Technique2: Faster Constrained Decoding

Liangsheng Yin

Shanghai Jiao Tong University & UC Berkeley

Constrained Decoding Matters

OpenAI JSON Mode

Copilot Code Generation

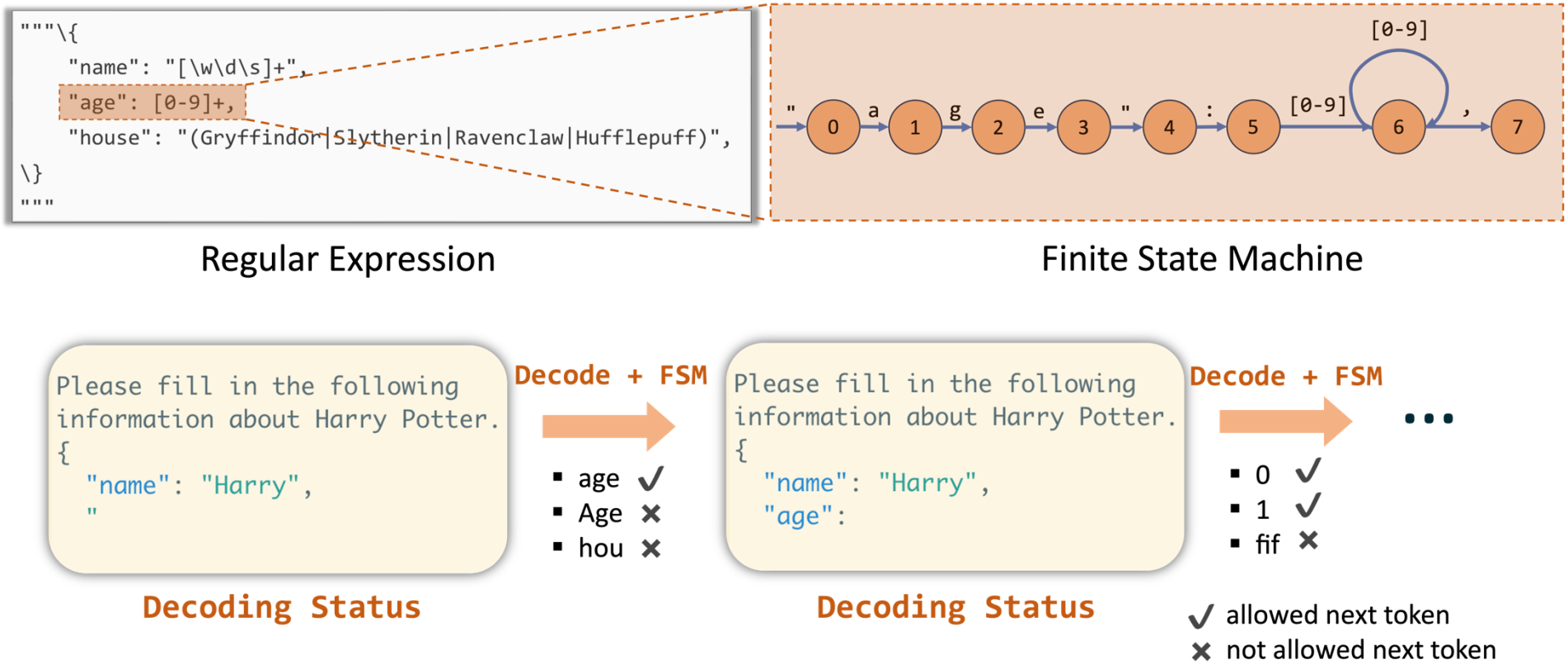
LangChain JSON parser

- output will always adhere to the specified syntactic constraints
- reduces the need for ad-hoc parsing, retrying, and prompting
- without the need for fine-tuning or additional post-processing

https://arxiv.org/pdf/2403.06988v1

Constrained decoding works by masking the invalid tokens

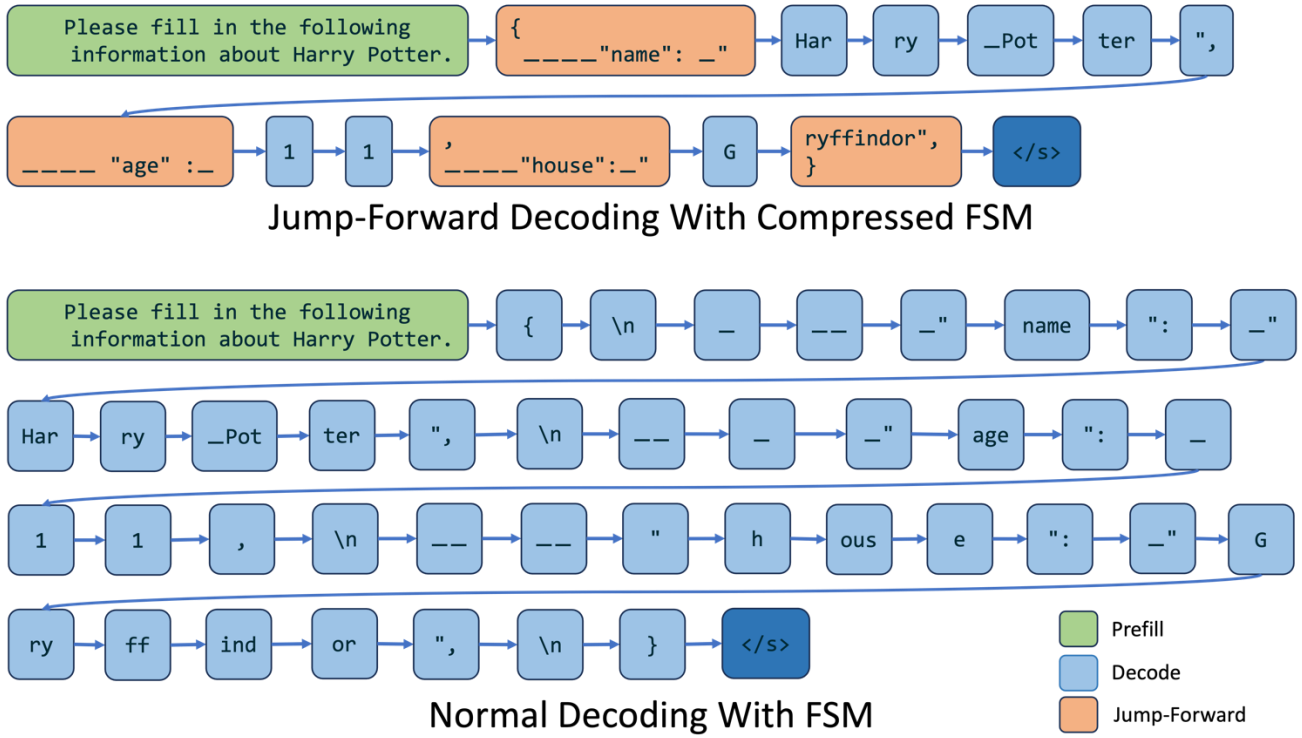
Constraint decoding: JSON schema -> regular expression -> finite state machine -> logit mask

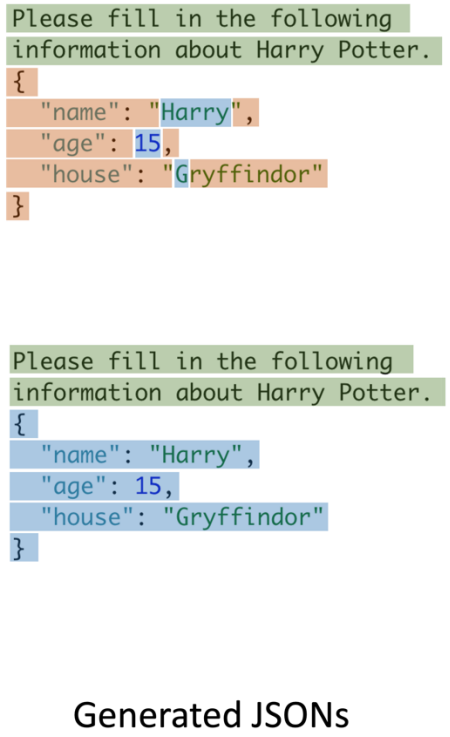


Constrained Decoding With Logits Mask

Compressing the finite state machine allows decoding multiple tokens

We can compress many deterministic paths in the state machine

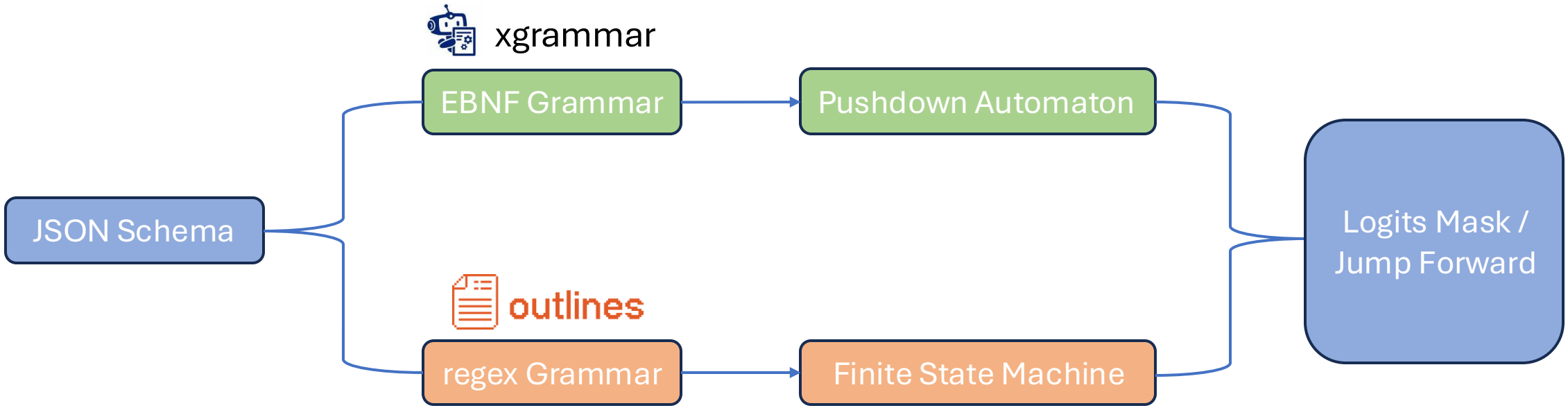




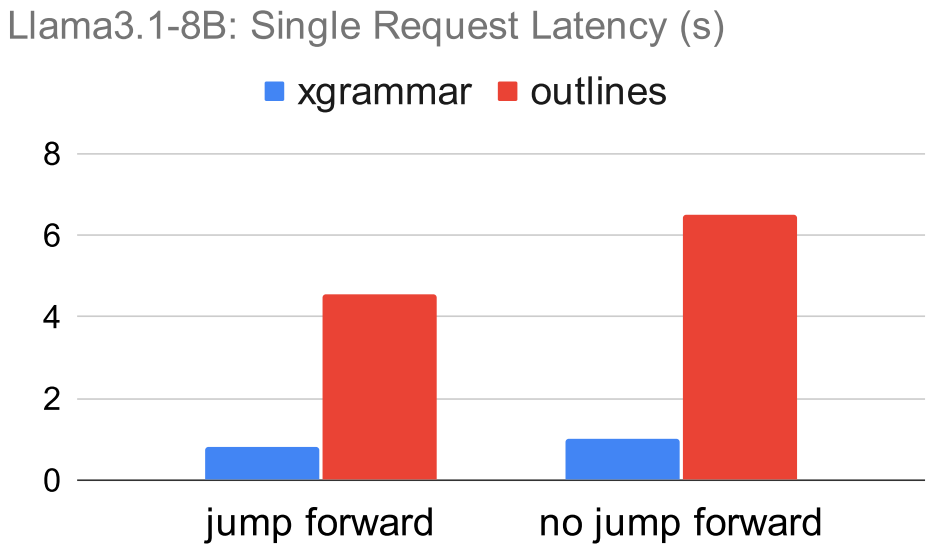
Efficient constrained decoding

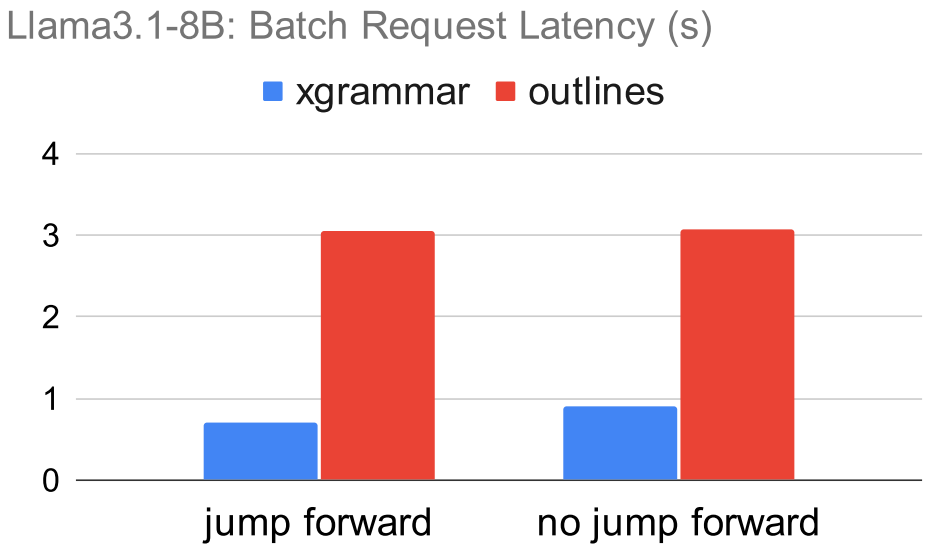
Constrained Decoding Backend: xgrammar and Outlines

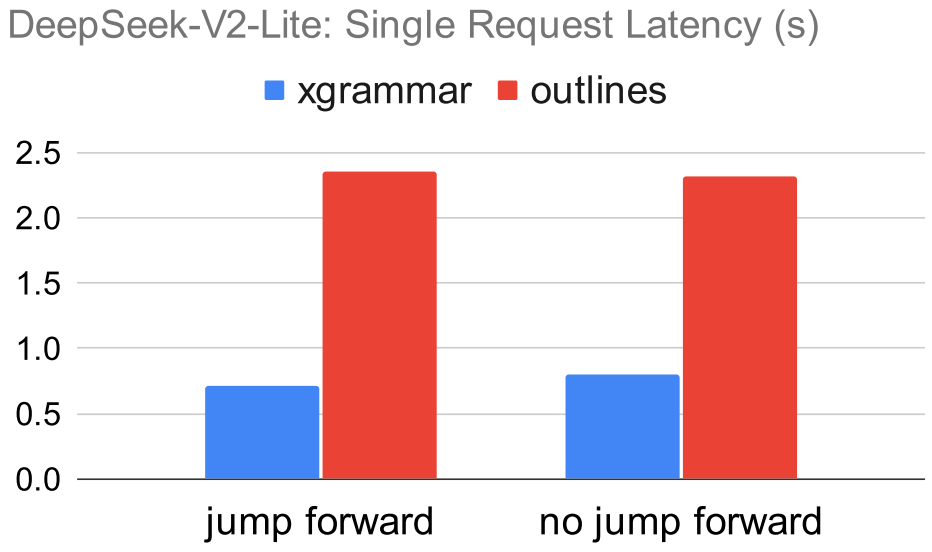
We benchmark with 400 JSON schemas from (BFCL) Berkeley Function-Calling Leaderboard

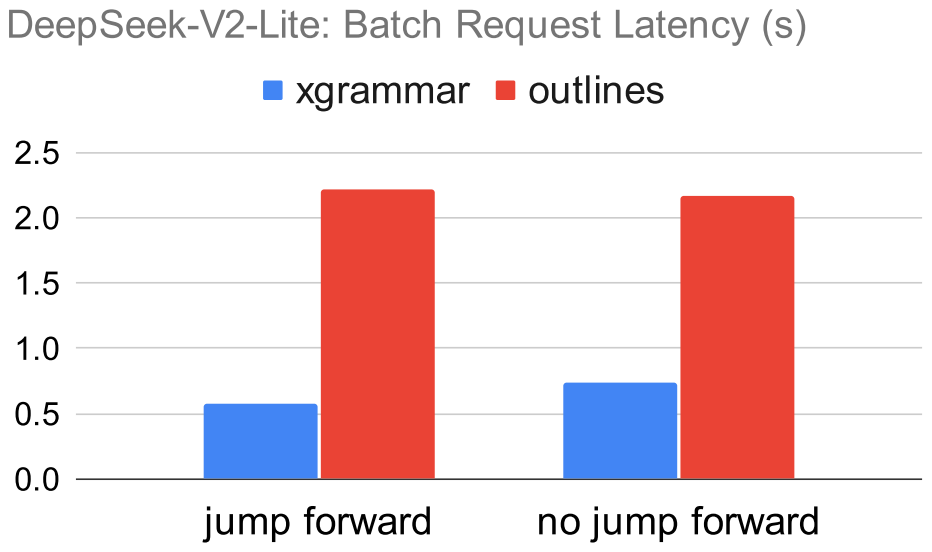


Benchmark Result (Lower is Better)









Question & Answer

Thanks to Ziyi, the xgrammar integration with SGLang is on the way:

https://github.com/sgl-project/sglang/pull/1680

Yixin will deliver more details about xgrammar soon!