Decoding Strategies

- LMs place a distribution $P(y_i \mid y_1, \ldots, y_{i-1})$
- seq2seq models place a distribution $P(y_i \mid \mathbf{x}, y_1, \ldots, y_{i-1})$
- Generation from both models looks similar; how do we do it?
  - Option 1: $\max_{y_i} P(y_i \mid y_1, \ldots, y_{i-1})$: greedily take best option
  - Option 2: use beam search to find the sequence with the highest prob.
  - Option 3: sample from the model; draw y_i from that distribution
- Beam search is great for applications like machine translation or question answering where the answers are somewhat constrained. But LLMs are increasingly being used for **open-ended** generation tasks where there is not one right answer. How do these compare here?

Story Generation (GPT-2)

Context: 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.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

- Beam search degenerates and starts repeating. If you see a fragment repeated 2-3x, it has very high probability to keep repeating
- Sampling is too noisy, introduces many grammatical errors

Holtzman et al. (2019)

Drawbacks of Beam Search: Degeneration

- Beam search fails because the model is *locally normalized*
- Let's look at all the individual decisions that get made here

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

P(*Nacional* | ... *Universidad*) is high

P(*Autónoma* | ... *Universidad Nacional*) is high

P(*de* | ... *Universidad Nacional Autónoma*) is high

P(*México* | *Universidad Nacional Autónoma de*) is high

P(*/* | ... *México*) and P(*Universidad* | ... *México /*) — these probabilities may be low. But those are just 2/6 words of the repeating fragment

- Each word is likely given the previous words, but the sequence is bad. (Globally-normalized models can judge this, but are computationally hard to use)

Holtzman et al. (2019)

Drawbacks of Sampling: Long Tail

- Sampling is "too random"

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV

P(y | … they live in a remote desert uninterrupted by)

0.01 roads

0.01 towns

0.01 people

0.005 civilization

Good options, maybe accounting for 90% of the total probability mass. So a 90% chance of getting something good

…

0.0005 town

Long tail with 10% of the mass

- On average, every 10 words we will get something from the 10% tail of the distribution. 100 words -> 1%. Such words can really derail us!

Holtzman et al. (2019)

Nucleus Sampling

P(y | … they live in a remote desert uninterrupted by)

0.01	roads
0.01	towns
0.01	people
0.005	civilization

→ renormalize and sample

cut off after p% of mass

- Define a threshold p. Keep the most probable options account for p% of the probability mass (the *nucleus*), then sample among these.
- To implement: sort options by probability, truncate the list once the total exceeds p, then renormalize and sample from it

Holtzman et al. (2019)

Nucleus Sampling

Method	Perplexity	Self-BLEU4	Repetition %	HUSE
Human	12.38	0.31	0.28	-
Greedy	1.50	0.50	73.66	-
Beam, b=16	1.48	0.44	28.94	-
Stochastic Beam, b=16	19.20	0.28	0.32	-
Pure Sampling	22.73	0.28	0.22	0.67
Sampling, t=0.9	10.25	0.35	0.66	0.79
Top-k=40	6.88	0.39	0.78	0.19
Top-k=640	13.82	**0.32**	**0.28**	0.94
Top-k=40, t=0.7	3.48	0.44	8.86	0.08
Nucleus p=0.95	**13.13**	**0.32**	0.36	**0.97**

- Nucleus: decent perplexity, doesn’t have bad repetitions like greedy/beam do, HUSE (metric that incorporates human evaluation) is much higher, indicates naturalness

Holtzman et al. (2019)

Decoding Strategies

- LMs place a distribution $P(y_i | y_1, \ldots, y_{i-1})$
- seq2seq models place a distribution $P(y_i \mid \mathbf{x}, y_1, \ldots, y_{i-1})$
- Generation from both models looks similar; how do we do it?
  - Option 1: $\max_{y_i} P(y_i | y_1, \ldots, y_{i-1})$ — take greedily best option
  - Option 2: use beam search to find the sequence with the highest prob.
  - ~~Option 3: sample from the model; draw y_i from that distribution~~
  - Option 4: nucleus sampling