Current text generation techniques

Sambuddha Roy

March 27, 2020

Roadmap

- Scope of problem: language generation.
- Open ended/closed ended generation.
- Main objectives of generation: modeling human language.
- Previous approaches: how they optimize for one or the other of the objectives.
- The approach of the Nucleus sampling paper.

Scope of problem

Overall topic: we are going to discuss language models. Specifically, how do we use language models to *generate* text? There are two aspects to such language models:

- training
- inference

Here, we are concerned with the second part - inference (i.e. decoding).

Language Models

So... how does a language model work? It models the next token prediction process, i.e. maximizes likelihood of the next token.
Can we use that for generating a sentence? Will the sentences be like "human" sentences?
Natural way: use the context to generate next token (according to the likelihoods) then incorporate that token into the context, and continue.

- This is also called an *auto-regressive* (AR) approach.
- Here is a nice definition of "auto-regressive" from the XLNet paper:
- AR language modeling factorizes the likelihood into a forward product

$$p(x) = \Pi_{t=1}^{T} p(x_t | x_{<t})$$

  and then a parametric model (e.g. a neural network) is trained to model each conditional distribution.

Aside

There is another lens to view language generation from:

- closed ended language generation. Such as, machine translation, image captioning, etc. (the paper calls this “directed generation”)

Aside

There is another lens to view language generation from:

- closed ended language generation. Such as, machine translation, image captioning, etc. (the paper calls this “directed generation”)
- open ended language generation. Like for instance abstractive summarization, etc.

Main desiderata of Language Generation

There are two aspects to language generation:

- Quality
- Diversity

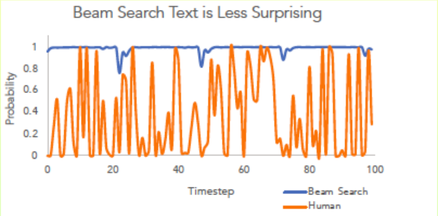
Human beings use language:

- while quality is a “need”,
- diversity is a “want”.

We want to pack in information content in our language, and to this effect, we (as in humans) add in an “element of surprise” in our language.

Diversity is underrated

Here is a surprising image from the paper:



Beam Search

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

Human

...which grant increased life span and three years warranty. The Antec HCG series consists of five models with capacities spanning from 400W to 900W. Here we should note that we have already tested the HCG-620 in a previous review and were quite satisfied With its performance. In today's review we will rigorously test the Antec HCG-520, which as its model number implies, has 520W capacity and contrary to Antec's strong beliefs in multi-rail PSUs is equipped...

Figure 2: The probability assigned to tokens generated by Beam Search and humans, given the

How do we attain *quality*?

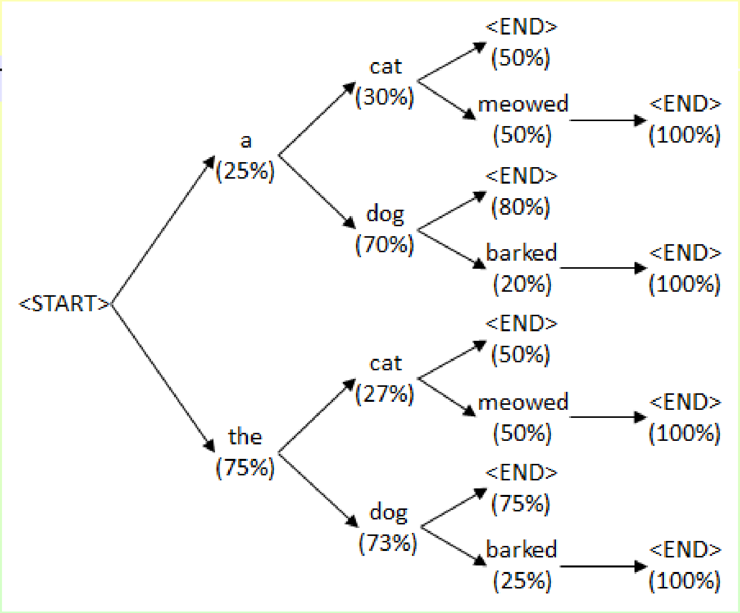
- *Answer*: maximum likelihood decoding. Essentially greedy. At least we can hope that the language generated will be grammatical.
- We essentially want the *sentence* that has the highest probability/likelihood under the language model.

How do we obtain *diversity*?

- *Answer*: usually, by some kind of sampling.
- I.e. We consider the probability distribution of the next token, and sample from that distribution.
- At least in this way, we are giving different candidates a chance (a step in the direction of diversity)

The two extremes

- Maximum likelihood decoding is perhaps too suboptimal. How about some *approximations* to the actual optimum?
- Enter Beam Search. At every step, you have a beam of candidate extensions.
    - At the end pick up the top k beams.
    - We will gloss over details: length normalization, etc.



(Courtesy: geekyisawesome blog)

The two extremes

- Sampling. While we do get diversity here, we sacrifice quality. Why?
- If at some point there is a (slightly) heavy tail, and we end up sampling a low-probability token (word), then that might steer the generated text far away from optimum.
- So how do we disincentivize sampling from the tail? A couple of approaches:
  - Temperature T:
    $$\text{logits} \leftarrow \text{logits}/T$$
    and imagine $T < 1$. Thin out the tail: *rich get richer* effect.
  - Top-k sampling: fix k, send the probability mass of the tail (beyond the top k probability tokens) to 0.

Taking stock

- Ok... so we understand that sampling can get us diversity, perhaps we agree that it might cause a loss in quality.

Taking stock

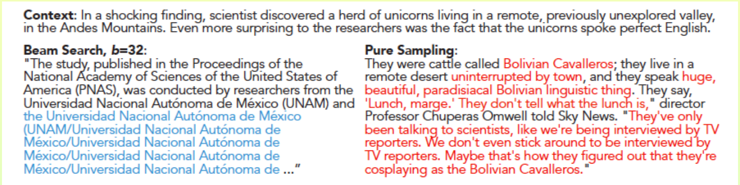
- Ok... so we understand that sampling can get us diversity, perhaps we agree that it might cause a loss in quality.
- But maybe Beam Search is good enough - it gets us quality, perhaps diversity too, right?

Taking stock

- Ok... so we understand that sampling can get us diversity, perhaps we agree that it might cause a loss in quality.
- But maybe Beam Search is good enough - it gets us quality, perhaps diversity too, right?
- Wrong.

Taking stock

- Ok... so we understand that sampling can get us diversity, perhaps we agree that it might cause a loss in quality.
- But maybe Beam Search is good enough - it gets us quality, perhaps diversity too, right?
- Wrong.
- Beam Search tends to keep repeating itself.



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 ..."

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 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 Cavalleros."

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.

- Is this all that surprising? Possible ways this can happen...

- Is this all that surprising? Possible ways this can happen...
- (Very roughly), the language model tries to optimize the next token's fit given the context - something like try to maximize inner product between the embeddings of the token and that of the context, etc.
- So in the future, it is likely that one of the same tokens will again emerge as the "winner".
- This is a rough (and not entirely correct view), but helps us make some sense of the *repetition problem*.
- Part of the problem also is: we generate new text based not on ground truth data (there might be none), but instead, based on other generated text.

Main idea of the paper: motivation

- The main idea is easily derived from understanding failure modes of top-k sampling.
- In top-k sampling, we might still end up picking useless (low probability) candidate tokens.
- Depends on whether the next token distribution is *peaked* or *flat*

Peaked and flat distributions

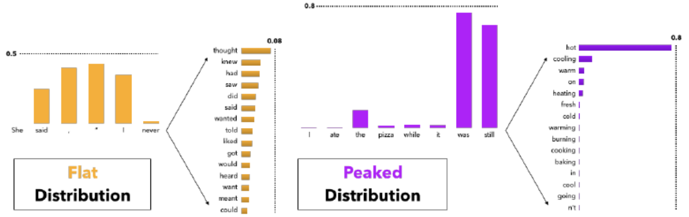


Figure 5: The probability mass assigned to partial human sentences. Flat distributions lead to many moderately probable tokens, while peaked distributions concentrate most probability mass into just a few tokens. The presence of flat distributions makes the use of a small k in top-k sampling problematic, while the presence of peaked distributions makes large k's problematic.

Main idea: top p instead of top-k

- Pick up the top candidates that together account for a probability mass of $\geqslant p$.
- For these candidates, up-weight them, and then sample.

Over to the paper..

What metrics does the Neural degeneration paper consider?

- For the likelihood evaluation, compute **perplexity** (being vague: the amount of *confusion* in the model)

Over to the paper..

What metrics does the Neural degeneration paper consider?

- For the likelihood evaluation, compute **perplexity** (being vague: the amount of *confusion* in the model)
- Important takeaway: it is not enough to “minimize” perplexity but to target the perplexity of the ground truth.

Some takeaways...

- Very well written paper, easy read! Clear motivations.

Some takeaways...

- Very well written paper, easy read! Clear motivations.
- Some more possibilities could have been explored (discuss):
  - This paper considers GPT2 as the language model since GPT2 had shown remarkable prowess in generating stories.
  - From the OpenAI website, they use truncated top-k sampling.

Some takeaways...

- Borrow top p into Beam Search - do not use a uniform beam width of size b but vary that according to p. Computationally a bit costlier, but not sure it would resolve diversity issues.

Some takeaways...

- Borrow top p into Beam Search - do not use a uniform beam width of size b but vary that according to p. Computationally a bit costlier, but not sure it would resolve diversity issues.
- Other ways of ensuring diversity: every token “generates” a repulsive field around itself and prevents the same token from being generated again.

Some takeaways...

- Borrow top p into Beam Search - do not use a uniform beam width of size b but vary that according to p. Computationally a bit costlier, but not sure it would resolve diversity issues.
- Other ways of ensuring diversity: every token “generates” a repulsive field around itself and prevents the same token from being generated again.
- (such ideas abound in the web search world. For example this paper uses submodular maximization to ensure diversity of search results)
- Of course, not the token itself, but its embedding.

THANK YOU