LLM Optimization

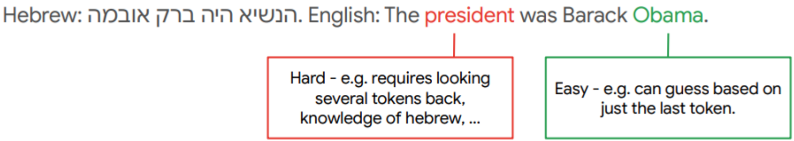
Jordan Boyd-Graber

University of Maryland

Speculative Decoding

Slides adapted from Lily Liu, Ao Li, Bogdan Piula, and Yaniv Leviathan

Motivation



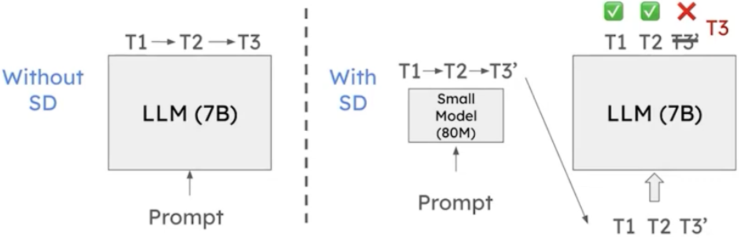
Hebrew: הנשיא היה ברק אובמה. English: The president was Barack Obama.

Hard - e.g. requires looking several tokens back, knowledge of hebrew, ...

Easy - e.g. can guess based on just the last token.

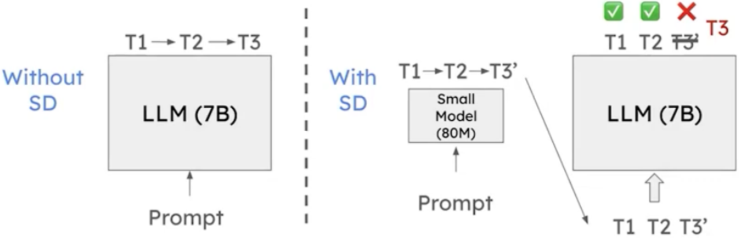
- Models are big an expensive
- But you don’t need the full power for every token
- Can we use a cheaper model for easier tokens?

Idea



- Generate from cheap model
- If expensive model likes it, keep it
- If not, resample

Idea



- Generate from cheap model
- If expensive model likes it, keep it
- If not, resample
- **Why?** Verification cheaper / lower latency than generation (particularly for longer n-grams)

Procedure

- To sample $x \sim p(x)$, we instead sample $x \sim q(x)$
  - Keeping if $q(x) \leq p(x)$, and
  - If $q(x) > p(x)$ we reject the sample with probability $1 - \frac{p(x)}{q(x)}$ and sample x again from an adjusted distribution $p'(x) = \text{norm}(\max(0, p(x) - q(x)))$

For any distributions $p(x)$ and $q(x)$, and x sampled in this way, indeed $x \sim p(x)$.

Proving Equivalence (sketch)

For these sampling strategies to be the same, we want this to be $p(x)$

$$P(x = x') = P(\text{guess accepted}, x = x') + P(\text{guess rejected}, x = x')$$

Proving Equivalence (sketch)

For these sampling strategies to be the same, we want this to be $p(x)$

$$P(x=x')=P(\text{guess accepted}, x=x')+P(\text{guess rejected}, x=x')$$

But we can break this down into two parts:

$$P(\text{guess accepted}, x=x')=q(x')\min\left(1,\frac{p(x')}{q(x')}\right)=\min(q(x'),p(x'))$$

and

$$P(\text{guess rejected}, x=x')=(1-\beta)p'(x')=p(x')-\min(q(x'),p(x'))$$

Proving Equivalence (sketch)

For these sampling strategies to be the same, we want this to be $p(x)$

$$P(x = x') = P(\text{guess accepted}, x = x') + P(\text{guess rejected}, x = x')$$

But we can break this down into two parts:

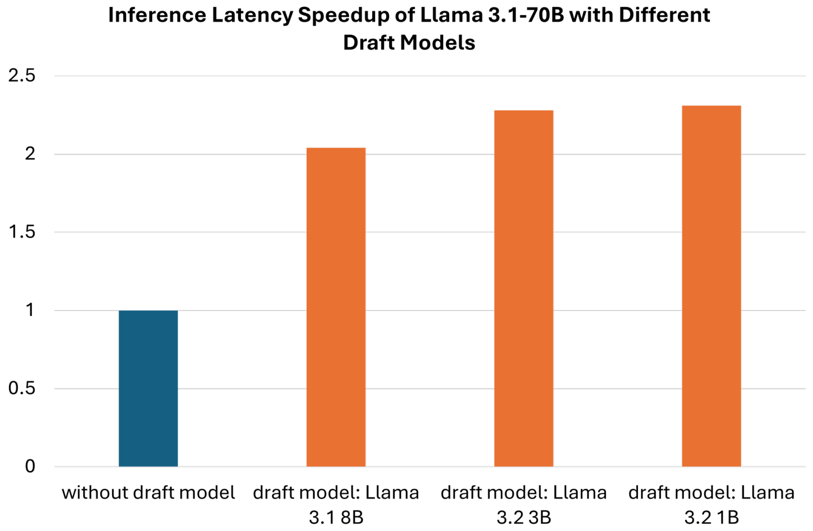
$$P(\text{guess accepted}, x = x') = q(x')\min\left(1, \frac{p(x')}{q(x')}\right) = \min(q(x'), p(x'))$$

and

$$P(\text{guess rejected}, x = x') = (1-\beta)p'(x') = p(x') - \min(q(x'), p(x'))$$

And you get equality when $\beta \equiv \begin{cases} 1 & q(x) \leq p(x) \\ \frac{p(x)}{q(x)} & q(x) > p(x) \end{cases}$

Speedup vs 70B



From Chang Liu

Wrapup

- It can be faster, but not more efficient: strictly more calls
- It's not always faster: if you reject a lot, it can slow things down
- But if you're generating a bunch of simple stuff, can help a lot