}}(x_t|x_{<t})$
- We use the asymptotic bounds to estimate those expectations
- For the last step, we can also compute the expectation analytically

Per-Step Perplexity

$$P_{\mathsf{joint}}(x_t|x_{<t}) = P_\phi(x_t|x_{<t})\frac{\mathbb{E}_{x'_{t+1},\cdots,x'_T\sim P_\phi(\cdot|x_{\leq t})}[\exp(-E_\theta(x_{\leq t}, x'_{t+1:T}))]}{\mathbb{E}_{x'_t,\cdots,x'_T\sim P_\phi(\cdot|x_{\leq t-1})}[\exp(-E_\theta(x_{\leq t-1}, x'_{t:T}))]}$$

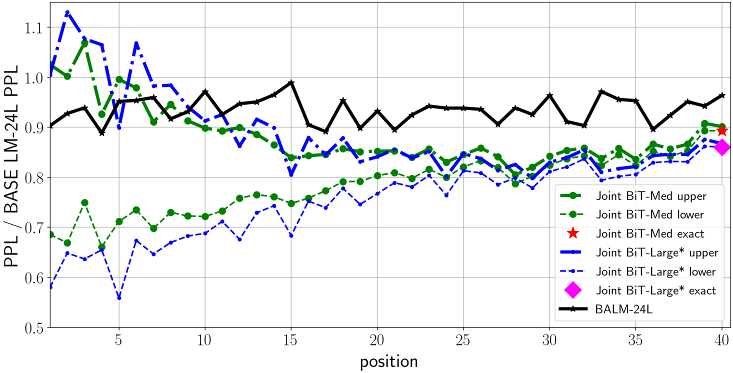
- We marginalize $P_{\mathsf{joint}}(\mathbf{x})$ to get $P_{\mathsf{joint}}(x_t|x_{<t})$
- We use the asymptotic bounds to estimate those expectations
- For the last step, we can also compute the expectation analytically

Results - Per-Step Perplexity



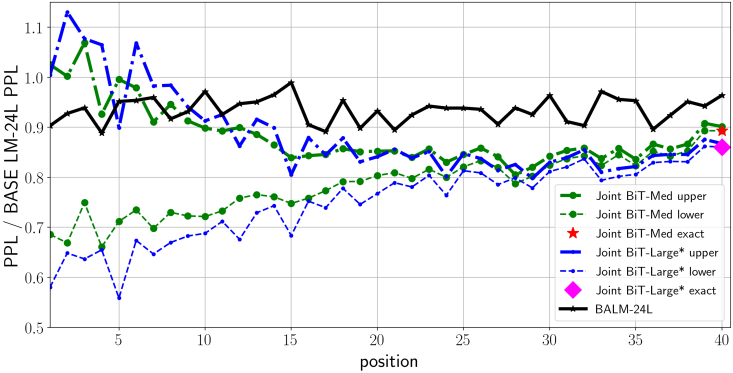
- We know for sure that the final step gets a better PPL
- The estimate is accurate at the final step

Results - Per-Step Perplexity



- We know for sure that the final step gets a better PPL
- The estimate is accurate at the final step

Results - Per-Step Perplexity



- We know for sure that the final step gets a better PPL
- The estimate is accurate at the final step

Does better PPL lead to better generations?

Read each of the three pairs of text below and decide which is a more reasonable **extension** of the **initial words** . Note: do not worry if one or both extensions is incomplete.

- ◯ **.... 'If you try to tinker with this without the tools that only Congress has, you are as likely to break the cloud as you are to fix it,' he said. Google, which has waged similar battles with the government, and an array of other leading tech companies are supporting Microsoft in the case. Justices Sonia Sotomayor and Ruth Bader Ginsburg suggested the wait-for-Congress approach had some appeal. 'Wouldn't it be wiser just to say, 'Let's leave things as they are—if Congress wants to regulate in this brave new world, it should just give it up,'' Ginsburg said, according to a summary of the opinion written for the high court's concurrence. The tech companies have a history of fighting government regulations in court, and have...**

- ◯ **.... 'If you try to tinker with this without the tools that only Congress has, you are as likely to break the cloud as you are to fix it,' he said. Google, which has waged similar battles with the government, and an array of other leading tech companies are supporting Microsoft in the case. Justices Sonia Sotomayor and Ruth Bader Ginsburg suggested the wait-for-Congress approach had some appeal. 'Wouldn't it be wiser just to say, 'Let's leave things as they are—if Congress wants to regulate in this brave new world, it should be regulating in this brave new world?',' wrote Sotomayor and Bader Ginsburg. A ruling is due by the end of June. If it's approved by Congress, the court could...**

Results - Human Evaluation

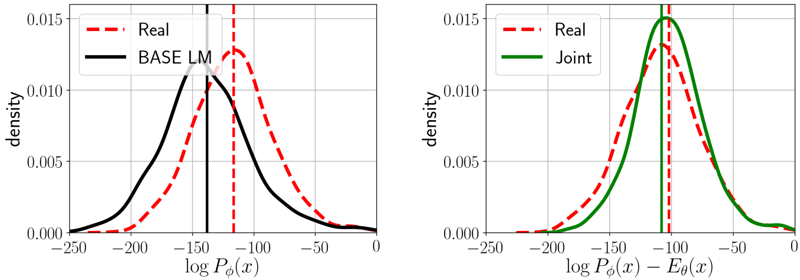
Model1 (baseline)		Model2 (compared)	Rate	p-value
base LM	$\leq$	BALM	54.85%	0.050
BALM	$<$	joint BiT-Med	56.23%	0.015
joint BiT-Large*		Human	55.21%	0.036

Energy Distribution

- If the joint model P_{joint} matches the data distribution P_{data}, then statistics of samples from the two distributions should also match (Du and Mordatch, 2019)

Energy Distribution

- If the joint model P_{joint} matches the data distribution P_{data}, then statistics of samples from the two distributions should also match (Du and Mordatch, 2019)



What does E_θ capture? Per-Step Distribution

Given a prefix $x_{<t}$, check $P(x_t|x_{<t})$

Base LM P_ϕ:

0: P: 0.21, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [and]

1: P: 0.15, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [loving]

2: P: 0.10, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [caring]

3: P: 0.04, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [kind]

4: P: 0.03, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [loved]

5: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [intelligent]

6: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [compassionate]

7: P: 0.01, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [gentle]

8: P: 0.01, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [sweet]

9: P: 0.01, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [beautiful]

What does E_θ capture? Per-Step Distribution

Given a prefix $x_{<t}$, check $P(x_t|x_{<t})$

Our model P_{joint}:

0: P: 0.34, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [and]

1: P: 0.04, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [funny]

2: P: 0.04, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [fun]

3: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [happy]

4: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [thoughtful]

5: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [hard]

6: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [generous]

7: P: 0.02, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [outgoing]

8: P: 0.01, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [always]

9: P: 0.01, ... He was a beautiful, loving, caring, kind, sweet, gentle, intelligent, [compassionate]

What does E_θ capture? Per-Step Distribution

Given a prefix $x_{<t}$, check $P(x_t|x_{<t})$

Base LM P_ϕ:

0: P: 0.15, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [movie]

1: P: 0.11, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [TV]

2: P: 0.10, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [comedy]

3: P: 0.08, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [show]

4: P: 0.05, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [song]

5: P: 0.03, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [television]

6: P: 0.02, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [film]

7: P: 0.02, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [series]

8: P: 0.02, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [book]

9: P: 0.02, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [documentary]

What does E_θ capture? Per-Step Distribution

Given a prefix $x_{<t}$, check $P(x_t|x_{<t})$

Our model P_{joint}:

0: P: 0.14, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [TV]

1: P: 0.06, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [book]

2: P: 0.06, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [television]

3: P: 0.06, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [show]

4: P: 0.06, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [song]

5: P: 0.05, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [documentary]

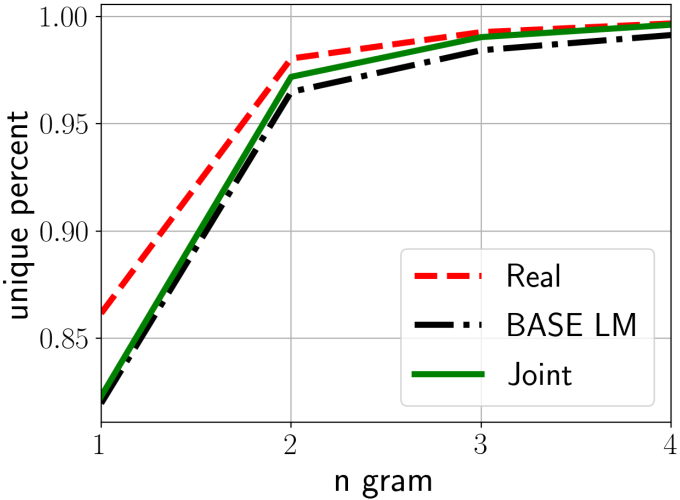
6: P: 0.05, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [comedy]

7: P: 0.04, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [movie]

8: P: 0.03, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [musical]

9: P: 0.03, ... a TV show. I'm going to do a show. I'm going to do a movie. I'm going to do a [film]

Unique Ratio of Ngrams



Are generations from P_{joint} harder to discriminate?

- Generations from P_ϕ: false positive rate 17.8%, accuracy 89.9%[3]
- Generations from P_{joint}: false positive rate 31.8%, accuracy 82.9%

[3] Results are taken from Bakhtin et al. (2021)

Limitations

- Implicit assumption that P_ϕ is good (since generation is only resampling samples from it)
- Generation is expensive
  - Energy function classes that enable efficient generation (such as fixed-order MRFs (Deng and Rush, 2020))?
  - Other sampling techniques (such as MH)?
- Generator P_ϕ is fixed during training
  - Use RL to update P_ϕ as well?

Limitations

- Implicit assumption that P_ϕ is good (since generation is only resampling samples from it)
- Generation is expensive
  - Energy function classes that enable efficient generation (such as fixed-order MRFs (Deng and Rush, 2020))?
  - Other sampling techniques (such as MH)?
- Generator P_ϕ is fixed during training
  - Use RL to update P_ϕ as well?

Limitations

- Implicit assumption that P_ϕ is good (since generation is only resampling samples from it)
- Generation is expensive
  - Energy function classes that enable efficient generation (such as fixed-order MRFs (Deng and Rush, 2020))?
  - Other sampling techniques (such as MH)?
- Generator P_ϕ is fixed during training
  - Use RL to update P_ϕ as well?

Conclusions

- A generative model of text combining base language models and energy network (discriminator) residuals
  - Different from GAN (Goodfellow et al., 2014), the discriminator is part of the model
- NCE training is very stable
- The joint model gets higher data likelihood than baselines
- The sampling-resampling scheme generates better text
- A natural way to use pretrained bi-directional transformers BERT variants (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019, inter alia) for text generation

Conclusions

- A generative model of text combining base language models and energy network (discriminator) residuals
  - Different from GAN (Goodfellow et al., 2014), the discriminator is part of the model
- NCE training is very stable
- The joint model gets higher data likelihood than baselines
- The sampling-resampling scheme generates better text
- A natural way to use pretrained bi-directional transformers BERT variants (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019, inter alia) for text generation

Conclusions

- A generative model of text combining base language models and energy network (discriminator) residuals
  - Different from GAN (Goodfellow et al., 2014), the discriminator is part of the model
- NCE training is very stable
- The joint model gets higher data likelihood than baselines
- The sampling-resampling scheme generates better text
- A natural way to use pretrained bi-directional transformers BERT variants (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019, inter alia) for text generation

Conclusions

- A generative model of text combining base language models and energy network (discriminator) residuals
  - Different from GAN (Goodfellow et al., 2014), the discriminator is part of the model
- NCE training is very stable
- The joint model gets higher data likelihood than baselines
- The sampling-resampling scheme generates better text
- A natural way to use pretrained bi-directional transformers BERT variants (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019, inter alia) for text generation

Conclusions

- A generative model of text combining base language models and energy network (discriminator) residuals
  - Different from GAN (Goodfellow et al., 2014), the discriminator is part of the model
- NCE training is very stable
- The joint model gets higher data likelihood than baselines
- The sampling-resampling scheme generates better text
- A natural way to use pretrained bi-directional transformers BERT variants (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019, inter alia) for text generation

Q & A

Thank you!

References I

Bakhtin, A., Deng, Y., Gross, S., Ott, M., Ranzato, M., and Szlam, A. (2021). Residual energy-based models for text. *Journal of Machine Learning Research*, 22(40):1–41.

Bakhtin, A., Gross, S., Ott, M., Deng, Y., Ranzato, M., and Szlam, A. (2019). Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Deng, Y. and Rush, A. (2020). Cascaded text generation with markov transformers. *Advances in Neural Information Processing Systems*, 33.

References II

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Du, Y. and Mordatch, I. (2019). Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689.

Fedus, W., Zoph, B., and Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Gehrmann, S., Strobelt, H., and Rush, A. M. (2019). Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

References III

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. (2019). Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., and Urtasun, Raquel an d Fidler, S. (2015). Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*.

Nagel, S. (2016). Cc-news. `http://web.archive.org/save/http://commoncrawl.org/2016/10/news-dataset-available/`.

Nowozin, S. (2018). Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*.

Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., and Sutskever, I. (2019a). Better language models and their implications. *OpenAI Blog https://openai. com/blog/better-language-models*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019b). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

References V

Wang, B. and Ou, Z. (2018). Learning neural trans-dimensional random field language models with noise-contrastive estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6134–6138. IEEE.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. (2019). Defending against neural fake news. *arXiv preprint arXiv:1905.12616*.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.