Microsoft

TVM @ Microsoft

Jon Soifer, Software Engineer, @jonso/@soiferj
Minjia Zhang, Researcher, @zhangninja

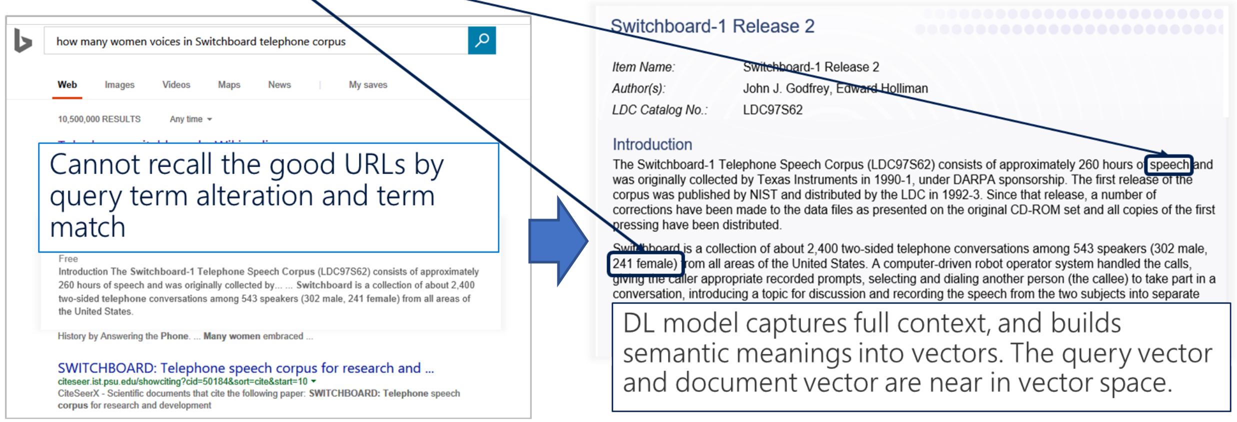
Our team: Zehua Hu, Menghao Li, Jeffrey Zhu , Elton Zheng, Mingqin Li, Jason Li, Yuxiong He

Microsoft AI and Research

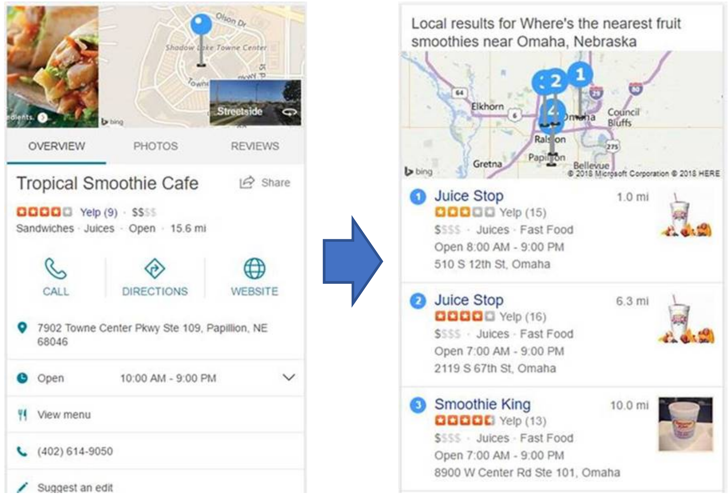
Deep Learning at Microsoft

Web Search

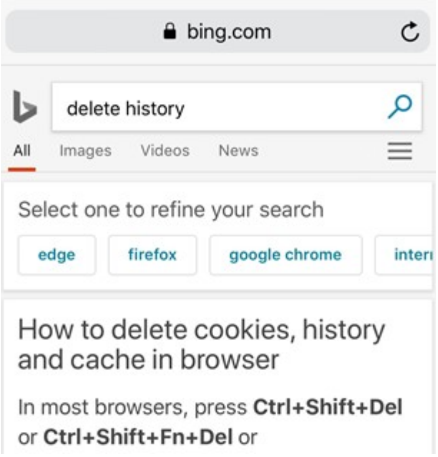
Query: {how many women voices in Switchboard telephone corpus }



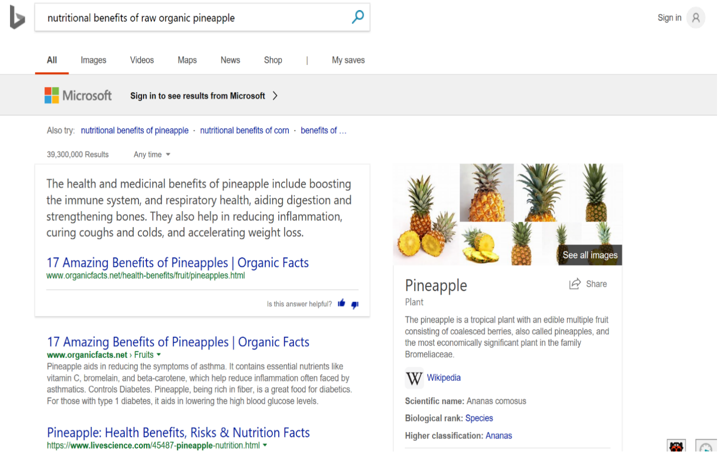
Entity Search



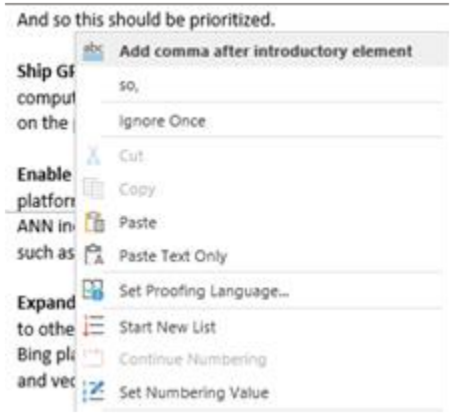
Conversational Search



QnA at Web Scale

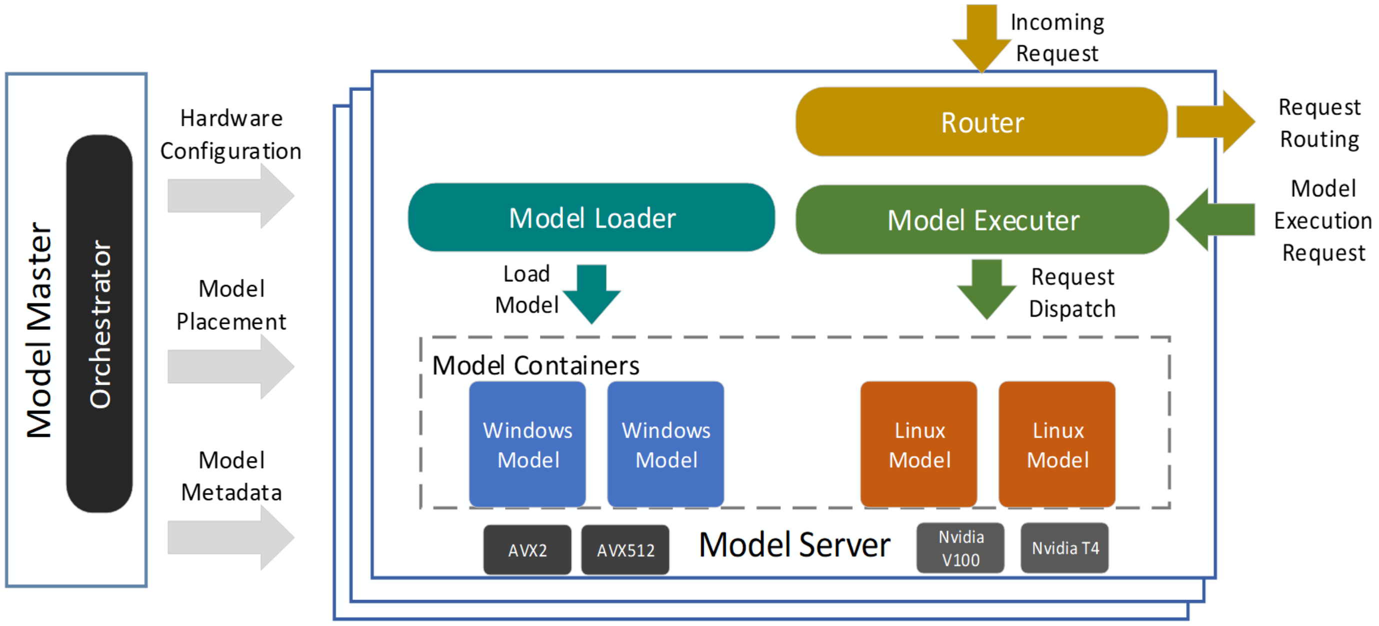


Grammar Checking



Deep Learning Inference Service

- Serves Bing, Office, and Cortana
- Large scale
  - Millions of model inferences per second
  - Hundreds of models
  - Tens of thousands of servers
  - Forty data centers worldwide
- Variety of serving requirements
  - TensorFlow, PyTorch
  - Windows, Linux
  - CPU, GPU
- Strict latency requirements
  - Often single-digit milliseconds



Model Optimization Example

- Large-scale BERT[1] for Bing web ranking
  - 1 million queries per second
- TensorFlow latency and throughput were unacceptable
- Hand-optimized BERT on V100 GPU
- 800x throughput increase
- Millions of dollars saved
- Over a month of dev time
- Blog post
  - https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/

1. Devlin et. al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", https://arxiv.org/pdf/1810.04805.pdf

Model Optimization Challenges

- Existing DL frameworks don’t fit our requirements
- Challenges
  - Reducing latency to a scenario-acceptable number
  - Supporting advanced models at large scale while saving cost
  - Agility to bring new optimization techniques into production
- We need new solutions to ship new and exciting models

Model Optimization Solutions

Custom Optimizations

- Rewrite models with high performance C++ library
- Customized serving runtime and performance tuning
- Example: DeepCPU, DeepGPU, TensorRT

✓ Low latency and high throughput

✓ Best utilization of hardware

✗ Low agility

Framework Integration

- Integrate custom ops with existing frameworks (e.g., TF, PyTorch)
- Replace nodes in model graphs and leverage existing framework serving engine
- Example: Customized TensorFlow, WinML

✓ Less development work

✓ Decent latency improvement

✗ Suboptimal performance

Compiler

- Graph-level optimizations
- Optimized code generation
- Cross-platform, cross-device

Can we achieve low latency, high throughput, and high agility?

Case-Study 1: Query Understanding for Bing

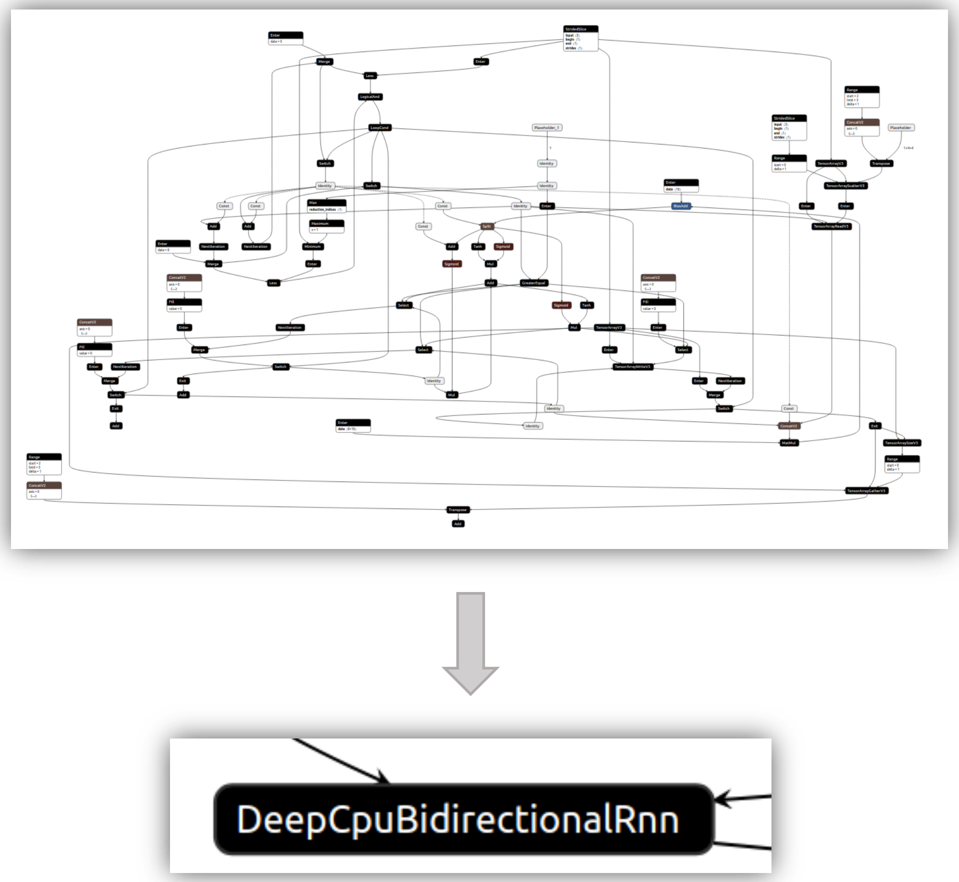
- Generate query encoding for ranking
- Model: CNN embedding + LSTM + scoring function

- Latency SLA: 35ms

- TensorFlow: 112ms on CPU
- TVM + Custom RNN: 34ms on CPU

A Hybrid Approach: TVM + DeepCPU

- DeepCPU[1] is plugged in as TVM external library
    - Automatically identify high-level TF constructs
        - Utilize TensorFlow scopes
        - Identify single- and bi-directional LSTMs
    - Rewrite Relay graph
        - Replace subgraph with a custom op node
    - 63ms -> 15ms



- CNN and the rest of graph are optimized and auto-tuned by TVM
    - 49ms -> 19ms (2.5 times speedup)

1. “DeepCPU: Serving RNN-based Deep Learning Models 10x Faster”, Zhang et. al. USENIX ATC 2018

Case-Study 2: Azure QnA Maker Service

- Azure cognitive service that creates question-and-answer bots
- Model: Distilled BERT

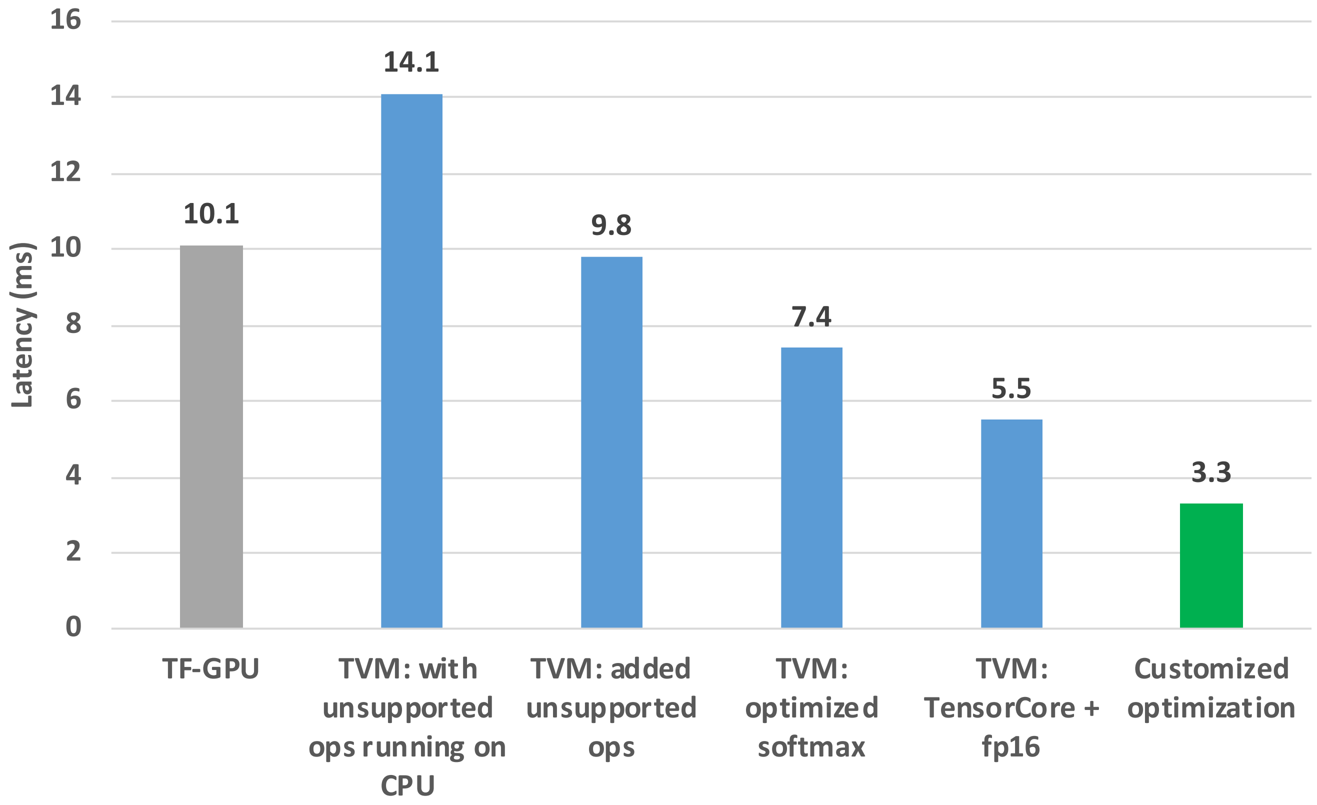
- Latency SLA: 10ms

- TensorFlow: 73ms on CPU, 10.1ms on GPU
- TVM + our improvements: 28ms on CPU, 5.5ms on GPU

Optimizing BERT with TVM on GPU

- New operators
  - OneHot, Erf, BatchMatMul with > 3 dimensions
- New softmax schedule tailored for large-vocabulary projection
- Adding support for half-precision and extended GEMM on TensorCore

- Still a gap with hand-tuned version but decent speedup over TF-GPU (46% improvement)



On Nvidia V100

Contributions to TVM

- CombineParallelDense IR pass
- Operators for TensorFlow and ONNX frontends
- Improve softmax compute and CPU schedule
  - Auto-tune softmax schedule
  - > 80% improvement on 16 cores
- Fix schedule_extern to prevent fusion of external ops
  - ~50% improvement when using external libraries on CPU
- Support MKL and cuBLAS for BatchMatMul
- Windows support and fixes

We’re hiring!

Our Experience with TVM

- Vibrant, supportive, and open community
- Developer-friendly
- Emphasis on innovating and experimenting with new techniques
- Performance improvement over popular DL frameworks
- Several models shipped to production

- We are looking forward to contributing and trying new features from the community!
    - Dynamic shapes, TensorFlow dynamic RNN, bring-your-own-codegen

Thank you!