Generative models for discrete data (mostly language)

Max Ryabinin, Yandex Research

In this lecture

- Modern generative models with discrete data
- What works... and what doesn't (currently)
- Discrete versions of:
  - Autoregressive models
  - GANs
  - VAEs
  - Flow-based models
- Current trends and open questions

The difficulties of generating language

- For continuous domains, using samples as input is simple
- Tokens of text are not (strictly) differentiable!
- Even small changes in local/global structure can lead to quality degradations

Language modeling

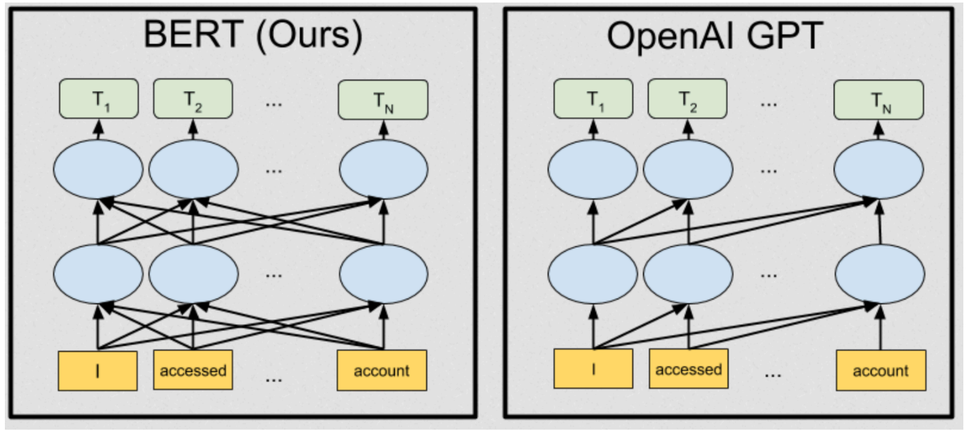
- Simplest approach underlying SoTA models (GPT-3, MT etc.)
- Model each token as a categorical distribution conditioned on prefix:

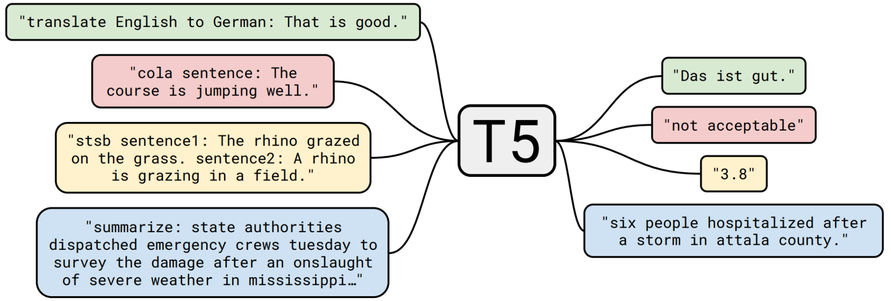
$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P_\theta(y_t|y_{<t}, \mathbf{x}).$$

- Pros: easy to train without bells and whistles, achieves impressive results
- Cons: ~~not fancy enough~~ might suffer from exposure bias, slow sampling etc.

Examples and extensions

- Causal (autoregressive LM): GPT etc.
- Masked («bidirectional») LM: BERT etc.
- Sequence-to-sequence models: all of machine translation, T5 etc.





https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

Language GANs

- Work well for images, why not use them for texts?
- Solve the issue of non-differentiable samples with policy gradient:

$$\nabla_\theta \mathbb{E}_{p_\theta(\mathbf{x})}[R(\mathbf{x})] = \mathbb{E}_{p_\theta(\mathbf{x})}\left[R(\mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{x})\right]$$

- Arising issues are similar to RL: instability due to gradient variance, sparse rewards, tricky evaluation

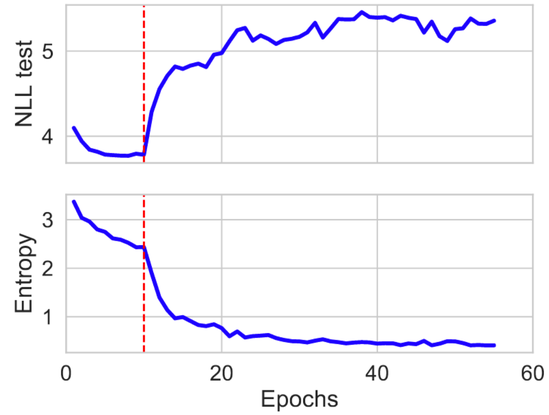


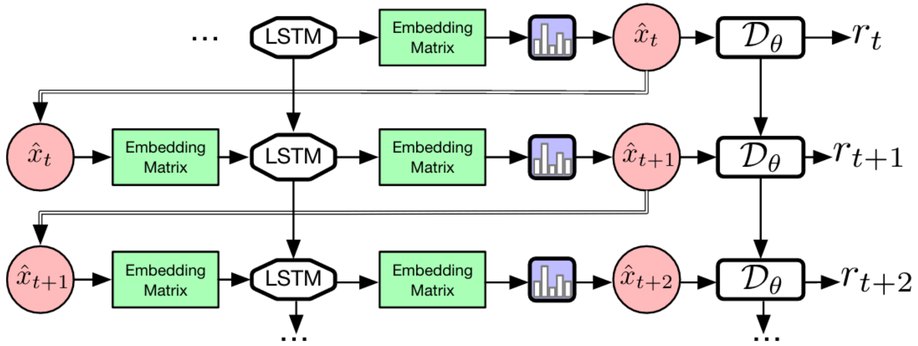
Figure 5: Dotted line indicates the start of GAN training. We notice a clear drop in entropy (spike in NLL_{test}) when moving from maximum-likelihood to adversarial updates.

ScratchGAN

- Large batches for variance reduction

$$\nabla_\theta = \sum_{n=1}^{N} \sum_{t=1}^{T} (R_t^n - b_t) \nabla_\theta \log p_{\boldsymbol{\theta}}(\hat{x}_t^n | \hat{x}_{t-1}^n ... \hat{x}_1^n), \qquad \hat{x}_t^n \sim p_{\boldsymbol{\theta}}(x_t^n | \hat{x}_{t-1}^n ... \hat{x}_1^n)$$

- Recurrent discriminator provides dense rewards



$$r_t = 2\mathcal{D}_{\boldsymbol{\phi}}(\hat{x}_t | x_{t-1} ... x_1) - 1$$

$$R_t = \sum_{s=t}^{T} \gamma^{s-t} r_s$$

Figure 1: ScratchGAN architecture and reward structure.

...falling short

- Intend to improve generation quality and reduce exposure bias
- Do they achieve the goal? Turns out the answer is not so simple
- Both for synthetic and real data, LMs can outperform these methods

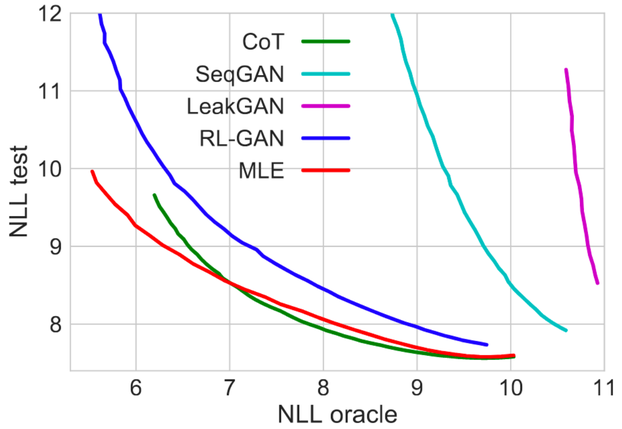
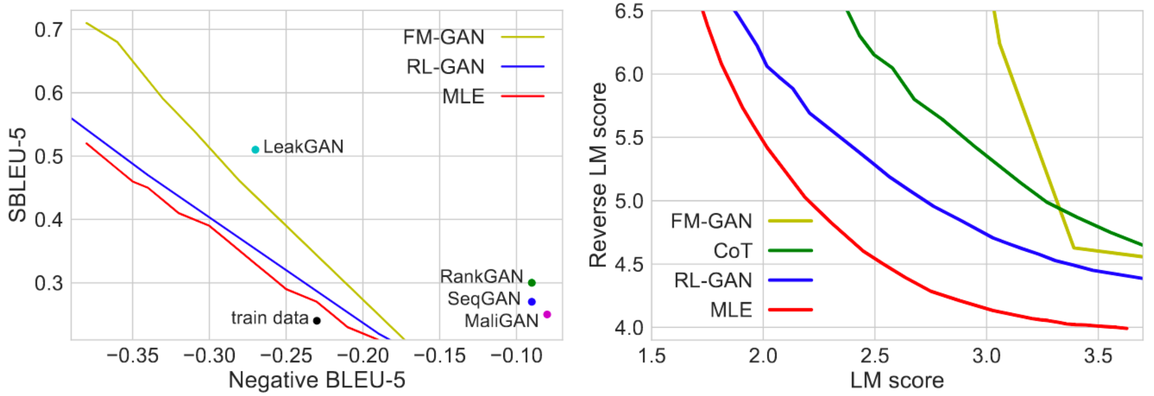


Figure 3: Effect of temperature tuning on the global metrics *(lower is better for both metrics)* for the synthetic task.



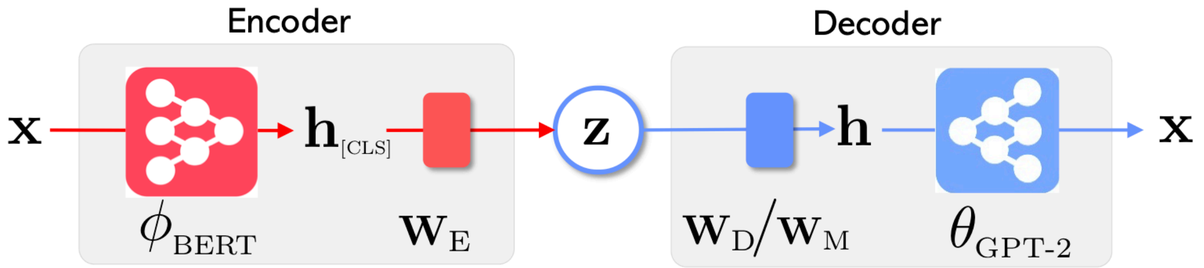
(a) Local metrics (b) Global metrics

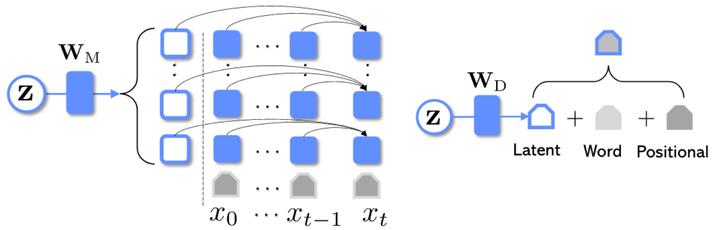
Figure 4: Results on the EMNLP 2017 News dataset. *(lower is better for all metrics)*. MLE under a temperature sweep achieves better quality-diversity trade-off compared to the GAN approaches.

src: https://arxiv.org/abs/1811.02549

VAEs for discrete data

- OPTIMUS is the first large-scale pretrained VAE for language
- Tricks: merge BERT and GPT-2,
- For stability: use KL cyclic annealing, larger batches and thresholding:





(a) Memory (b) Embedding

$$\mathcal{L}'_R = \sum_i \max[\lambda, \mathrm{KL}(q_{\boldsymbol{\phi}}(z_i|\boldsymbol{x})||p(z_i))]$$

https://arxiv.org/abs/2004.04092

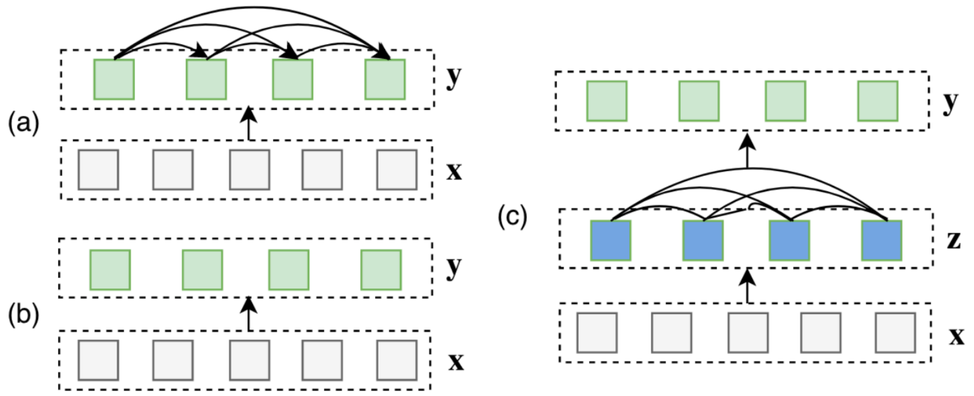
VAEs with discrete latent space

- Might be useful to generate «tokens» for image generation
- Straight-through relaxation or Gumbel-softmax trick (expectation over q)
- Temperature annealing allows to train efficiently

https://arxiv.org/abs/2102.12092

FlowSeq

- A non-autoregressive approach to text generation
- Impressive gains compared to prior work, still worse than sequential



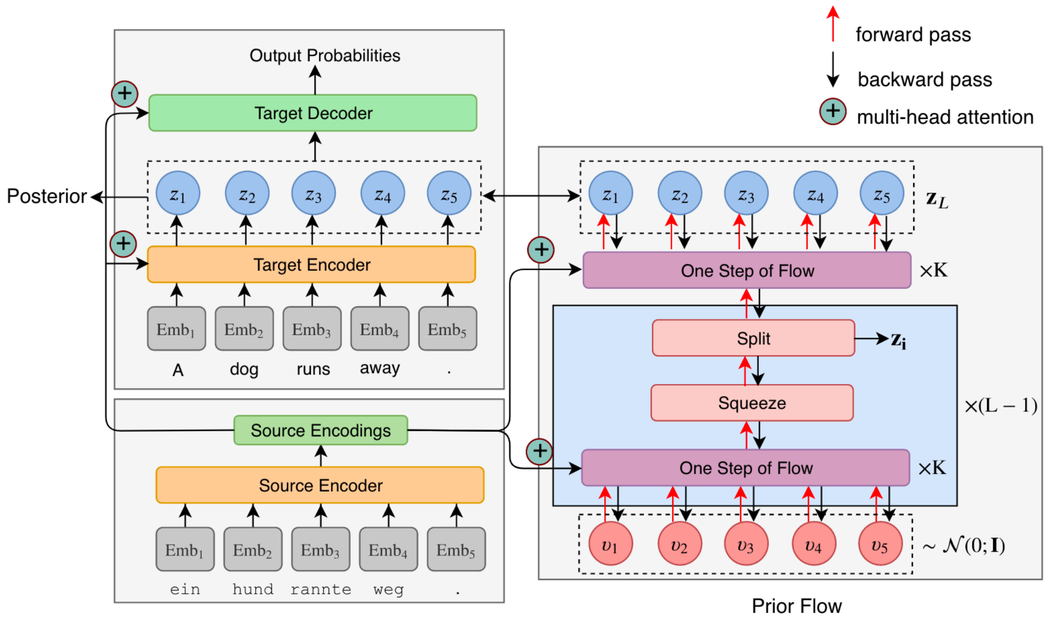


Figure 2: Neural architecture of FlowSeq, including the encoder, the decoder and the posterior networks, together with the multi-scale architecture of the prior flow. The architecture of each flow step is in Figure 3.

https://arxiv.org/abs/1909.02480

Trends and questions

- Are LMs really the best we can come up with?
- Are there limits to scale for pretrained language Transformers?
- Can we utilize latent space models for language in novel ways?

Thank you!