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.

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.

How do we attain *quality*?

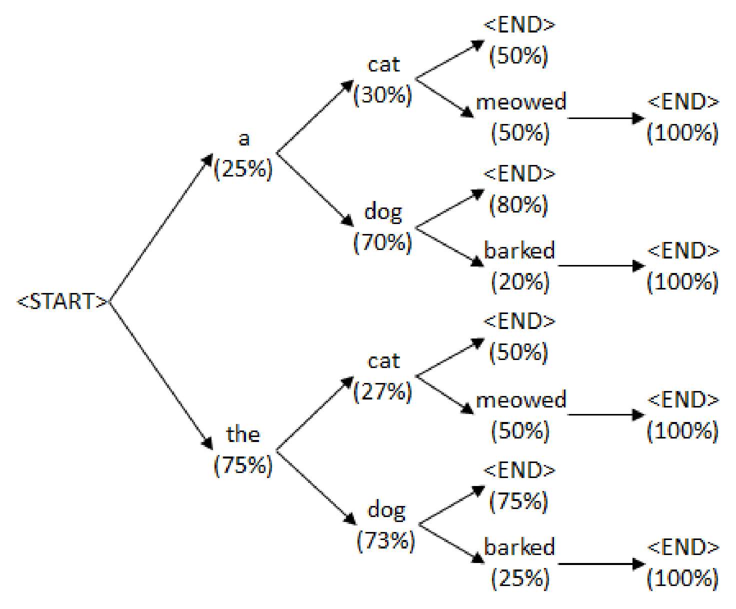
- *Answer*: maximum likelihood decoding. Essentially greedy. At least we can hope that the language generated will be grammatical.
- We essentially want the *sentence* that has the highest probability/likelihood under the language model.

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 believe that it might cause a loss in quality.

Taking stock

- Ok... so we understand that sampling can get us diversity, perhaps we believe 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 believe 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 believe 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.

And some examples...

- Example of nucleus sampling

THANK YOU