NEURAL INFORMATION PROCESSING SYSTEMS

ByteDance

UCD DUBLIN

Insight Centre for Data Analytics

PaDeLLM-NER: Parallel Decoding in Large Language Models for Named Entity Recognition

Jinghui Lu*

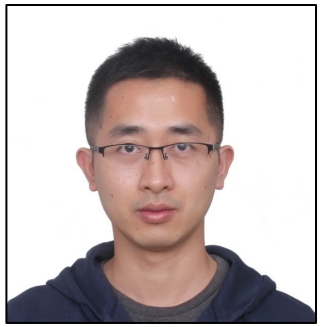
Ziwei Yang*

Yanjie Wang*

Xuejing Liu

Brian Mac Namee

Can Huang



The Bottleneck for LLM Inference Latency Lies in the Decoding Phase.

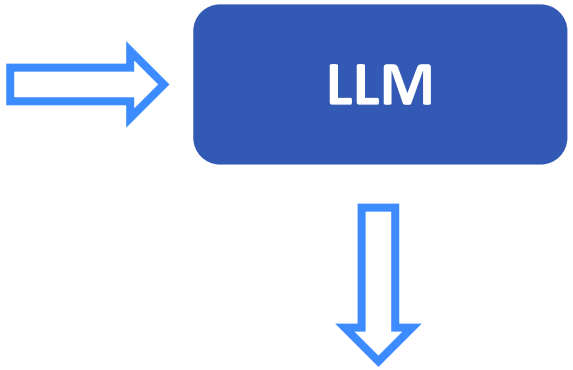
	GPU	Prefilling	Decoding
Qwen2-7B	A100	2708.64 token/s	60.63 token/s
Llama3.1-8B	A100	1947.76 token/s	64.11 token/s

Inference with LLMs/MLLMs using the vLLM[1] framework shows that the decoding phase is the main bottleneck.

[1] https://github.com/vllm-project/vllm

Autoregressive NER Produces Long Sequences, Slowing Decoding

Japan, co-hosts of the World Cup in 2002 with Korea and ranked 20th in the world by FIFA, are favourites to regain their title here.

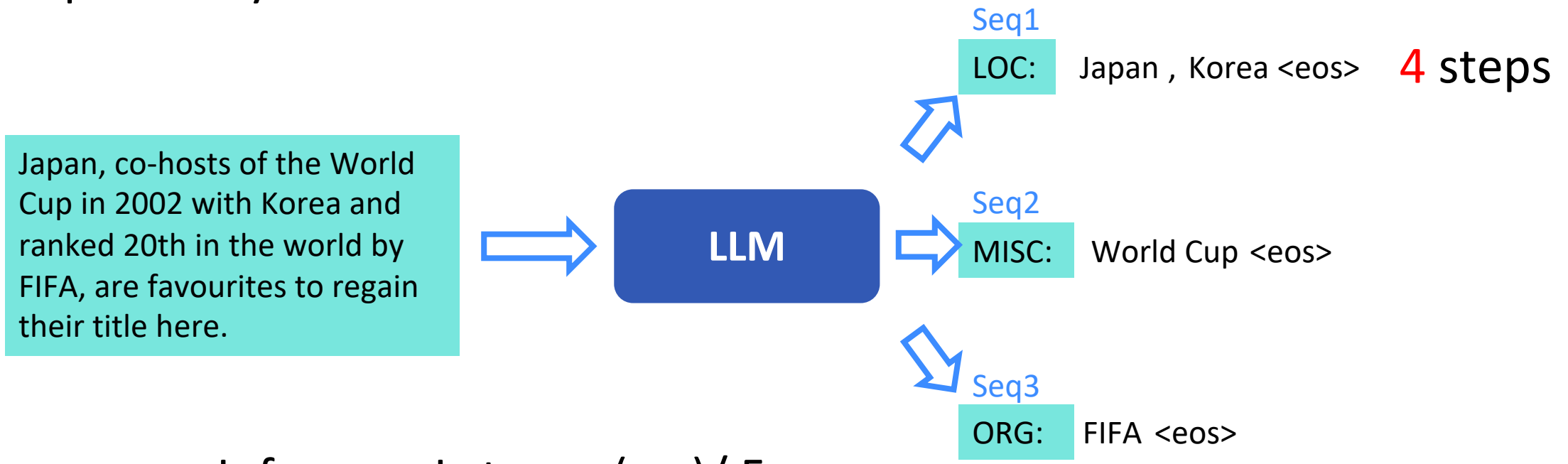


((LOC : Japan) , (LOC : Korea) , (MISC : World Cup) , (ORG : FIFA)) <eos>

26 steps

Can we speedup decoding phase by shortening output sequence length?

Preliminary Experiment---Breakdown Sequence by Labels.

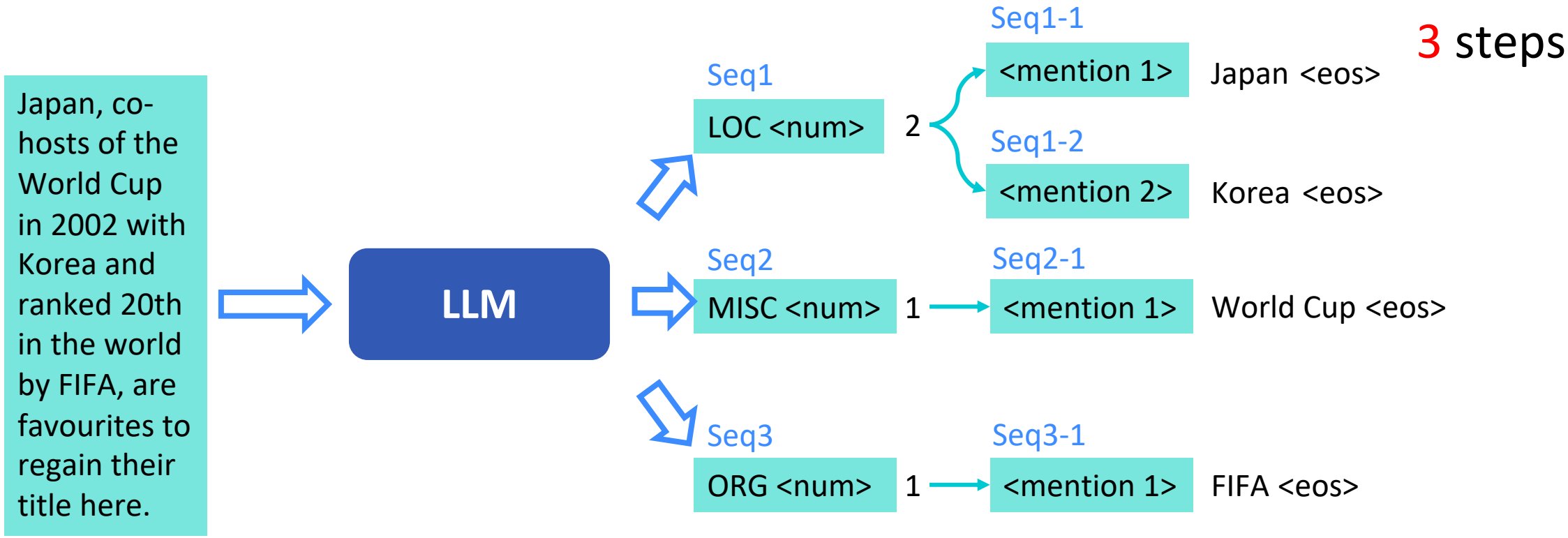


Inference Latency (ms)/ F-score

	ACE05	CoNLL03	GENIA
OneStep	**386.93** /80.98	**272.22** /91.36	**513.56** /76.27
$\text{AutoReg}_{\text{Aug}}$	944.90 /83.04	992.70 /93.08	1515.35 /70.16
$\text{AutoReg}_{\text{Struct}}$	1293.87 /82.99	753.36 /91.87	1266.34 /77.90

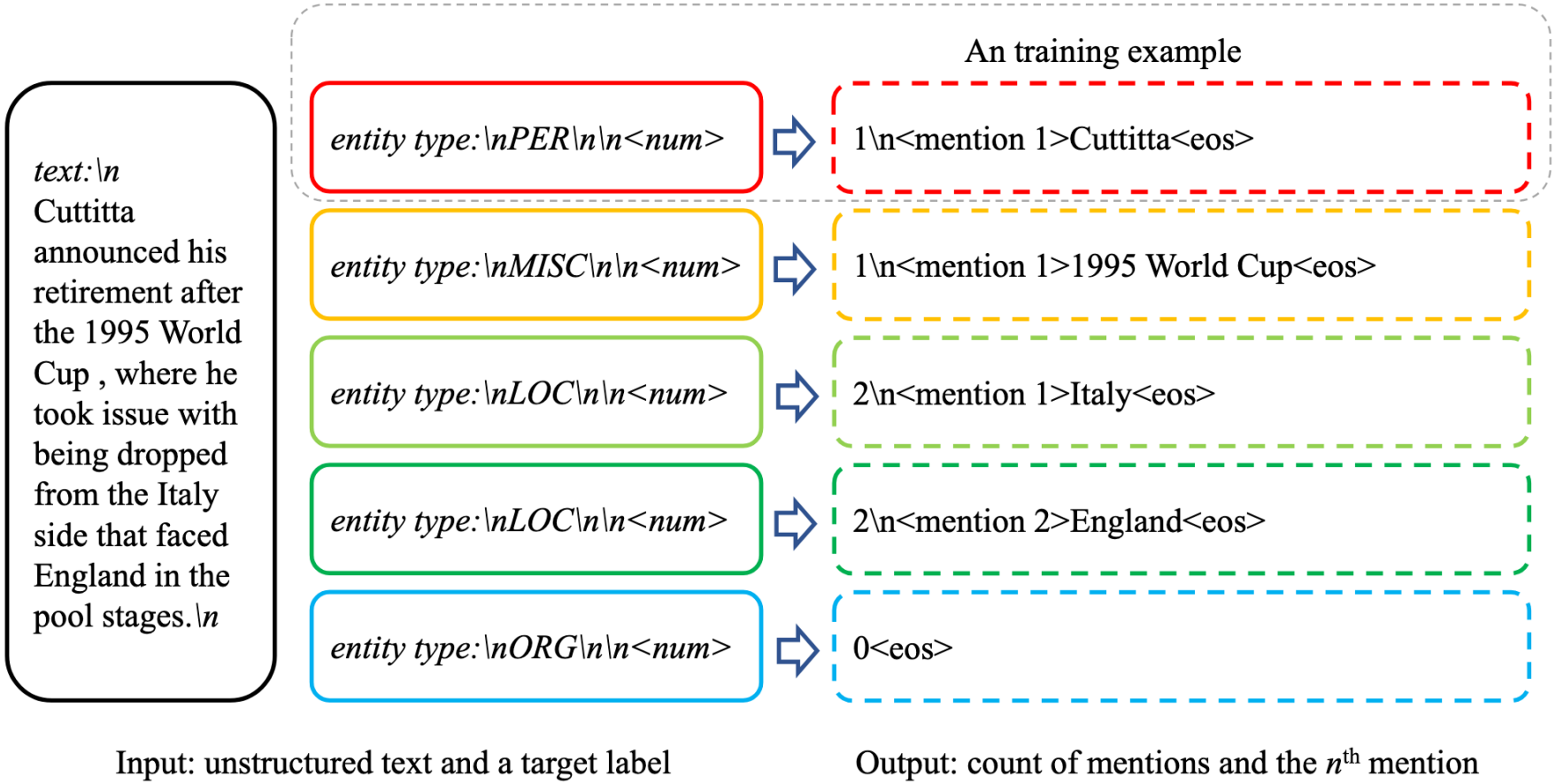
Significantly reduced latency with competitive accuracy

PaDeLLM-NER: Further Shorten Sequence Length



Text in is appended rather than generated by LLMs.

PaDeLLM-NER: How to train



Train LLMs to:

- Predict the number of mentions given the label.
- Predict the specific mention based on the mention index.

Experimental Results on Inference Latency (milliseconds)

	ACE05	CoNLL03	GENIA	Avg.
PaDeLLM	**255.53**	**229.74**	**316.90**	**267.39**
OneStep	386.93	272.22	513.56	390.90
$\text{AutoReg}_{\text{Aug}}$	944.90	992.70	1,515.35	1,150.98
$\text{AutoReg}_{\text{Struct}}$	1,293.87	753.36	1,266.34	1,104.52

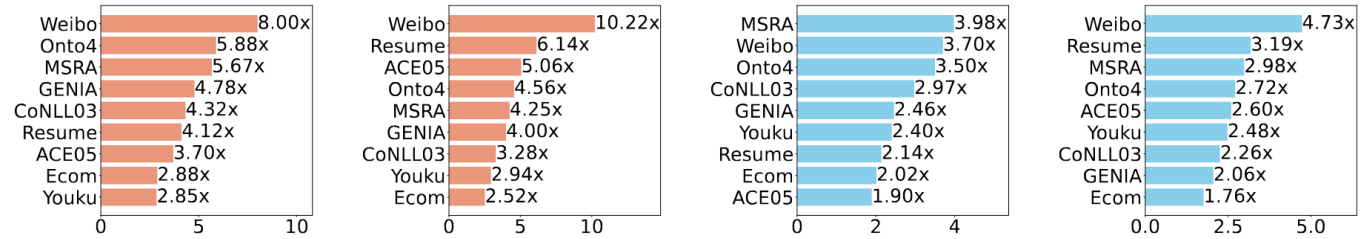
	Weibo	MSRA	Onto4	Resume	Youku	Ecom	Avg.
PaDeLLM	**159.57**	**143.47**	**171.67**	**238.27**	**203.63**	**293.40**	201.67
$\text{AutoReg}_{\text{Aug}}$	1,276.32	812.78	1,009.68	982.39	579.99	845.42	917.76
$\text{AutoReg}_{\text{Struct}}$	1,630.62	609.34	783.28	1,462.56	598.59	738.20	970.43

Prediction Quality SFT & Zero-shot (Micro F-score)

	CoNLL03	ACE05	GENIA	Weibo	MSRA	Onto4	Resume	Youku	Ecom	Avg.
$\text{AutoReg}_{\text{Aug}}$	**93.08**	83.04	70.16	59.04	95.5	79.20	95.80	86.07	76.02	81.99
$\text{AutoReg}_{\text{Struct}}$	91.87	82.99	**77.90**	56.07	90.92	**80.97**	**95.74**	86.85	81.57	82.76
PaDeLLM	92.52	**85.02**	77.66	**67.36**	**95.03**	80.81	94.98	**87.91**	**81.85**	**84.79**

Model	AI	Literature	Music	Politics	Science	Movie	Restaurant	Avg.
SOTA								
GoLLIE-7B	59.1	62.7	67.8	57.2	67	63	43.4	60.02
UniNER-7B	53.5	59.7	65	60.8	61.1	42.4	31.7	53.45
GLiNER-L	57.2	64.4	69.6	**72.6**	62.6	57.2	42.9	60.92
GNER-LLaMA-7B	**63.1**	**68.2**	**75.7**	69.4	**69.9**	**68.6**	**47.5**	**66.05**
Ours								
PaDeLLM-NER-7B	60.7	66.1	67.6	68.1	64.4	61.3	43.6	61.68

Speedup & Sequence Length



(a) AR_{Aug} vs. $PDLM_{Multi}$ (b) AR_{Struct} vs. $PDLM_{Multi}$ (c) AR_{Aug} vs. $PDLM_{Batch}$ (d) AR_{Struct} vs. $PDLM_{Batch}$

Figure 3: Speedup of PaDeLLM-NER compared to Autoregressive methods.

	CoNLL 03	ACE05	GENIA	Weibo	MSRA	Onto4	Resume	Youku	Ecom	Avg.
$AutoReg_{Aug}$	33.85	37.10	60.50	45.02	27.42	35.90	30.39	18.21	31.50	35.54
$AutoReg_{Struct}$	28.36	49.95	49.03	62.45	18.97	25.53	53.02	18.56	22.51	36.48
PaDeLLM	**6.54**	**8.29**	**10.05**	**2.19**	**2.23**	**2.68**	**4.87**	**3.66**	**3.27**	**4.86**

- **1.76 to 10.22x** faster inference across English and Chinese NER tasks.
- Average sequence length is reduced to around **13%** of that produced by conventional autoregressive methods.
- Maintain prediction quality.

QA & Collaboration

Git: https://github.com/GeorgeLuImmortal/PaDeLLM_NER
We are hiring interns!
Email: lujinghui@bytedance.com