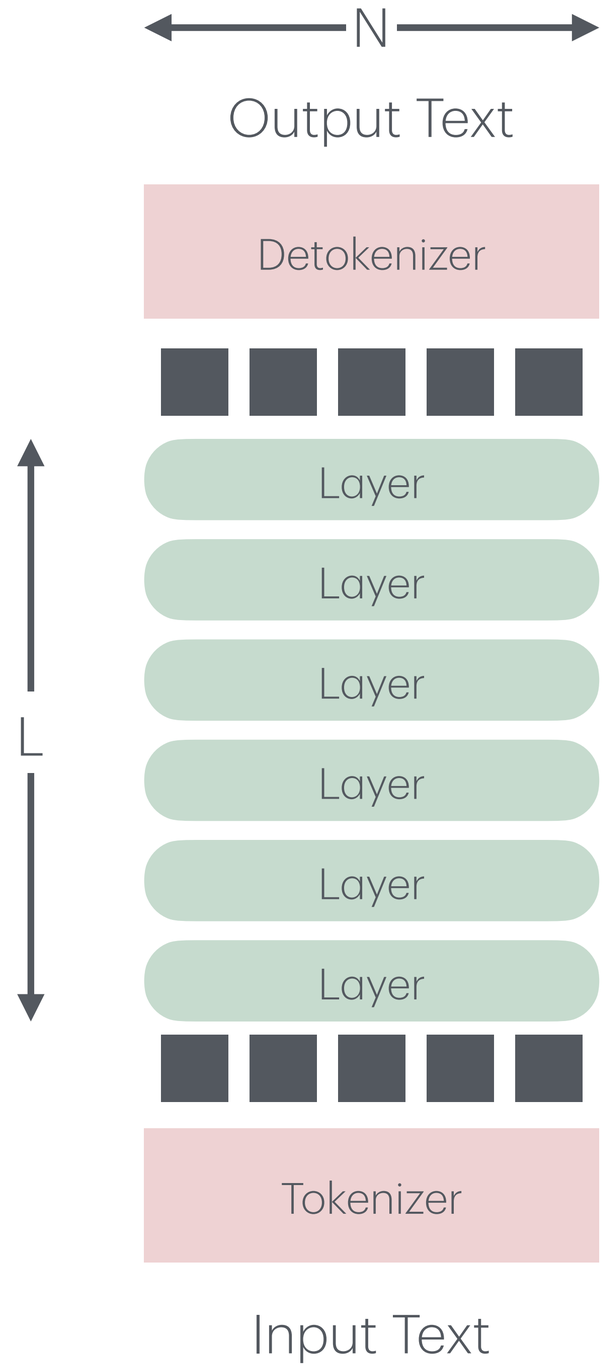
# Speculative Decoding

Philipp Krähenbühl, UT Austin

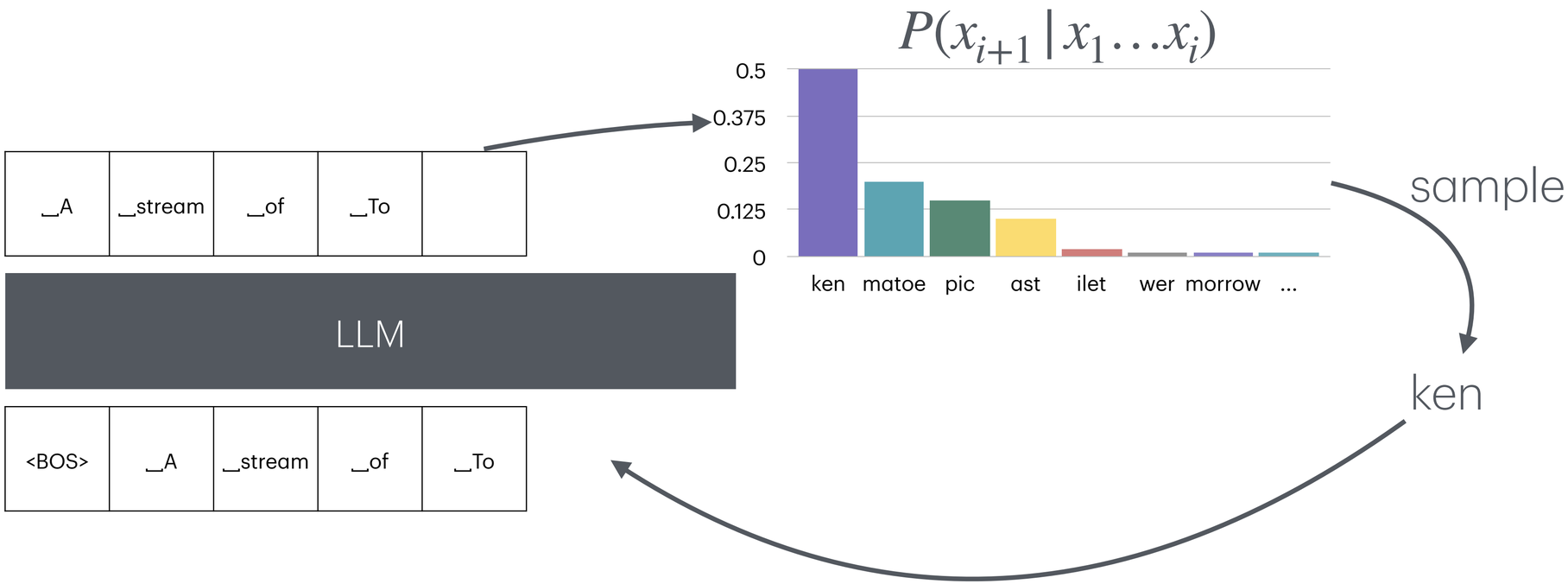
# Training and Generation

## Vanilla Generation

| | Training | Training - Checkpointing | Generation |
|---|---|---|---|
| Peak Memory | O(NL) | O(NL½) | O(N) |
| Runtime | O(N²L) | O(2 N²L) | O(N³L) |
| # forward calls | 1 | 1 | N |



# Generation - A closer look

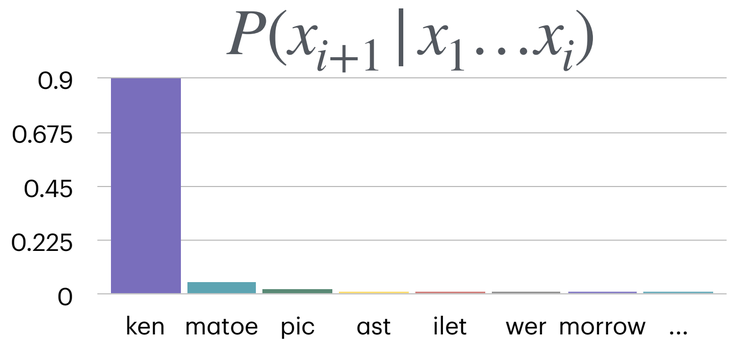



```
x = [<BOS>]
for i = 1..N:
  x_i ~ P(x_i | x)
  x.append(x_i)
```

Overall process: Sample from

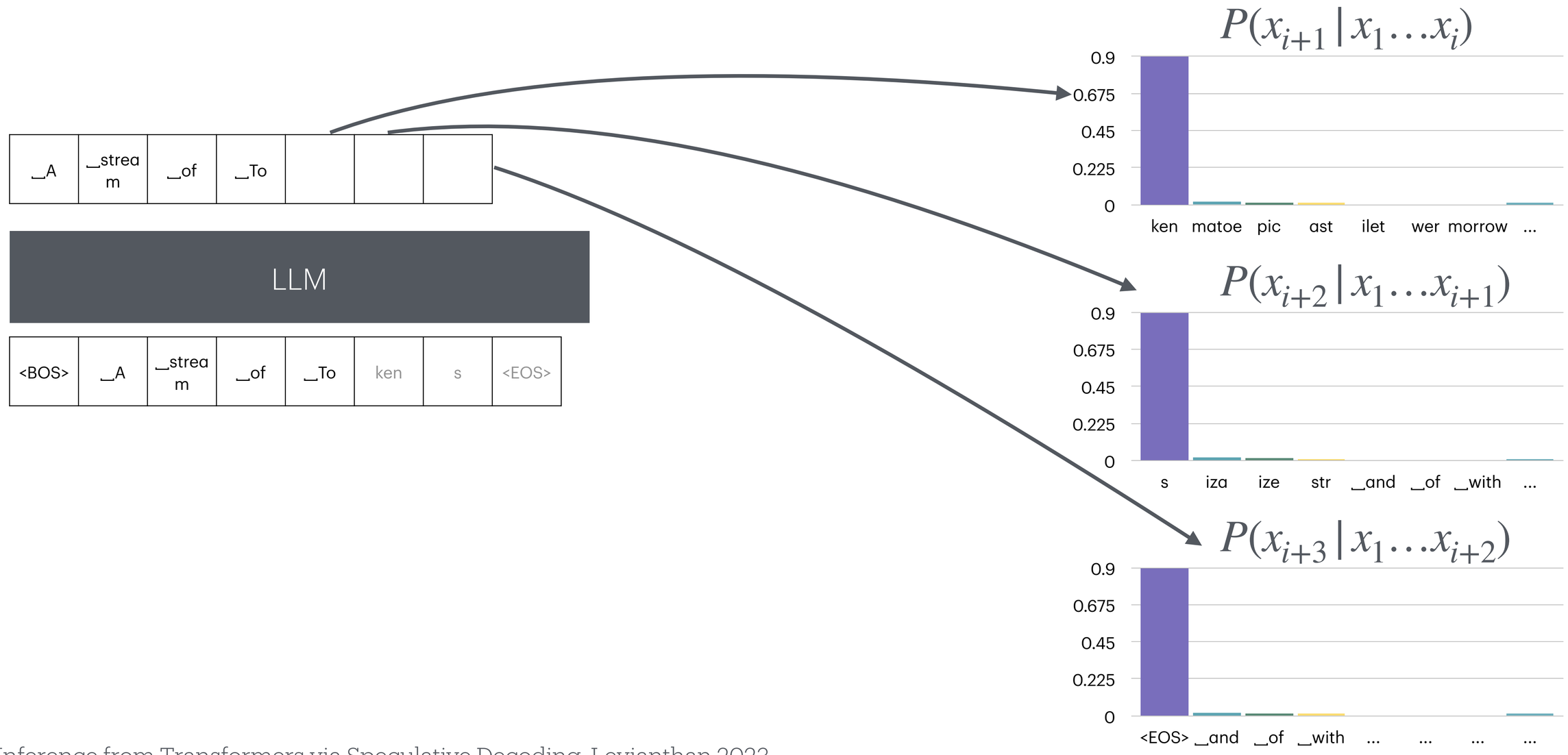
$$P(\mathbf{x}) = \prod_{i=1}^{N} P(x_{i+1} \mid x_1 \ldots x_i)$$

# Generation - A closer look




- Next token probability often highly peaked (next token is obvious)
  - Can we skip predicting obvious tokens?
  - Verify instead

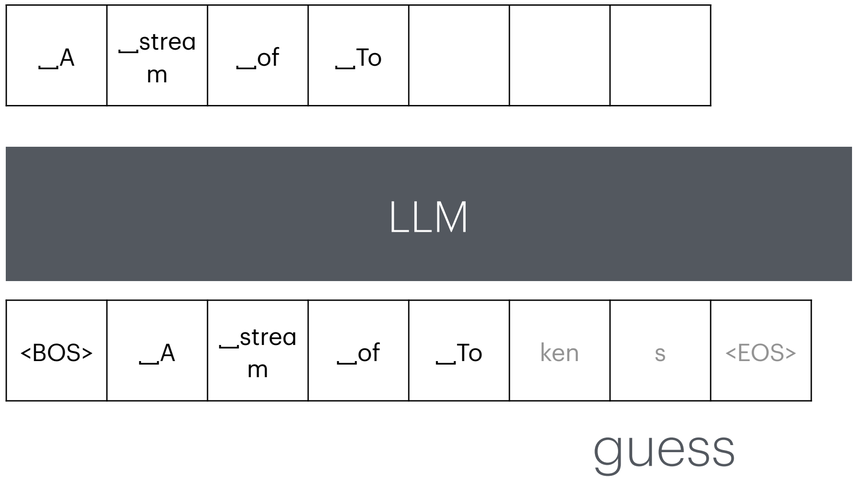
# Verification vs generation




Fast Inference from Transformers via Speculative Decoding, Levianthan 2023

# Speculative decoding

- How do we obtain guess of next tokens?
  - Use a smaller model $Q(x_{i+1} \mid Q_1 \ldots Q_x)$
  - Use a separate head to predict multiple tokens
- How to we accept guess?
  - $P(x_{i+1} \mid x_1 \ldots x_i) \geq Q(x_{i+1} \mid x_1 \ldots x_i)$
  - or proportional to $\dfrac{P(x_{i+1} \mid x_1 \ldots x_i)}{Q(x_{i+1} \mid x_1 \ldots x_i)}$




Fast Inference from Transformers via Speculative Decoding, Levianthan 2023
Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Speculative decoding

## Algorithm

**Algorithm 1** SpeculativeDecodingStep

**Inputs:** $M_p, M_q, prefix$.
▷ Sample $\gamma$ guesses $x_{1,\ldots,\gamma}$ from $M_q$ autoregressively.
**for** $i = 1$ **to** $\gamma$ **do**
  $q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$
  $x_i \sim q_i(x)$
**end for**
▷ Run $M_p$ in parallel.
$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$
  $M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$
▷ Determine the number of accepted guesses $n$.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from $M_p$ if needed.
$p'(x) \leftarrow p_{n+1}(x)$
**if** $n < \gamma$ **then**
  $p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
**end if**
▷ Return one token from $M_p$, and $n$ tokens from $M_q$.
$t \sim p'(x)$
**return** $prefix + [x_1, \ldots, x_n, t]$

*Table 2.* Empirical results for speeding up inference from a T5-XXL 11B model.

| TASK | $M_q$ | TEMP | $\gamma$ | $\alpha$ | SPEED |
|---|---|---|---|---|---|
| ENDE | T5-SMALL ★ | 0 | 7 | 0.75 | **3.4X** |
| ENDE | T5-BASE | 0 | 7 | 0.8 | 2.8X |
| ENDE | T5-LARGE | 0 | 7 | 0.82 | 1.7X |
| ENDE | T5-SMALL ★ | 1 | 7 | 0.62 | **2.6X** |
| ENDE | T5-BASE | 1 | 5 | 0.68 | 2.4X |
| ENDE | T5-LARGE | 1 | 3 | 0.71 | 1.4X |
| CNNDM | T5-SMALL ★ | 0 | 5 | 0.65 | **3.1X** |
| CNNDM | T5-BASE | 0 | 5 | 0.73 | 3.0X |
| CNNDM | T5-LARGE | 0 | 3 | 0.74 | 2.2X |
| CNNDM | T5-SMALL ★ | 1 | 5 | 0.53 | **2.3X** |
| CNNDM | T5-BASE | 1 | 3 | 0.55 | 2.2X |
| CNNDM | T5-LARGE | 1 | 3 | 0.56 | 1.7X |

English to German translation fine tuned on WMT EnDe

Text summarization fine tuned on CCN/DM

Fast Inference from Transformers via Speculative Decoding, Levianthan 2023

# Speculative decoding

## Discussion

- Sampling unbiased
- Requires 2 models
- Only one guess verified

*Table 2.* Empirical results for speeding up inference from a T5-XXL 11B model.

| TASK | $M_q$ | TEMP | $\gamma$ | $\alpha$ | SPEED |
|---|---|---|---|---|---|
| ENDE | T5-SMALL ★ | 0 | 7 | 0.75 | **3.4X** |
| ENDE | T5-BASE | 0 | 7 | 0.8 | 2.8X |
| ENDE | T5-LARGE | 0 | 7 | 0.82 | 1.7X |
| ENDE | T5-SMALL ★ | 1 | 7 | 0.62 | **2.6X** |
| ENDE | T5-BASE | 1 | 5 | 0.68 | 2.4X |
| ENDE | T5-LARGE | 1 | 3 | 0.71 | 1.4X |
| CNNDM | T5-SMALL ★ | 0 | 5 | 0.65 | **3.1X** |
| CNNDM | T5-BASE | 0 | 5 | 0.73 | 3.0X |
| CNNDM | T5-LARGE | 0 | 3 | 0.74 | 2.2X |
| CNNDM | T5-SMALL ★ | 1 | 5 | 0.53 | **2.3X** |
| CNNDM | T5-BASE | 1 | 3 | 0.55 | 2.2X |
| CNNDM | T5-LARGE | 1 | 3 | 0.56 | 1.7X |

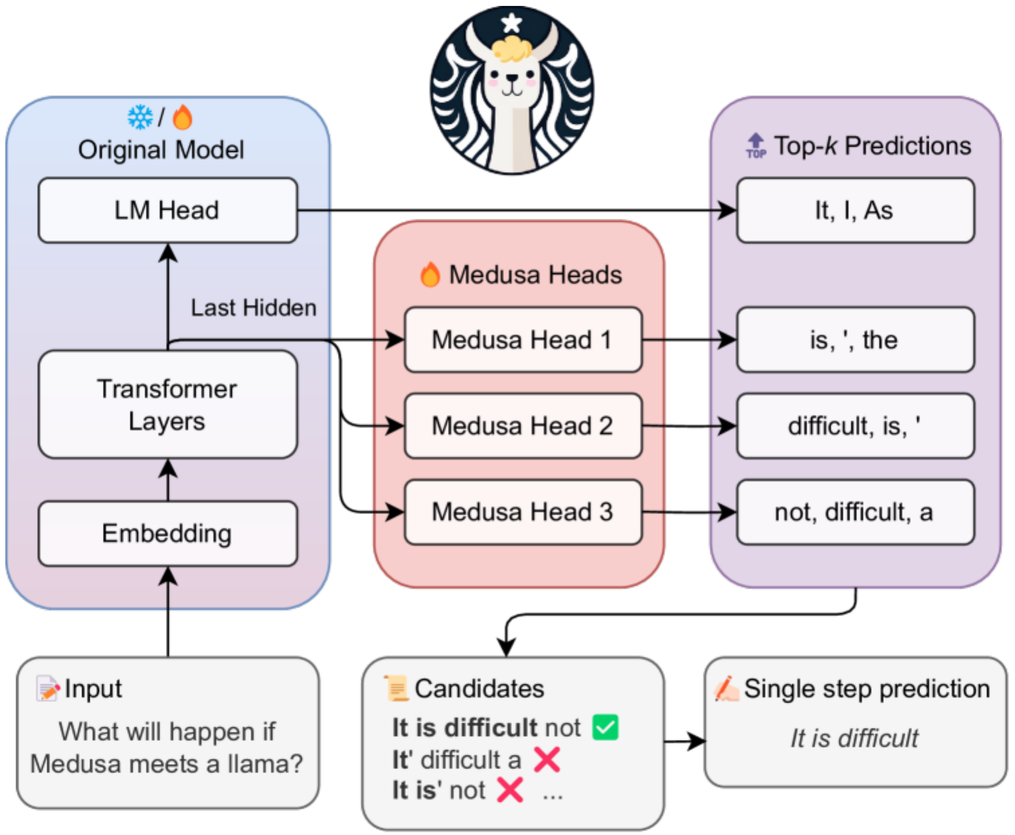
English to German translation fine tuned on WMT EnDe

Text summarization fine tuned on CCN/DM

Fast Inference from Transformers via Speculative Decoding, Levianthan 2023

# Medusa

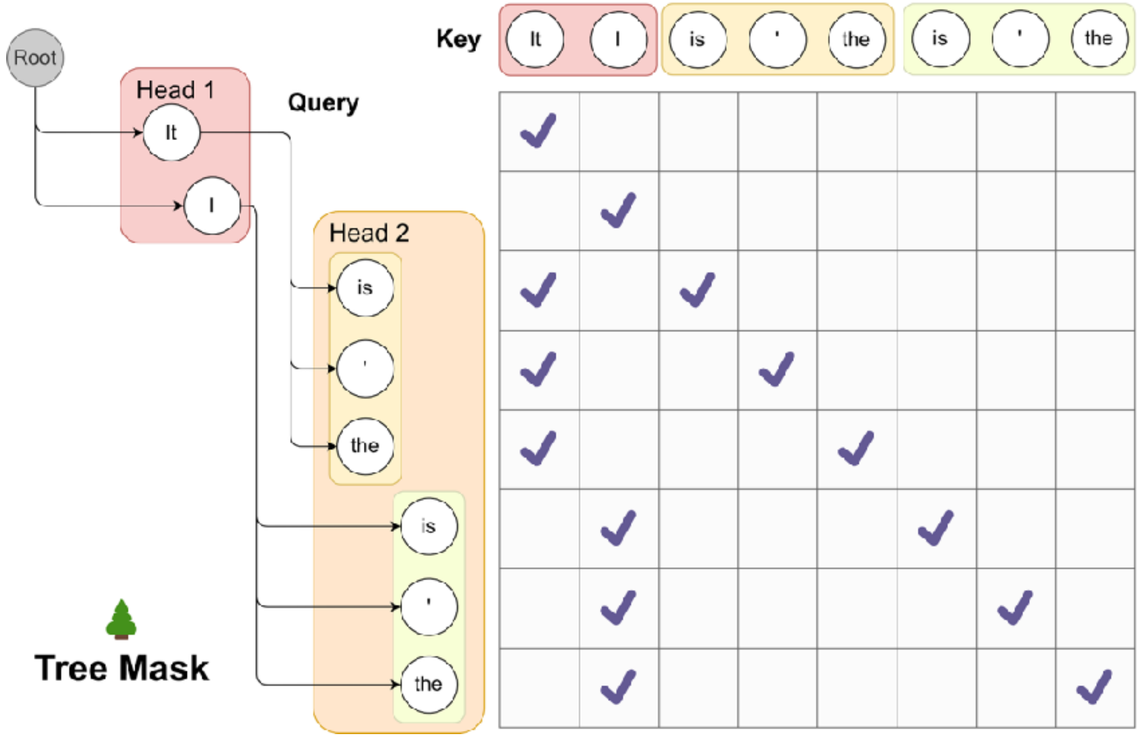
- Speculative decoding plus
  - Verifying many guesses at once
    - Tree attention
  - Use same model to produce guesses and verify
    - Medusa Head




Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Tree Attention

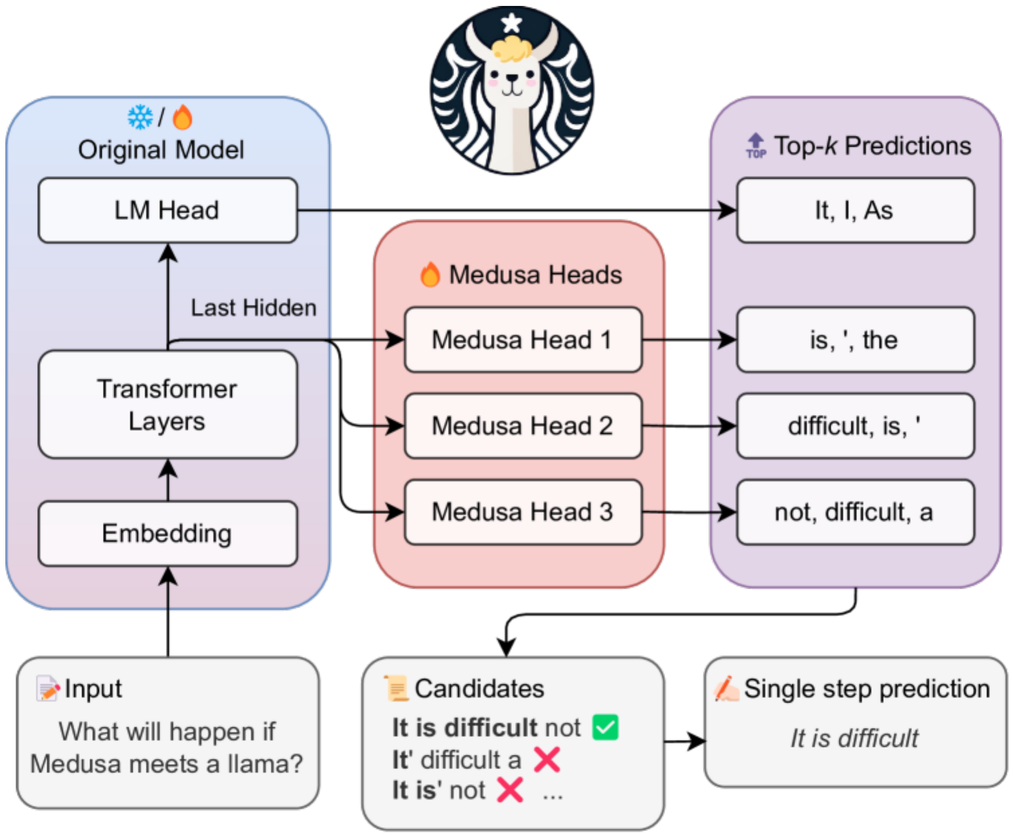
- Verifying many (tree structured ) guesses at once
  - Example:
    - It is, It', it the, I is, I', I the
- Mask attention to go back along tree
  - Tree Mask
  - Use FlexAttention




Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

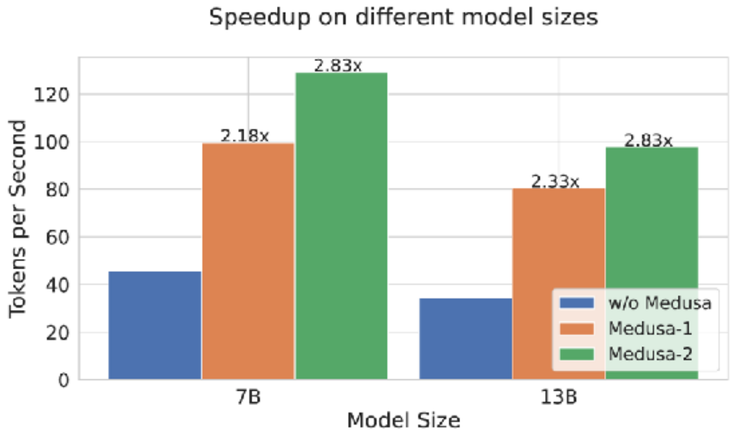
# Medusa Heads

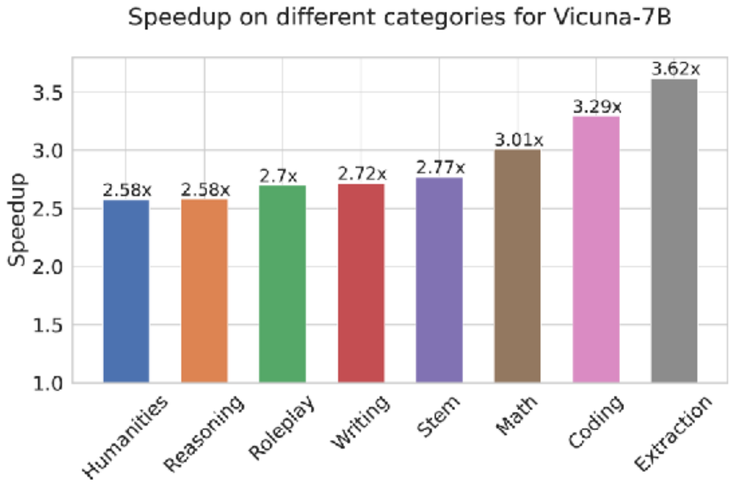
- K heads $q^{(1)} \ldots q^{(k)}$ produce independent probability of k-th next word
  - Chose top $s_1 \ldots s_k$ per node
  - **Try all combinations**




Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Medusa - Results



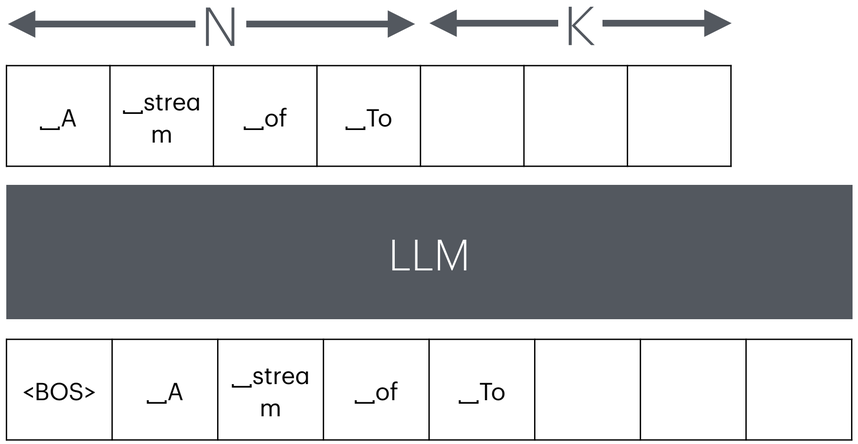




Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Speculative Decoding vs KV-Cache

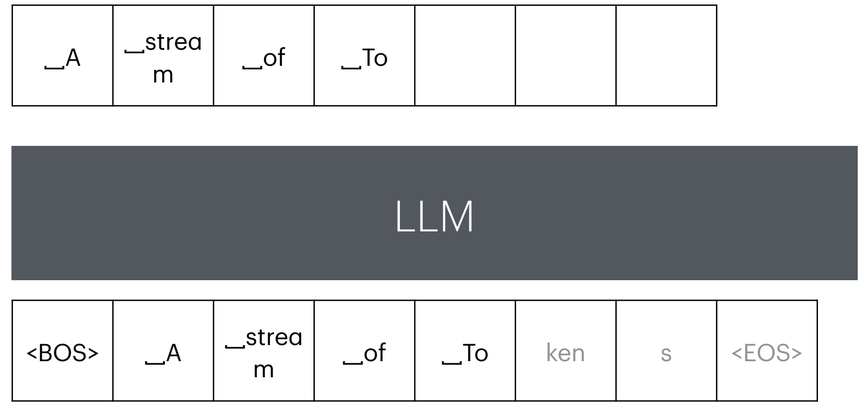
## Without KV-Cache

### Generation: K x Single token




$$K \times O(N^2) = O(KN^2)$$

### Generation: K tokens



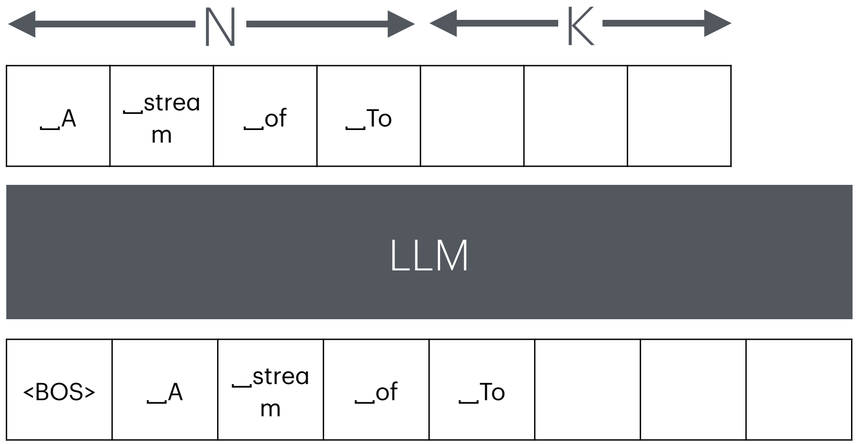

$$O((N+K)^2) \approx O(N^2) + O(NK)$$

Fast Inference from Transformers via Speculative Decoding, Levianthan 2023
Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Speculative Decoding vs KV-Cache

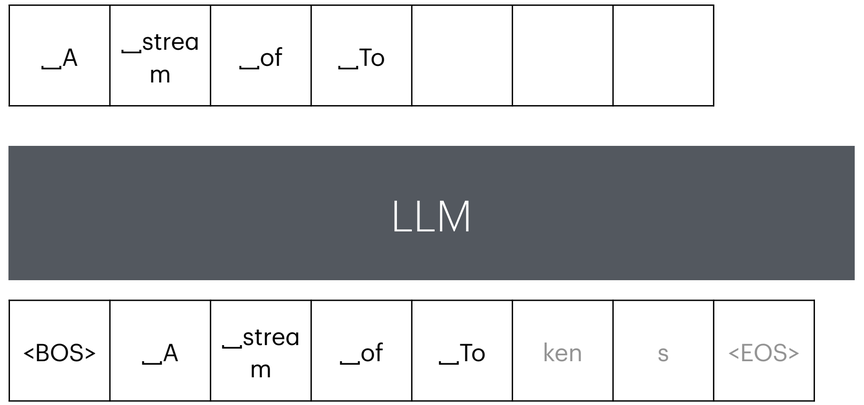
## With KV-Cache

### Generation: K x Single token




$$K \times O(N) = O(KN)$$

### Generation: K tokens



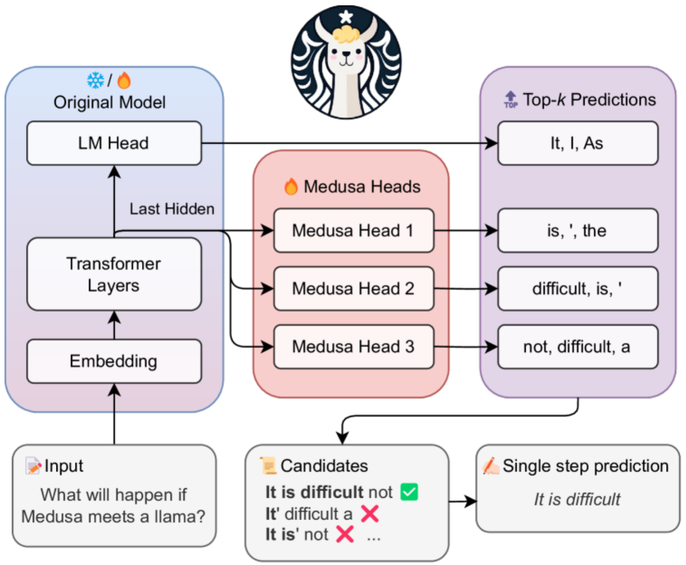

$$O(KN)$$

Gains: K vs 1 forward calls

Fast Inference from Transformers via Speculative Decoding, Levianthan 2023
Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Speculative decoding

- Beautiful technical insights
- Gains diminish with KV-Cache
- Quite tricky to implement for batched inference




**Algorithm 1** SpeculativeDecodingStep

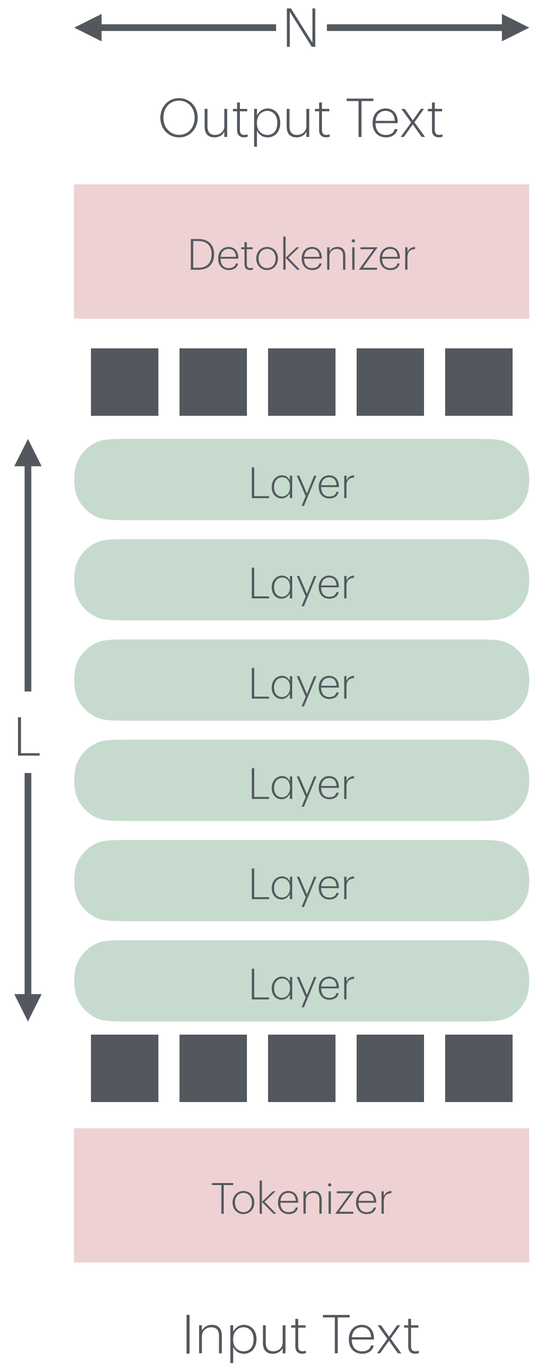
**Inputs:** $M_p, M_q, prefix$.
▷ Sample $\gamma$ guesses $x_{1,\ldots,\gamma}$ from $M_q$ autoregressively.
**for** $i = 1$ **to** $\gamma$ **do**
  $q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$
  $x_i \sim q_i(x)$
**end for**
▷ Run $M_p$ in parallel.
$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$
  $M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$
▷ Determine the number of accepted guesses $n$.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from $M_p$ if needed.
$p'(x) \leftarrow p_{n+1}(x)$
**if** $n < \gamma$ **then**
  $p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
**end if**
▷ Return one token from $M_p$, and $n$ tokens from $M_q$.
$t \sim p'(x)$
**return** $prefix + [x_1, \ldots, x_n, t]$

Fast Inference from Transformers via Speculative Decoding, Levianthan 2023
Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai etal 2024

# Training and Generation

## Paged Attention

| | Training | Training - Checkpointi | Generation | Paged Attention | Speculative decoding |
|---|---|---|---|---|---|
| Peak Memory | O(NL) | $O(NL^{½})$ | O(N) | O(NL) | O(NL) |
| Runtime | $O(N^2L)$ | $O(2\ N^2L)$ | $O(N^3L)$ | $O(N^2L)$ | $O(N^2L)$ |
| # forward | 1 | 1 | N | N | N / α |



# References


- [1] Fast Inference from Transformers via Speculative Decoding, Levianthan et al 2023. (link)
- [2] Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads, Cai et al 2024 (link)