Google

Multilingual and Fully Non-Autoregressive ASR with Large Language Model Fusion: A Comprehensive Study

Presenter: Tongzhou Chen

19 April, 2024

Authors

W. Ronny Huang (wrh@google.com), Cyril Allauzen (allauzen@google.com), **Tongzhou Chen (tongzhou@google.com)**, Kilol Gupta, Ke Hu, James Qin, Yu Zhang, Yongqiang Wang, Shuo-Yiin Chang, Tara N. Sainath

Overview

- Study the impact of Large Language Models in multilingual non-autoregressive ASR models on long-form data
  - **3.6%** gain for YouTube Captions
  - **10.7%** gain for FLEURS across languages
- Perform comprehensive ablation study of Large Language Models including
  - Model size
  - Number of hypotheses
  - Segment length
  - Context length
  - Vocabulary size
  - Comparison with shallow fusion

Speech Model

Universal Speech Model (USM)

- Architecture
  - 2 billion parameters
  - 32 layers of Conformers with dimension 1536
  - Chunk-wise attention
  - 16384 wordpiece vocabulary
  - CTC decoder, non-autoregressive, parallel inference
- Training
  - Trained with 12M hrs of unlabeled audio and 28B sentences of text data, along with 110K hrs of supervised and 100K hrs of semi-supervised audio
  - Multilingual with more than 100 languages

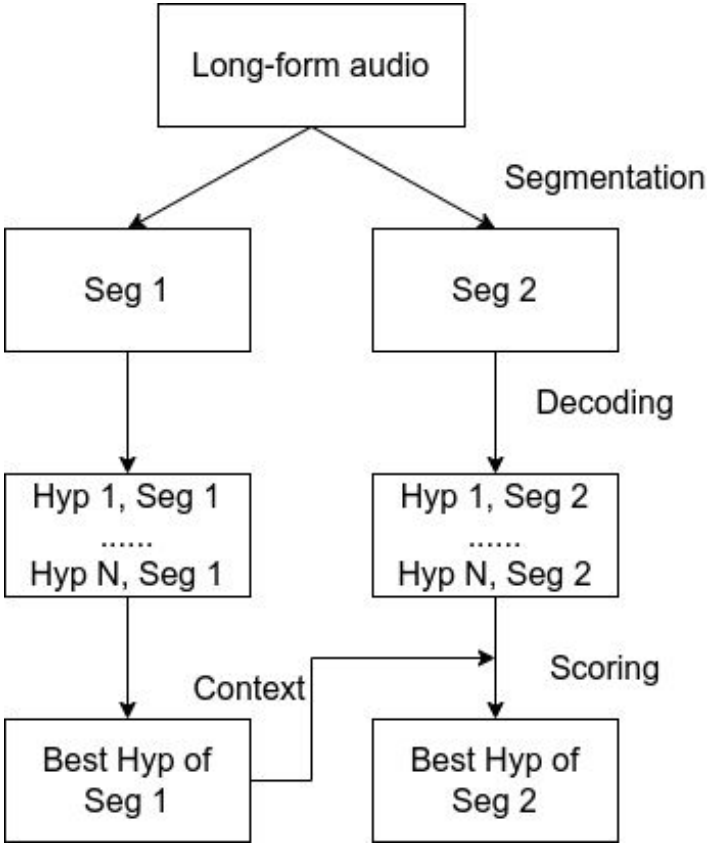
Language Model

Pathways Language Model 2 (PaLM 2)

- Trained on multilingual data sources including web documents, books, code, mathematics, and conversational data with hundreds of billions tokens
- Transformer-based, decoder only model
- 256K wordpiece vocabulary
- Model Size 128M to 340B

Inference and Scoring

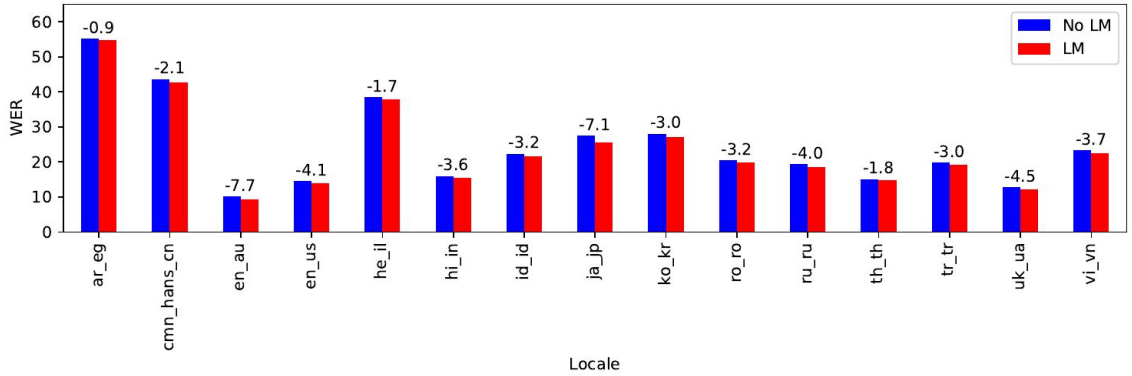
- To fit into memory, we chunk the long-form audio into fixed-length segments
- First-pass decoding is parallelizable
- Second-pass rescoring is done within each segment, using the one-best hypotheses from the previous segments as the context
- $\log P_{Final}(Y|X) = \log P_{CTC}(Y|X) + \lambda \log P_{LM}(Y)$
  - λ is the LM scoring weight, can be found by grid search

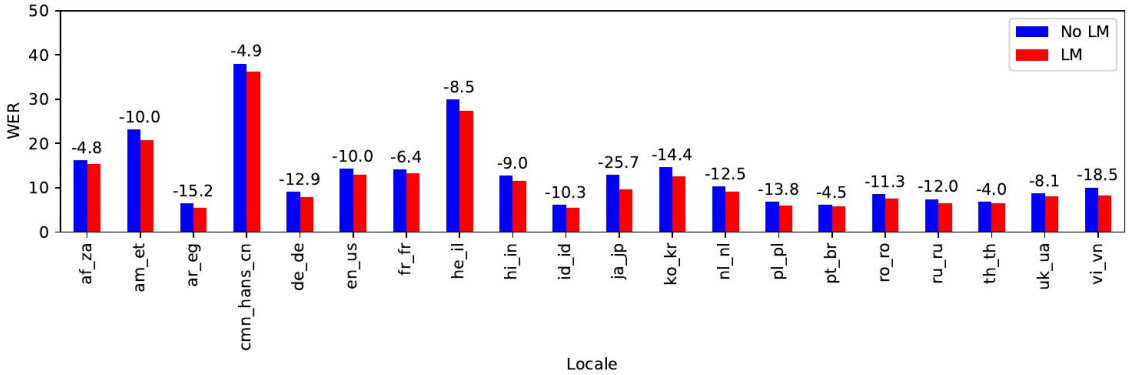


Results on All Languages

- We present our results on YouTube Captioning as well as FLEUR Test sets
  - YouTube: 16 languages, 50~80 utterances, average length 15 minutes
  - FLEUR: 20 languages, 600~900 utterances, average length 1~2 minutes
- Default scoring setups
  - 1B parameters PaLM 2
  - N-best list size 16 in each segment
  - 8 seconds segment length (~12 words)
  - One-best from 2 prior segments as context (16 seconds or 25 words)
  - 256K wordpiece vocabulary
  - Uniform LM weight λ=0.3 across all languages

Results on All Languages





Top: Youtube

- **4.1%** gain in en_us
- **3.6%** gain in other languages

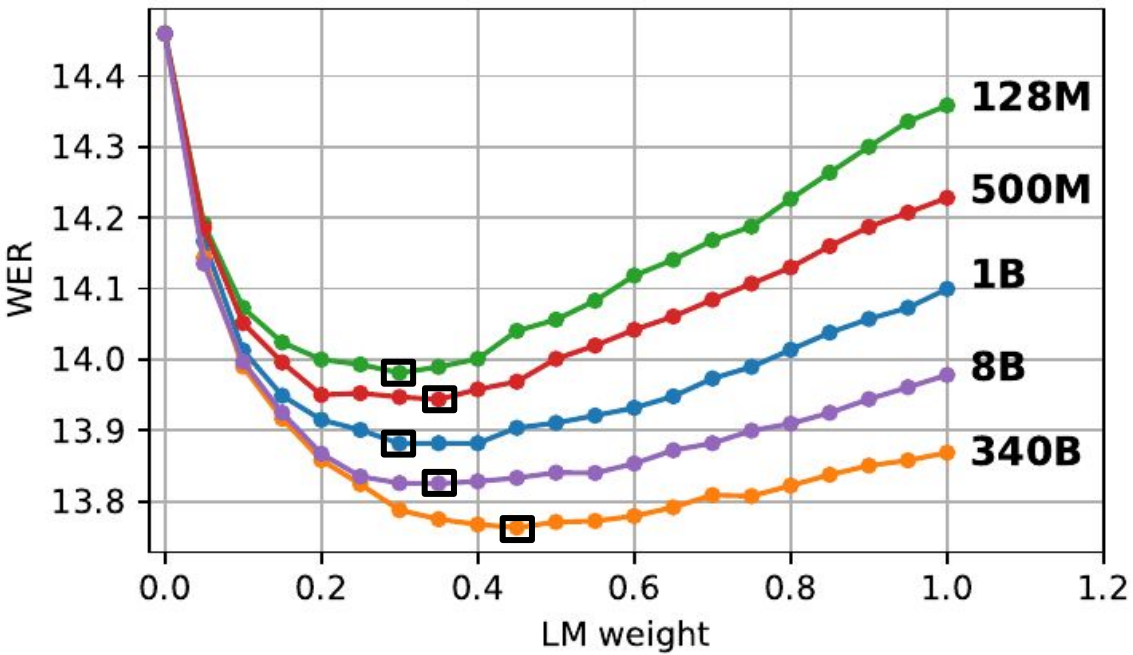
Bottom: FLEUR

- **10.0%** gain in en_us
- **10.8%** gain in other languages

Ablation Study

- We perform ablation study on en_us YouTube set
- Each time we vary one parameter in the default setups below and keep all the other parameters fixed
  - 1B parameters PaLM 2
  - N-best list size 16 in each segment
  - 8 seconds segment length (~12 words)
  - One-best from 2 prior segments as context (16 seconds or 25 words)
  - 256K wordpiece vocabulary

Ablation Study: Model Size and LM Weight

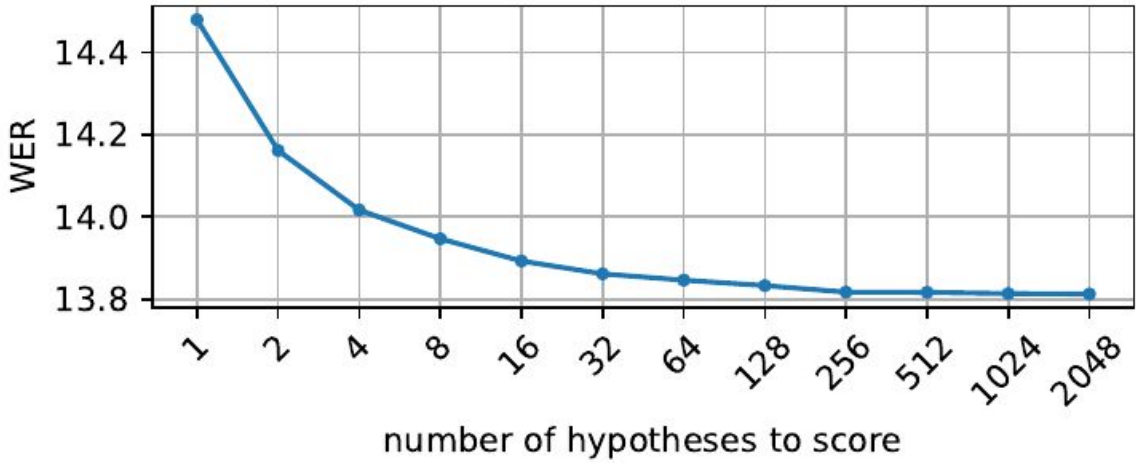


WER improves as LM size grows

Optimal LM weight increases slightly with model size

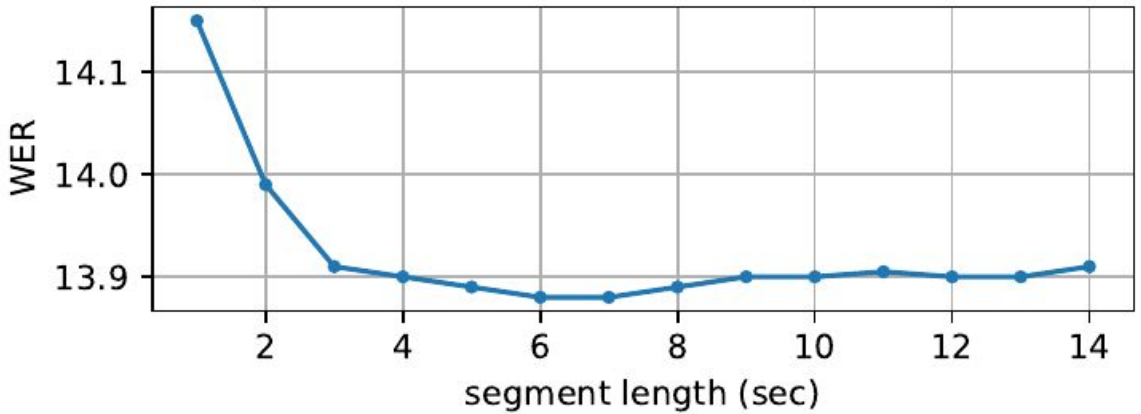
Larger models are less sensitive to LM weight changes

Ablation Study: Number of Hypotheses



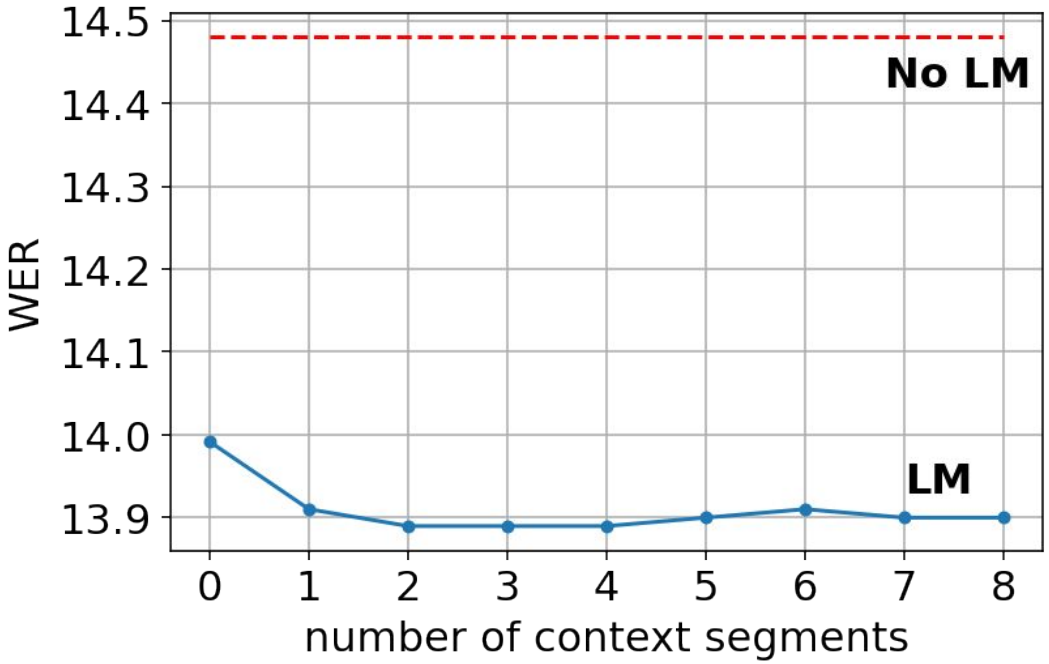
WER decreases as the n-best size expands
Dense lattice has potential, allowing the LLM to continue improving

Ablation Study: Segment Length



CTC is robust to premature segmentation
WER stabilized when segment length is beyond 3 seconds

Ablation Study: Context Length



Carrying over context from previous segments can help
Adding context beyond 4 segments (32s) offers limited improvement

Ablation Study: Vocabulary Size

Embedding and softmax layers take up 1/3 of 256K vocab 1B PaLM 2 params

Can we reduce that?

We fine-tuned the 1B model with 32K vocab, the model size was reduced by 20%

LM Vocab Size	WER
256K	13.9
32K	13.9

Smaller vocabulary can save computation while retaining performance

Ablation Study: Comparison with Shallow Fusion

Per-segment Scoring: LM acts at the token level, N_{avg_tokens} x N_{hyps} computations

Shallow fusion: LM acts at the frame level, N_{frames} x N_{hyps} computations

- On average 1 tokens corresponds to 4 frames, we skip scoring if the frame has more than 0.9 probability to be blank
- Retrained AM with matched vocabulary as LM

Scoring Type	WER
Per-segment Scoring	13.9
Shallow Fusion	13.7

Shallow fusion can further improve the WER in non-latency-critical scenarios

Conclusion

- We improved the performance of a non-autoregressive multilingual CTC system by per-segment LM scoring, showing 3.6% gain for YouTube Captions and 10.7% gain for FLEURS across languages
- We conducted a thorough examination of system parameters, contributing to a better understanding of their impacts on ASR performance.

Thanks!