

Notes: “The Curious Case of Neural Text Degeneration”

Sun

June 15, 2020

1 Notes:

1. As beam size increases, the model degenerates into either instantly ending the sequence, or repeating itself infinitely.
2. Sampling doesn't work either, as gibberish is generated at times.
3. This paper presents a solution: dynamic sampling. Only sample from the top n samples whos probabilities sum to above p , a hyperparameter.

1.1 High level:

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 México/...

Pure Sampling:

They were cattle called **Bolivian Cavaliers**; 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 **Omwelt** told Sky News. 'They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavaliers.'

Figure 1: Even with substantial human context and the powerful GPT-2 Large language model, Beam Search (size 32) leads to degenerate repetition (highlighted in blue) while pure sampling leads to incoherent gibberish (highlighted in red). When $b \geq 64$, both GPT-2 Large and XL (774M and 1542M parameters, respectively) prefer to stop generating immediately after the given context.

At a high level, the issue is easy to understand. When you use beam search, you (partially) greedily look for the sequence with the highest probability, conditioned on the context. The sequence with the highest probability is then chosen as the output.

But this is degenerate. Given a single phrase repeated 200 times (figure 2), the model assigns higher and higher probability after each iteration. In addition to that, it’s not obviously correct to assume that we should pick the most probable sequence. Grice’s “Maxims of Communication” states that humans actively avoid stating the ‘obvious’. This is further confirmed by looking at probability distributes created by language models on human text (figure 3).

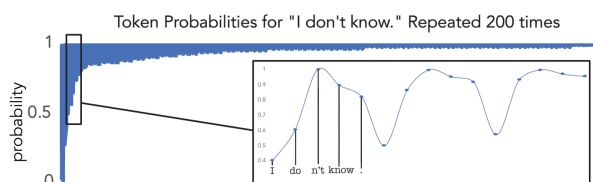


Figure 2: The probability of a repeated phrase increases with each repetition, creating a positive feedback loop.

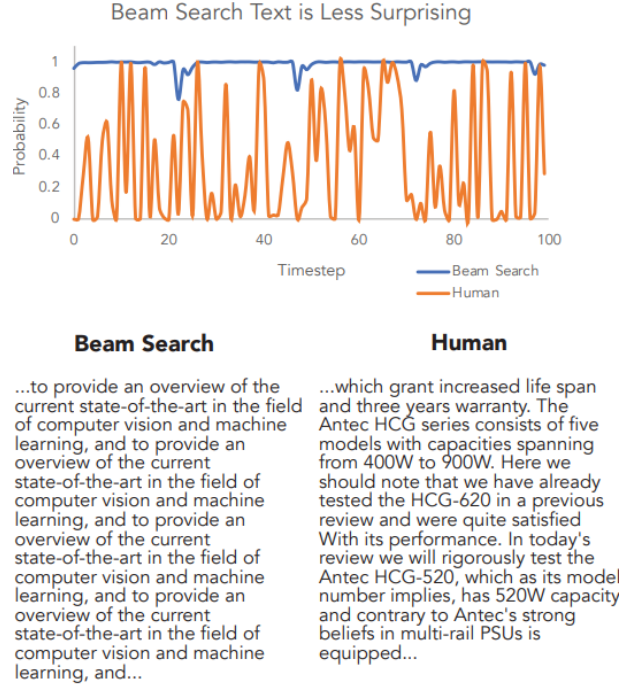


Figure 3: The probability assigned to tokens generated by beam search and humans given the same context. Note the variance within the human text.

This is obviously not a good situation for our language model. It seems that, in some way, we must inject life-giving randomness into our model. Doing this, however, isn't easy. Purely randomly sampling from the output has a decent chance to generate gibberish, and considering exposure bias (during training, transformer-based LMs usually only see the ground truth words due to teacher forcing), the model will 'blindly trust' this gibberish, also resulting in a possible nonsense generation. The authors find that this is a major issue: the model isn't very reliable when unlikely (from the POV of the model) tokens are chosen.

What if we sample only from the top-k words? That's also not so straightforward. Consider sampling from a top-k with very high vs very low entropy (figure 3). Top-k works fine on low entropy. but on high entropy distributions, top-k can fail completely.

This paper presents top-p sampling. Sample from the first n samples whos probabilities sum to greater than p . This removes the "trailing-end" of the distribution which helps cull exposure bias. Additionally, this method does not suffer from the same faults as top-k does.

Empirically? This method works very well. I can speak from my experience too.

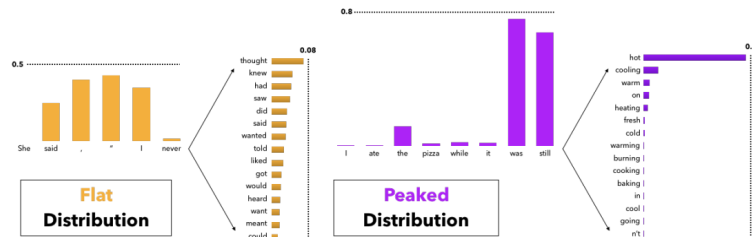


Figure 4: High versus low entropy distributions: visual example where top-k is non-ideal

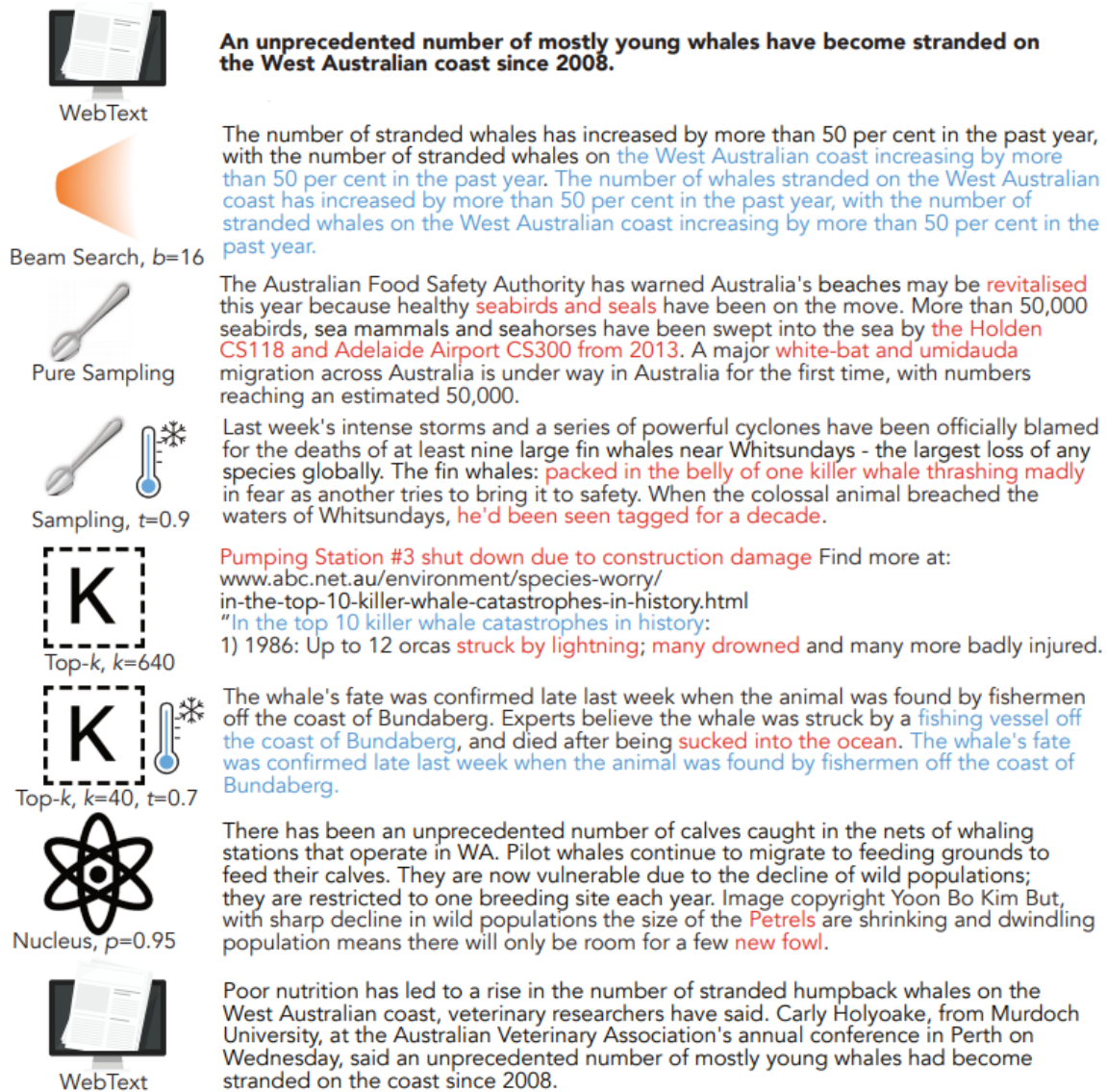


Figure 5: Some different sampling techniques.

1.2 Lowish level:

While the high-level section contains the core idea of the paper and intuitive justification, it's not entirely complete. For example, figure (3) is cool, but it's expected. A model that decodes in such a way to maximize probability will assign high probability to its own sequence, especially when it falls into repetition, which we already know is a pathological state where probability is increased via a positive feedback loop.

Another thing to consider is just how huge exposure bias is, and why it might cause the issues of the model assigning high probabilities to very uninformative sentences. It is possible that, during training, the model simply always picks these uninformative samples, purely relying on teacher forcing to 'force' it in the correct direction. If you consider it from the perspective of optimizing loss, this could be the ideal strategy.

Additionally, like mentioned earlier, natural language does *not* maximize probability, yet common decoding schemes do. Natural language can also be seen as having large stochastic parts (the topic of conversation is usually decided by unknown variables), so maximizing probability doubly does not make sense.

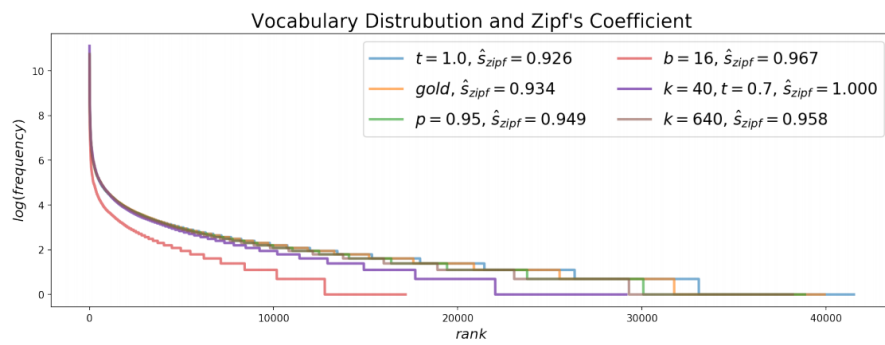


Figure 6: How similar the distributions of frequencies between different sampling techniques are. If you have color issues, you are not allowed to understand this graph.