Microsoft

GTC 2019

Mar 2019

San Jose, CA

Microsoft AI & Research

Microsoft

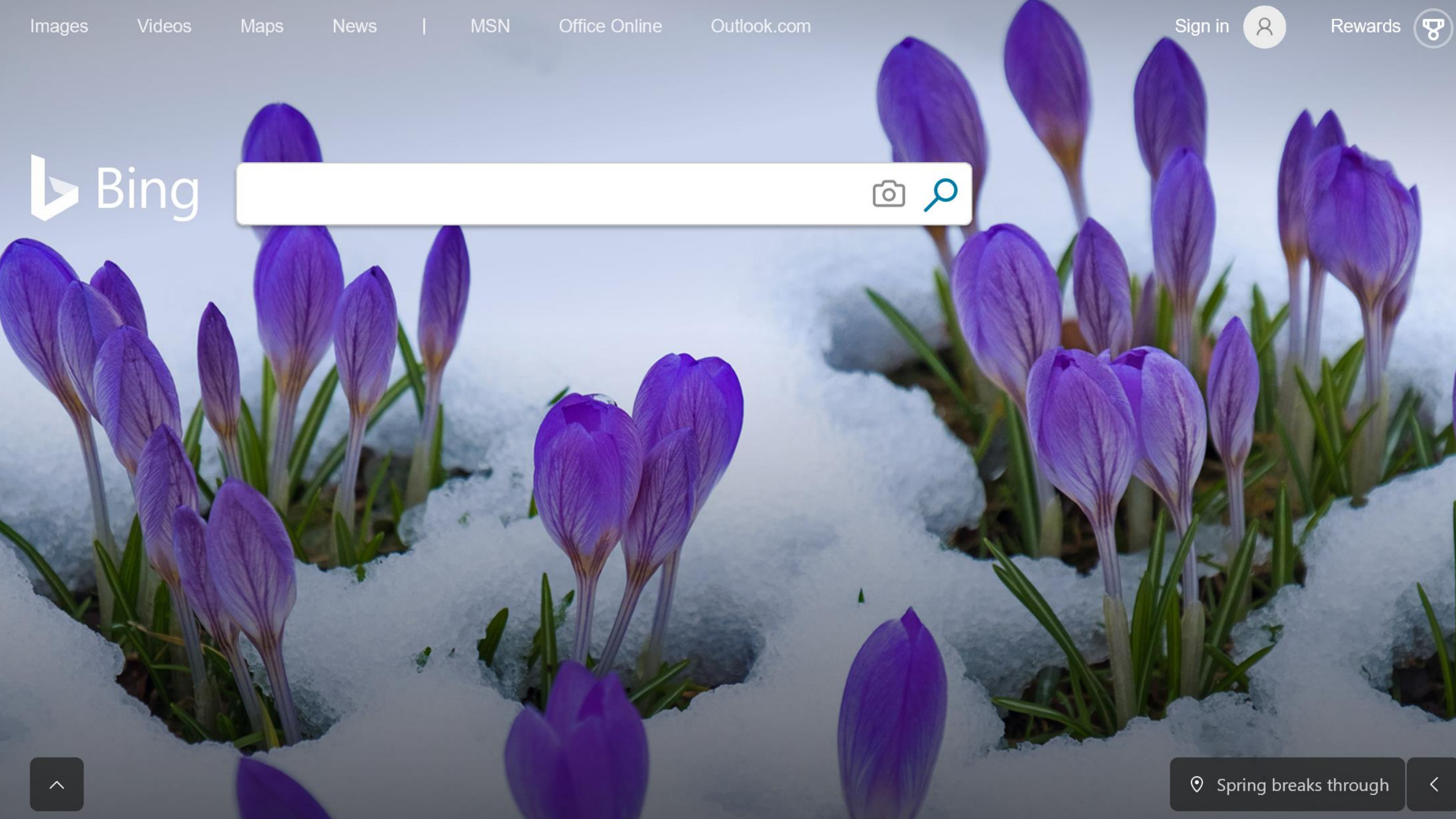
Using Deep Learning to Transform Internet Scale Web Searches

Guhan Suriyanarayanan, Principal Program Manager
guhans@microsoft.com

Adi Oltean, Principal Software Engineer
adi.oltean@microsoft.com

Agenda

- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

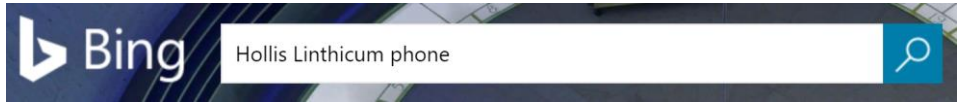

Images
Videos
Maps
News
|
MSN
Office Online
Outlook.com
Sign in
Rewards
Bing
Spring breaks through

International PC share growth – Bing Network

	2015	2018
United States	31.4%	**34.7%**
United Kingdom	17.6%	**21%**
France	11.3%	**15.7%**
Canada	15.2%	**20.8%**
Australia	10.9%	**15.4%**
Germany	7.5%	**20.7%**

Source: Bing Network PC Share. comScore qSearch, (custom), Dec 2015 and Sep 2018

The Evolution of Search



Keyword based Search

Traditional IR	
AUTB streams	Inverted index

Natural Language Search

User Engagement	
User clicks, metawords	Inverted index

Voice, Vision Context-Based Search

AI	
Deep Learning	Vectors

Intelligent Answers

what clothing do people wear in silicon valley

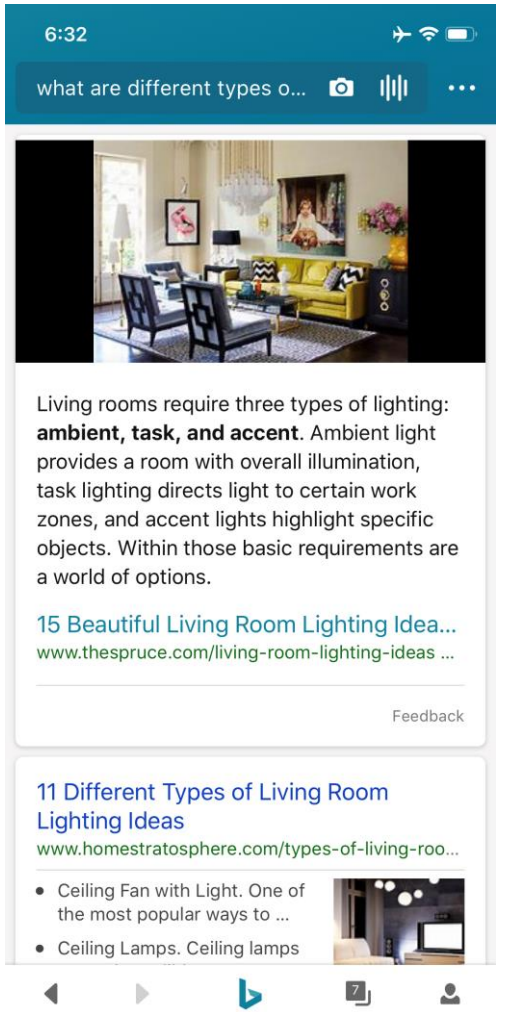
All | Images | Videos | Maps | News | Shopping | My saves

18,900,000 Results Any time

Silicon Valley, of course, is known for its casual dress, which means **t-shirts, jeans and sneakers**. But don't be fooled, techies care a lot more about fashion than they let on. Or put another way, there's a lot of code in the Silicon Valley dress code. In fact, engineer Alexey Komissarouk boasted he could tell if people were in tech and what they did by just looking at their dress.

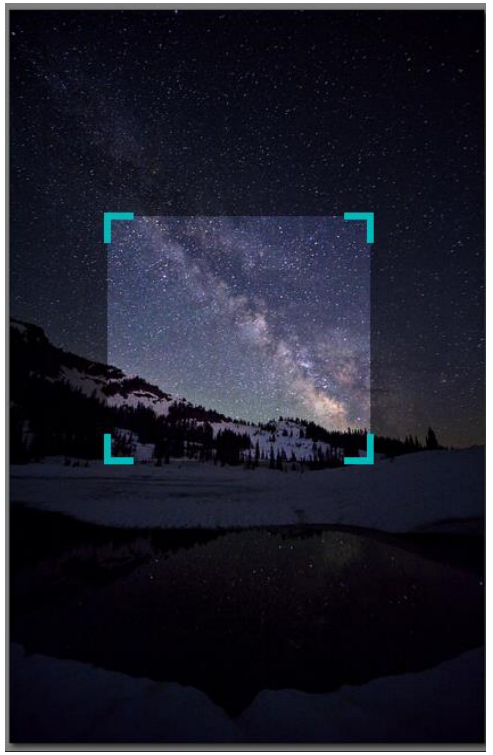
Silicon Valley has a dress code? You better believe it
www.marketplace.org/2014/01/28/tech/silicon-valley-has-dress-code-you-better-believe-it

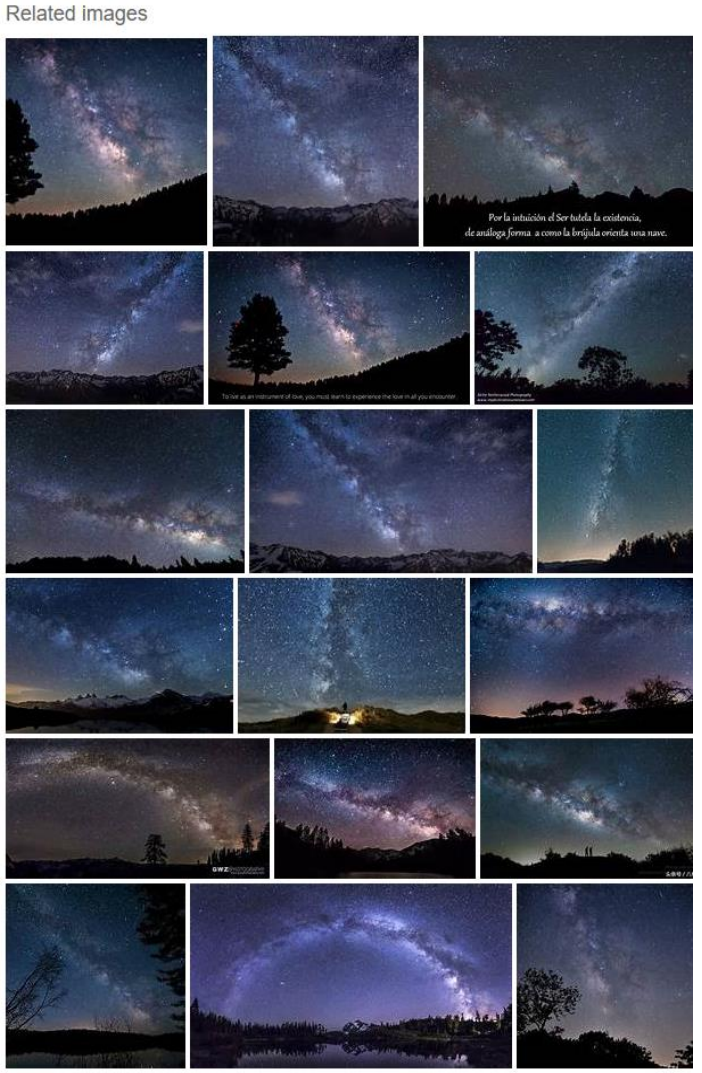
Is this answer helpful?



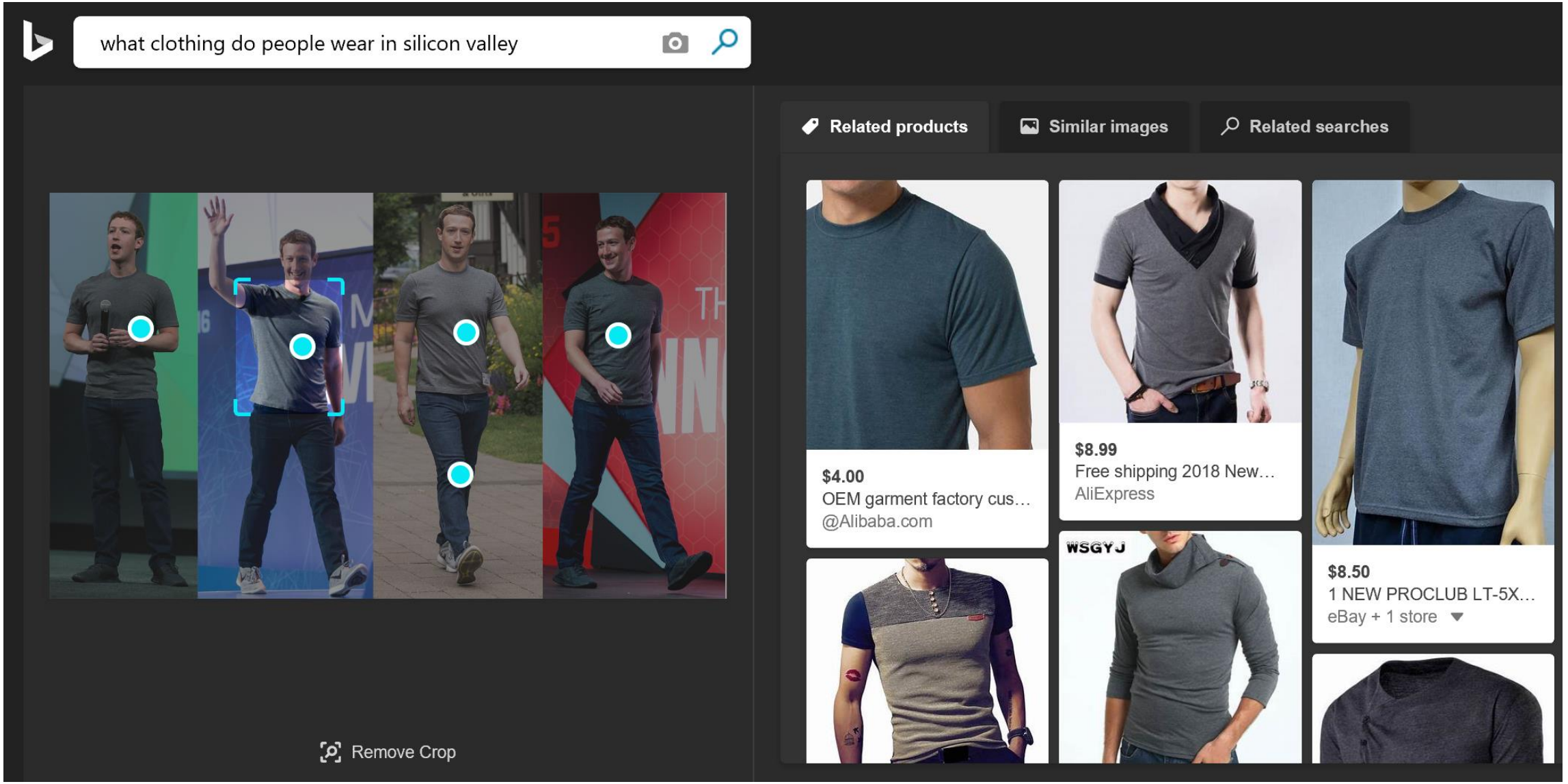
Visual Search

- Image similarity match
- Deep Learning vector encoding
- Approximate nearest neighborhood search
- Search in several hundred million images in a few milliseconds
- Significant improvement to image recall

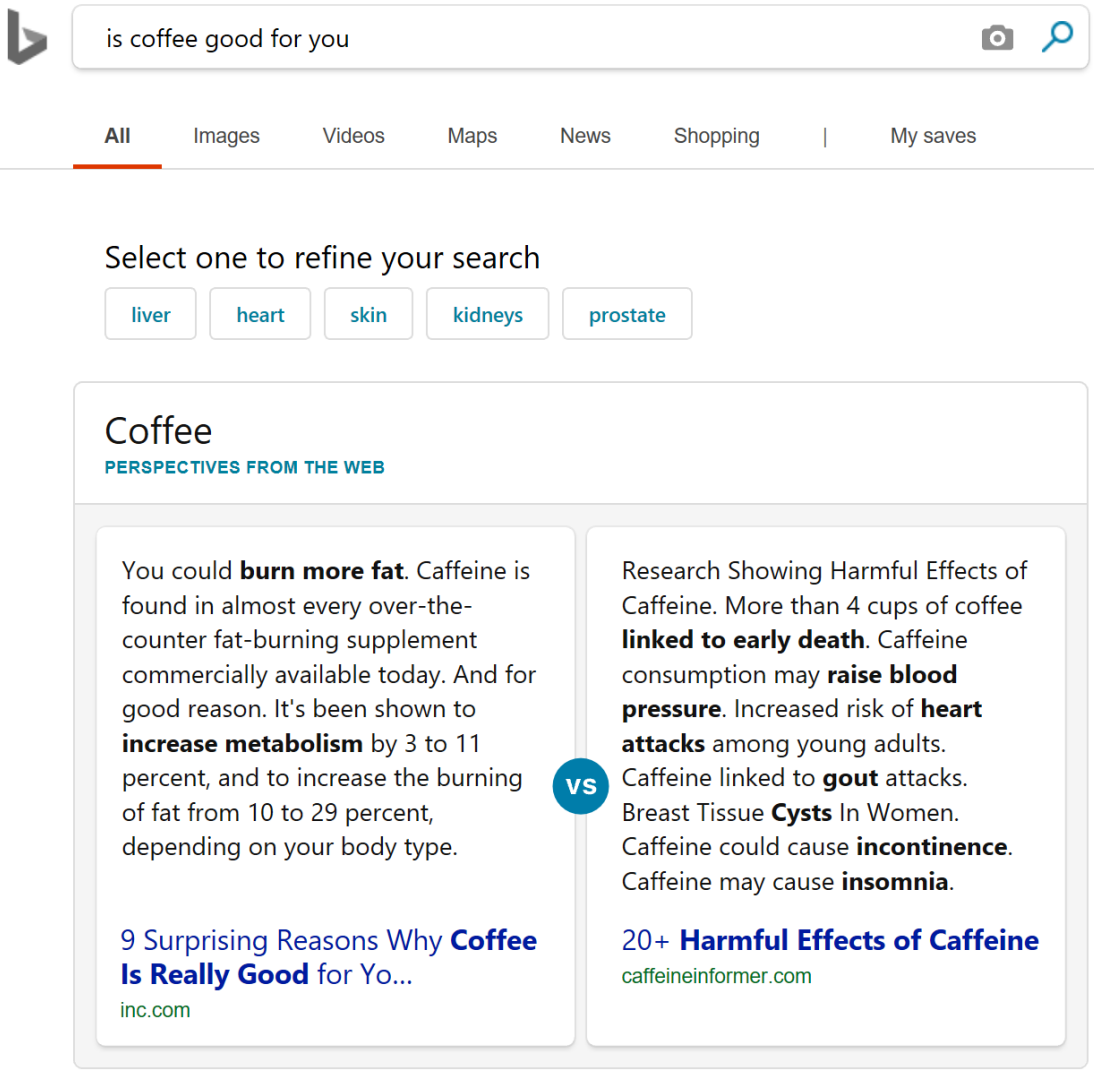


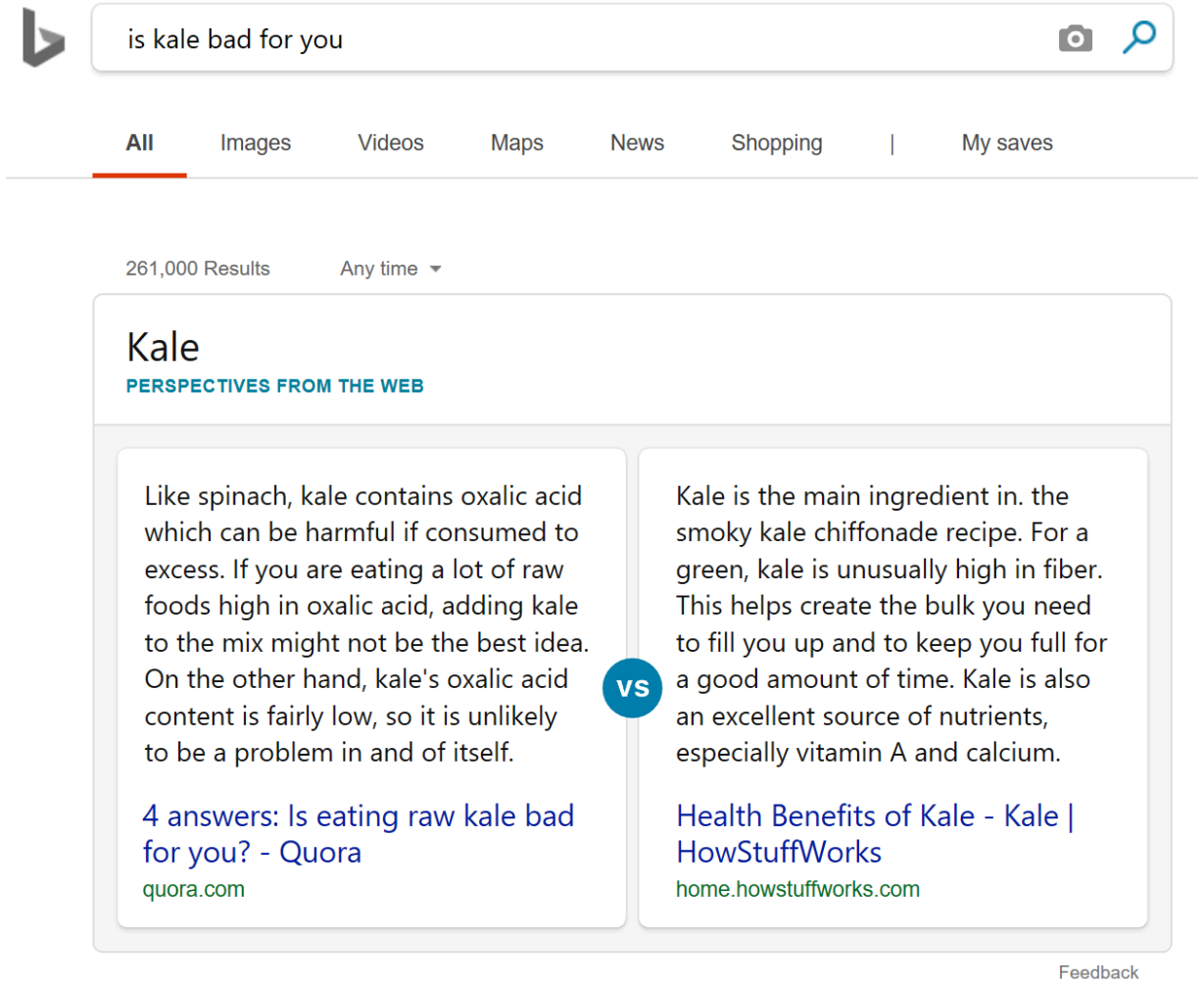


Object Detection



Multiple Perspectives





Entity Recognition

How old is the CEO of NVIDIA

Sign in

All | Images | Videos | Maps | News | Shopping | My saves

2,180,000 Results Any time

Jen-Hsun Huang · Age

56 years old

Born Feb 17, 1963

Lisa Su
50 years old

Chris Malachowsky
61 years old

Jerry Yang
50 years old

Vienna Teng
40 years old

Jen-Hsun Huang

CEO of NVIDIA

Jen-Hsun "Jensen" Huang is a Taiwanese American entrepreneur and businessman. He co-founded the graphics-processor company Nvidia in 1993 and has served as its president and CEO since inception. Huang graduated from Oregon State University … +

Wikipedia LinkedIn

Born: Feb 17, 1963 (age 56) · Taipei, Taiwan

Net worth: $3.60 billion USD (2018)

Spouse: Lori Huang

Founded: NVIDIA

Education: Stanford University · Oregon State University

Segments

travel to seattle

All Images Videos

travel to seattle Larger

Space Needle

Extraordinary Views of the Water

Viewpoints · Skylines

Seattle travel guide

bing.com/travelguide?q=Seattle

Best time to visit: Jul-Aug

Estimated cost: $635 for 2-day trip

Suggested itinerary: 2 days in Seattle

Travel guide

msft

All News Images Videos

Microsoft Corp (MSFT)

117.21 ▼ 0.44 (0.37%)

Amazo

1,772.

Microsoft Corp

NASDAQ: MSFT · March 20, 1:53 PM EDT

117.21 ▼ 0.44 (0.37%)

Day Week Mor

Open	117.39	P/E
High	117.58	52wk H
Low	116.71	52wk L

Recent news

weather

All News Images

50,000,000 Results Any time

San Jose, CA

Updated over an hour ago

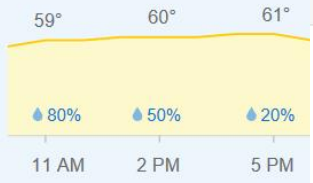
57 °F 62

C 44

Mostly cloudy · Wed 20, 1

11 AM 2 PM 5 PM

Wed 20	Thu 21	Fri 22
62° 44°	63° 47°	63° 48°



Detailed forecast · Radar map · Data from

restaurants

All Maps Images Videos News Shopping | My saves

Near San Jose, California · Change

Local results for restaurants Larger map Best match Cuisine Ratir

McCormick & Schmick's Se...

Yelp (1151)

Seafood · $$$$ · Downtown

11:30 AM - 10:00 PM

The Grill on the Alley

Yelp (843)

Steak · $$$$ · Downtown

Best Eats in San Jose

Mosaic Restaurant & Lounge

Yelp (505)

American · $$$$ · Downtown

6:30 AM - 10:00 PM

Restaurant.com - Official Site

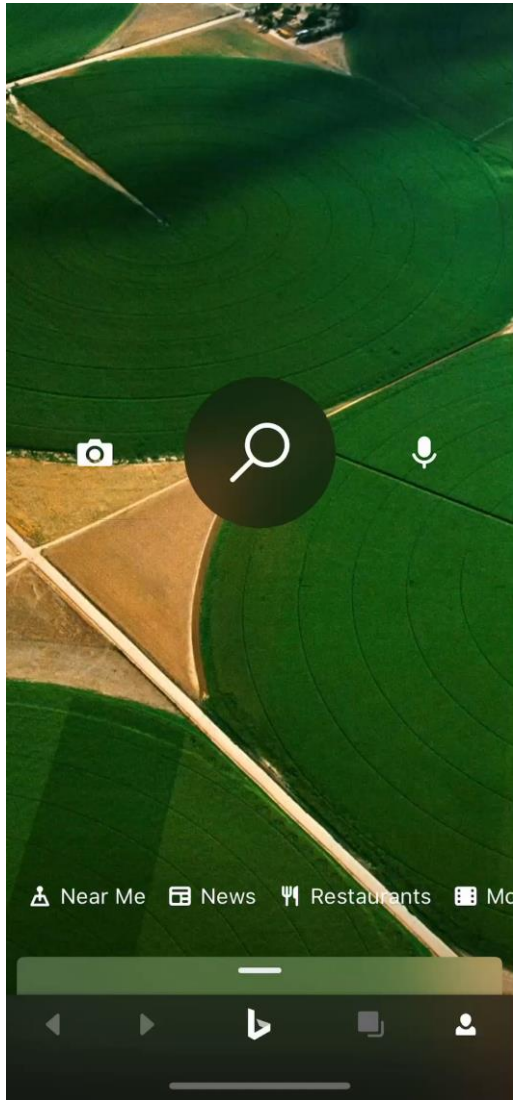
www.restaurant.com

Choose from more than 20,000 **restaurants** nationwide Present the deal at the restaurant on your smartphone or bring a printed copy If you change your mind, it's easy to exchange online

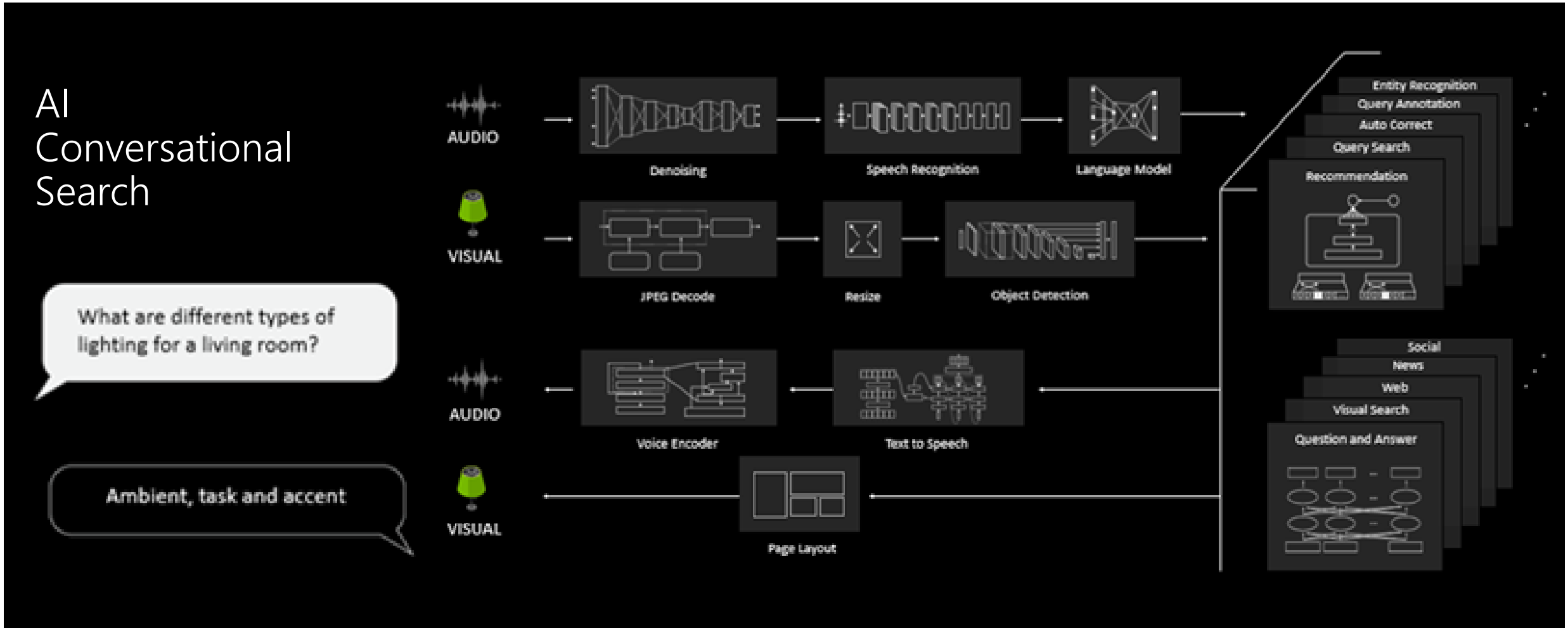
Sign In · Find Restaurants · Redeem Gift · Restaurant.com Cards · About Restaurant · How It Works

Speech

- Provides a very natural interaction model on mobile devices
- Uses Bing Speech API for Speech Recognition
- Uses Azure Neural TTS Service for Speech Synthesis



Slide from Keynote



What it Means for Each AI Model

Trade-offs between multiple priorities, all of which are important:

- Accuracy of Results
- Execution Latency
- Cost to Serve
- Time to Market

Agenda

- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

Challenge #1 – Model accuracy

- Bing relies on many models in production
  - Tracking their accuracy over time is a challenge
- Also – what about versioning?
  - Newer models may perform very well on a certain class of queries
  - But perform poorly on a existing set of queries
- So you might need to keep both versions side-by-side
  - Model overlapping -> platform under-utilization?
- Sometimes the constraints of the latency can be met while slightly sacrificing accuracy
  - Example: quantization

Challenge #2: Latency

- Latency is tied directly with user experience
  - Slow service -> lost customers
  - Large-scale services can allow measuring DSAT to a very fine degree
    - Changing latency bar even by 50 ms has noticeable impact on engagement metrics
- Latency is challenging to keep under control in complex systems
  - An occasional “spike” (say 10% of cases) might be OK in a simple system
  - But in a complex system with unreliable components you will have constant perf degradation
- For this reason we measure latency at@ 99% or 99.9%
  - Quantified over a standard interval (ex: 5 min) or volume of requests (100K)

Challenge #2: Latency (cont.)

- Typical metrics
  - Average (inverse of throughput in sequential systems)
  - 50% (not always the same as average)
  - 99% (most likely used to characterize performance)
  - 99.9% (useful to detect anomalies)
- Other metrics
  - Delta between 50% and 99% can be a measure of “spikiness”
  - Can be relative to the 50% or absolute value

Challenge #2: Latency (cont.)

- How to deal with spikiness at large scale?
  - Solution #1 = two servers
    - Send the request to two machines
    - Pick the first one
    - Pros = fast, easy to implement. Cons = expensive (2x the cost0
  - Solution#2 = Cross-server cancellation
    - Send the request to one machine
    - Wait 1..2 ms for ack
    - If not received, send the request to an additional server
    - Pick one of the two
    - Pros = much lower cost than solution #1. Cons = 1..2 ms added to the latency

Challenge #3: Cost to serve

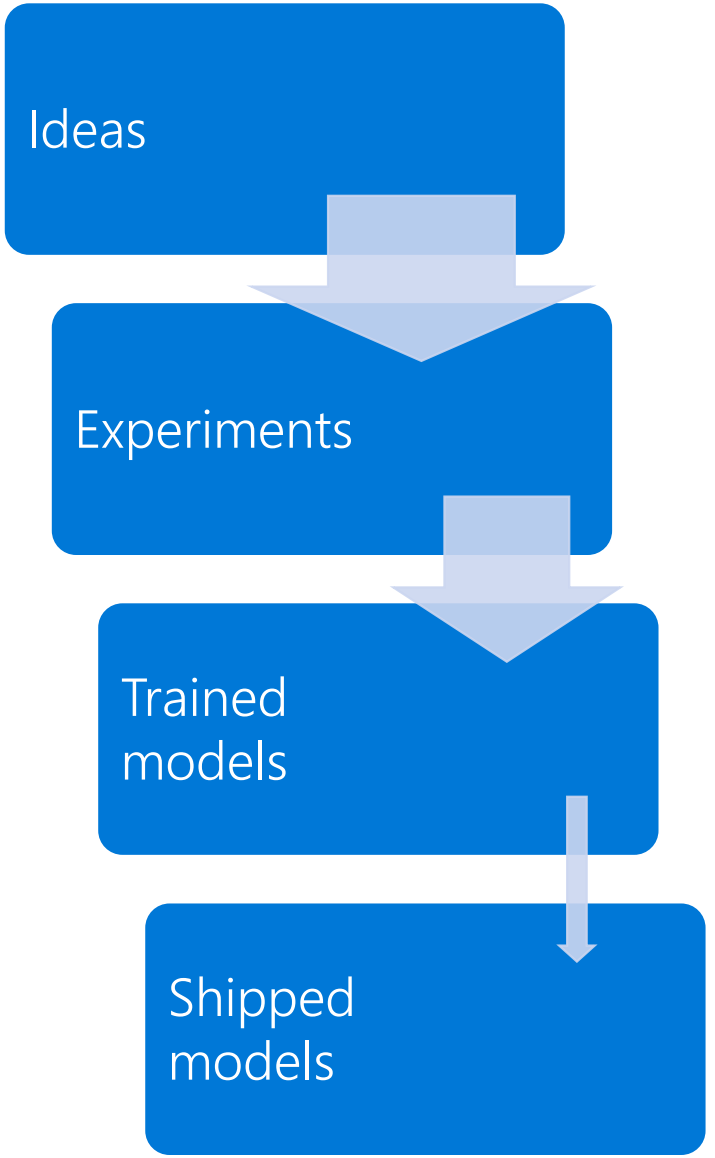
- Depends on the hardware cost
  - Obviously, newer GPUs like Nvidia Tesla V100 have higher cost per GPU
  - But in some cases they might offer higher throughput
- Also depends on hardware utilization
  - Smaller GPUs like P4/T4 are “safer” to saturate
  - Larger GPUs like P40/P100 are more expensive and harder to utilize at 100%
    - Interference between separate tasks/inferences
    - GPU saturation causes drop in performance, affecting many co-hosted models
  - It is harder to co-host models in a multi-tenancy fashion, but possible
    - MPS (Multi-processing extensions) introduced in V100
    - TensorRT Inference Server (allows sharing a CUDA context across multiple models)

Challenge #3: Cost to serve (cont.)

- Comparison across multiple HW platforms can be daunting
  - Solution – strive for Apples-to-apples comparison
  - Always use a meaningful metric such as ops/$/hour
- Example: CPU versus GPU comparison
  - CPUs can be cheaper than GPUs
    - For a significant fraction of inference scenarios, CPUs are the right choice
    - And there is nothing wrong with that
  - GPUs make sense only when inference latency in GPUs for the equivalent model is much faster
    - Example: if the target is 15 ms (already achievable on CPU) and GPUs are 2x more expensive then GPU inference needs to be at least 2..3x faster than CPU inference latency
    - You can still achieve the 15 ms target latency with room to spare (used in queuing requests)
    - Ops/$ HW cost is still better since you need ½ or less GPU instances

Challenge #4: Shipping!

- Asymmetric distribution of effort vs. complexity
- It's easy to write and iterate small prototypes
  - Takes a lot of experimentation effort (training, etc)
  - Many approaches are tried in parallel – low effort
- Assuming you have some candidate models
  - Then more effort needed to optimize the model and ship it
  - But optimization/shipping takes time
    - What if it doesn't meet the bar? -> Wasted effort
  - The feedback loop isn't always clear
- How can we unblock the data scientists?



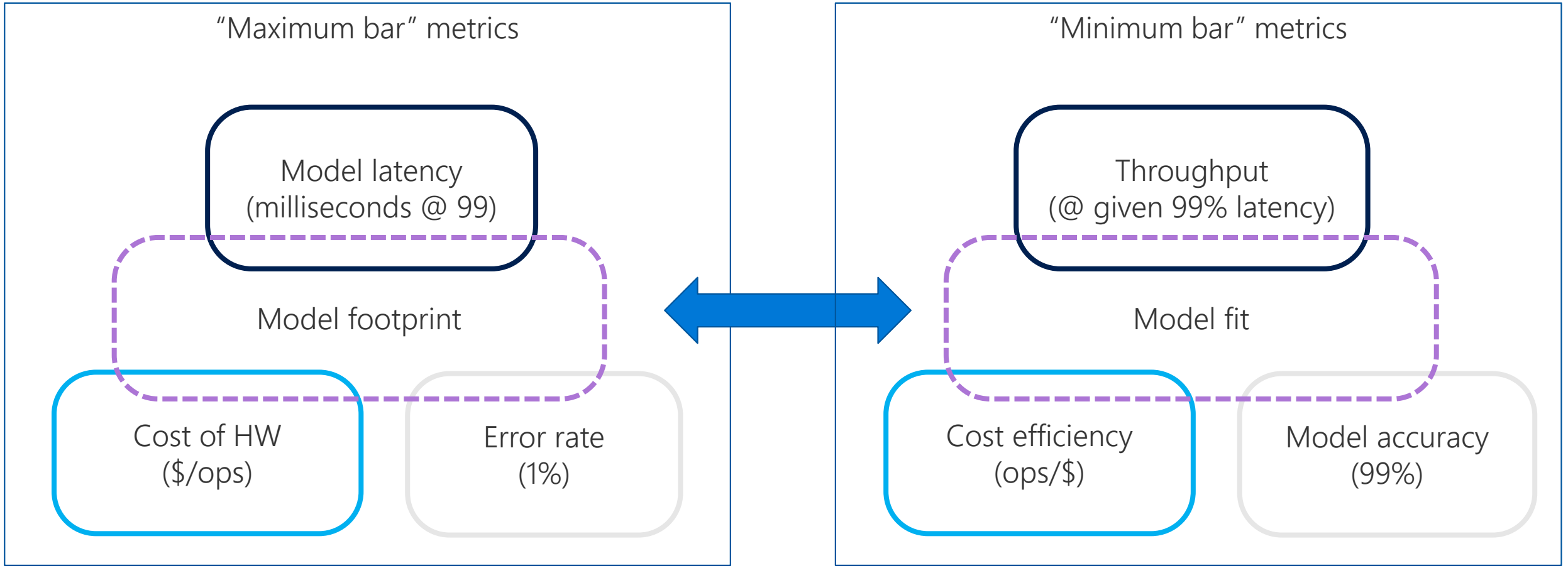
Model fit

- A common problem during shipping inference model is ensuring that the models are not bottlenecked by the memory architecture
  - Example: large weight/bias vectors that won't fit in "local" GPU memory
  - Can significantly slow down operations like GEMM, dot-product, etc.
- Solutions
  - **Temporal breakdown** of computational nodes in the graph
    - Separate computational graph into smaller operations that can individually execute much faster while minimizing DRAM access
  - **Spatial breakdown**
    - Reorganize larger multiplication operations into groups of smaller operations
    - May even improve parallelism
  - **Model quantization**
    - Reduce precision of model parameters
- Bottom line: achieving "model fit" fast is critical for agility

Agenda

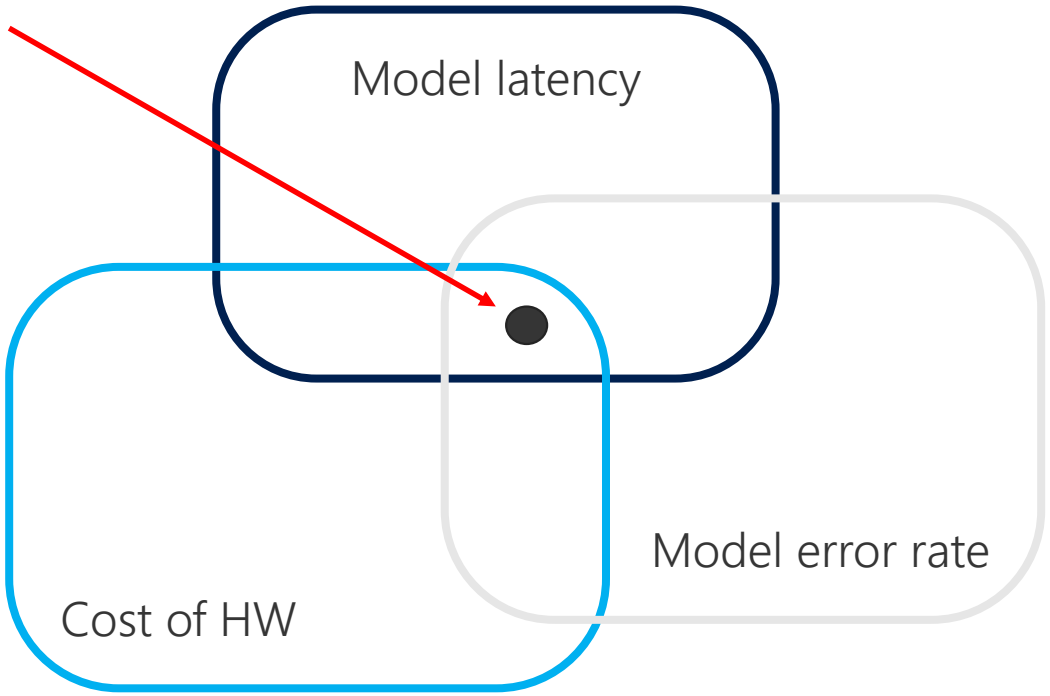
- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

Shipping tradeoffs - a visual taxonomy



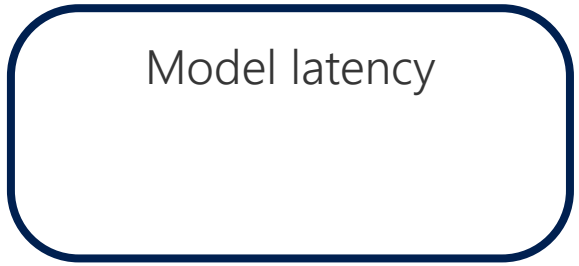
Example 1: the ideal case

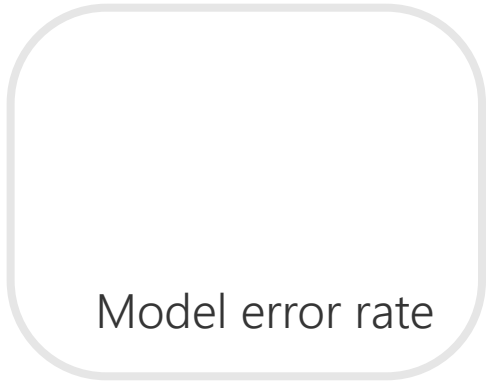
- Everything fits!
- There is at least one model that meets
  - Throughput/latency
    - Meets the shipping bar
    - Example: < 10 ms @ 99%
  - Cost efficiency
    - Allows serving traffic within budget
  - Accuracy
    - Better than previous models



Example 2: nothing fits

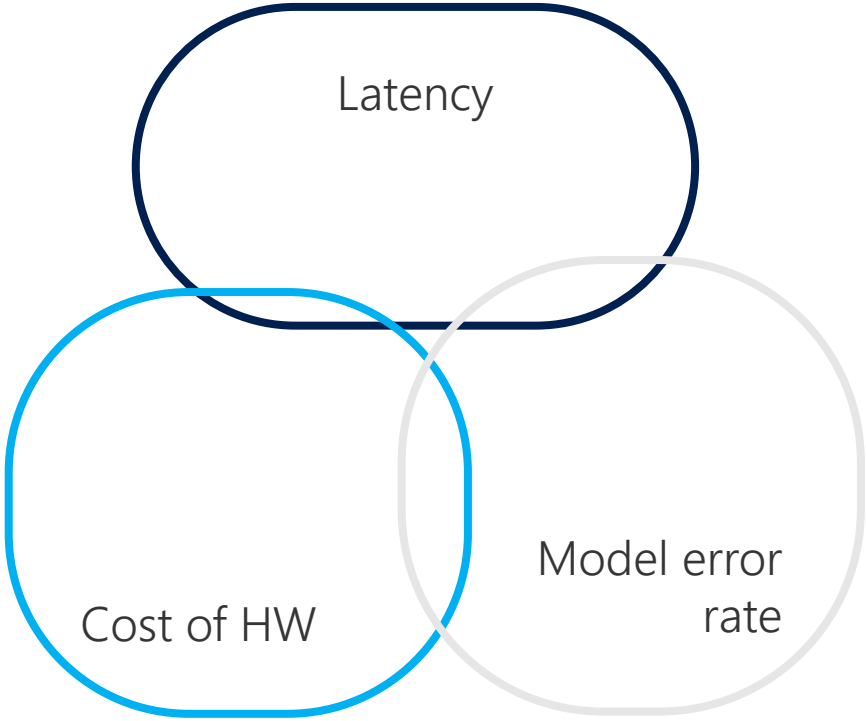
- No model that meets
  - Throughput/latency
    - Meets the shipping bar
    - Example: < 10 ms @ 99%
  - Cost efficiency
    - Allows serving traffic within budget
  - Accuracy
    - Better than previous models





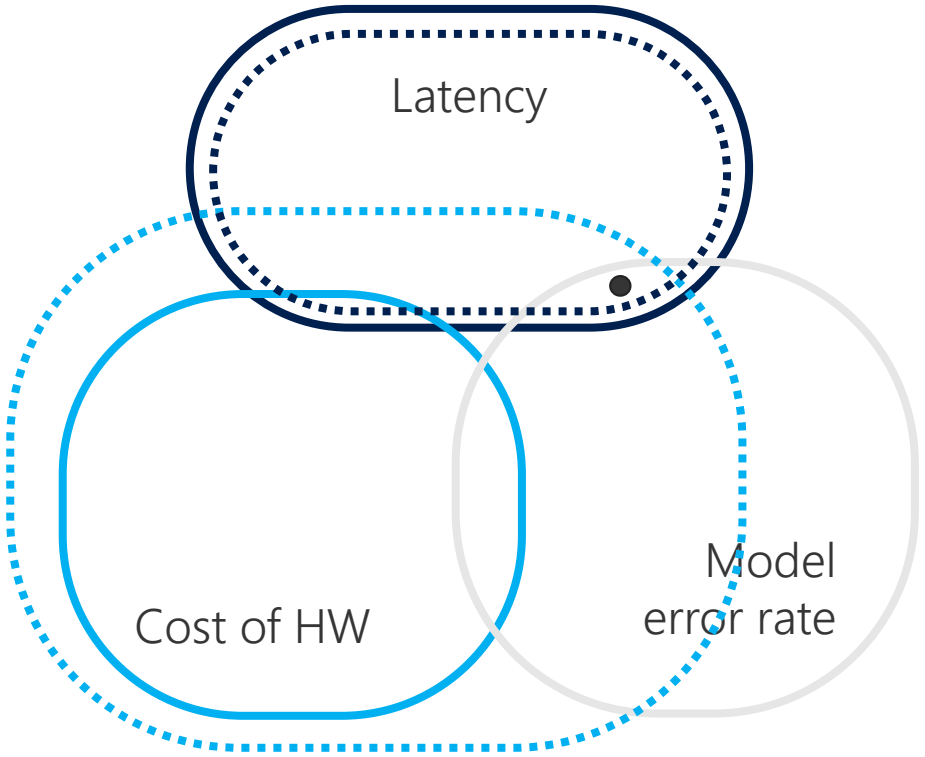
Example 3: Almost a good fit

- You can have any two out of three …



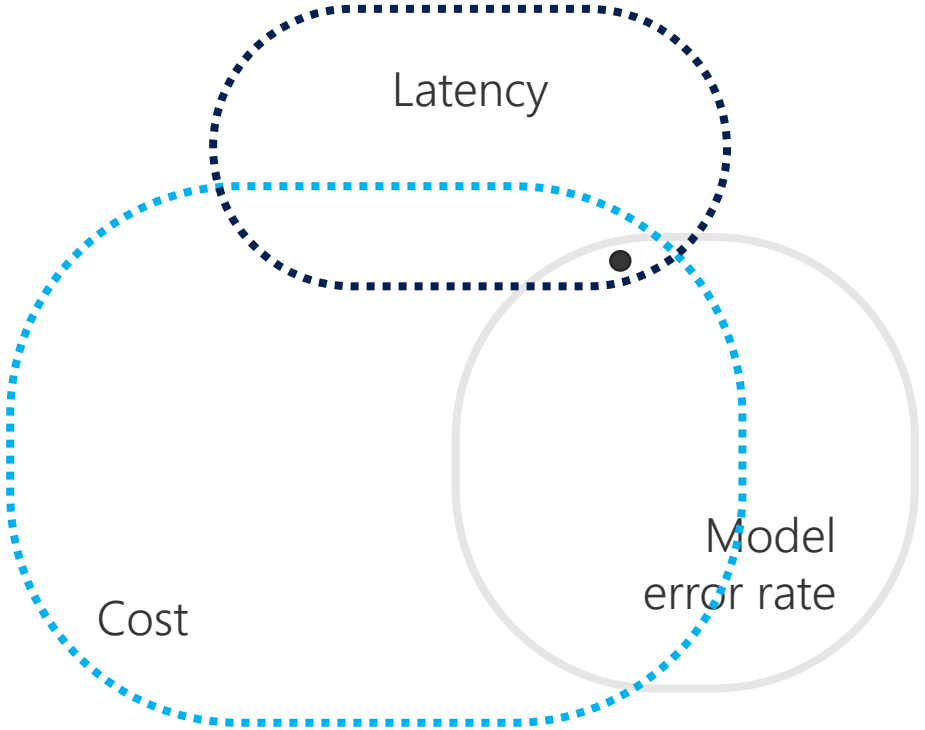
Example 3: Almost a good fit (cont.)

- You can have any two out of three …
- But what if you can tweak the system?
  - Use an older HW
  - At a slight expense on latency



Example 3: Almost a good fit (cont.)

- You can have any two out of three …
- But what if you can tweak the system?
  - Use an older HW
  - At a slight expense on latency

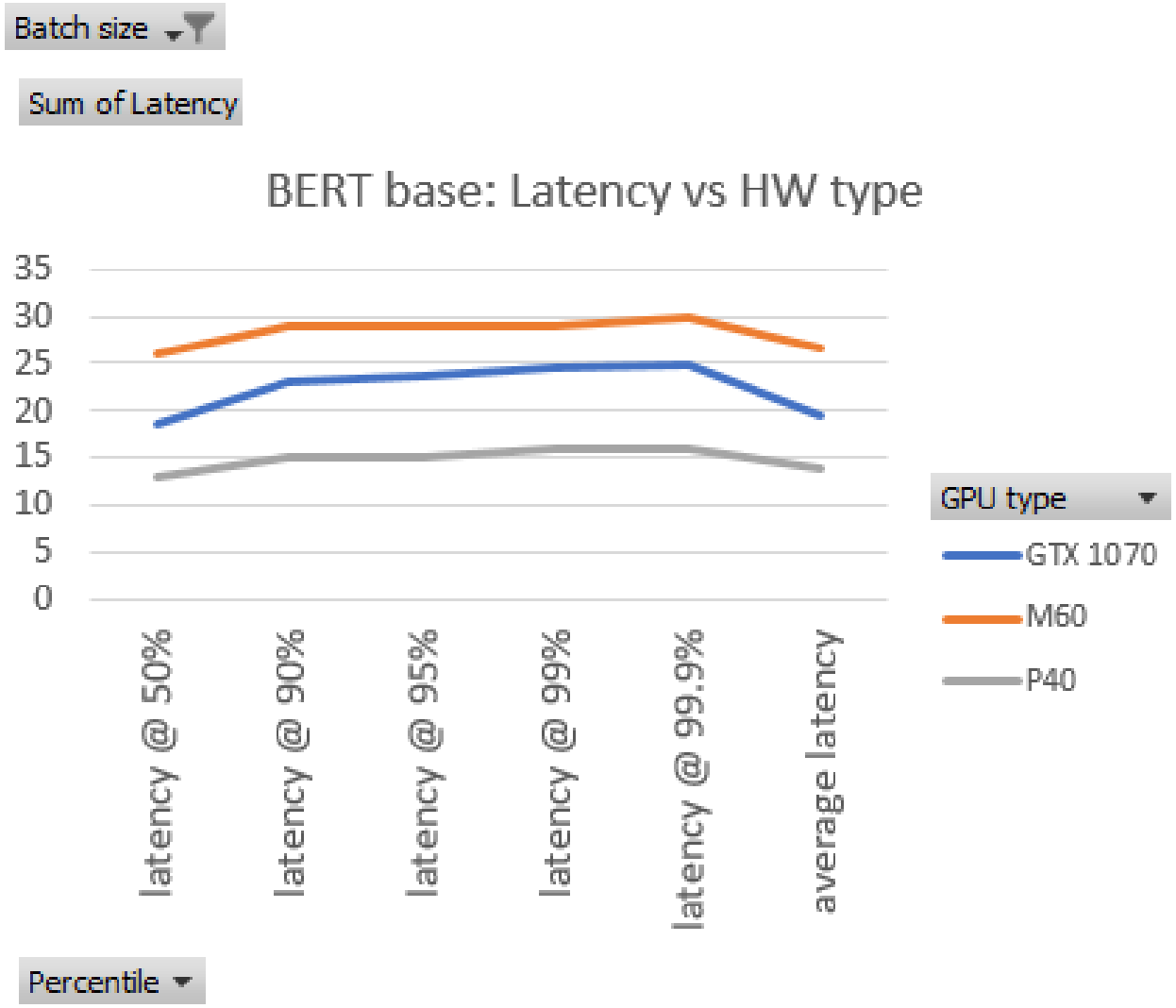


Agenda

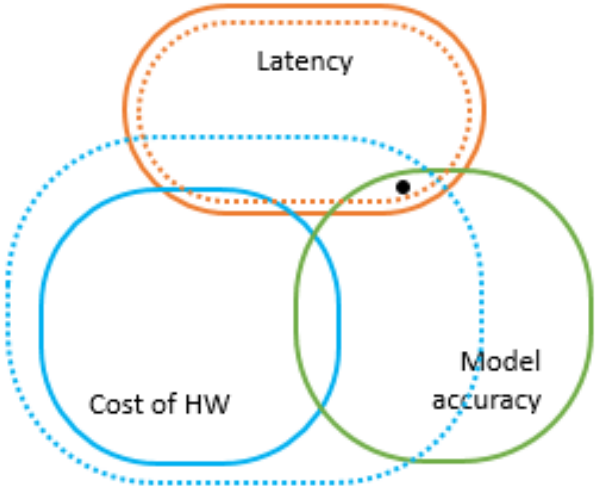
- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

Case study: BERT base: latency vs. HW type

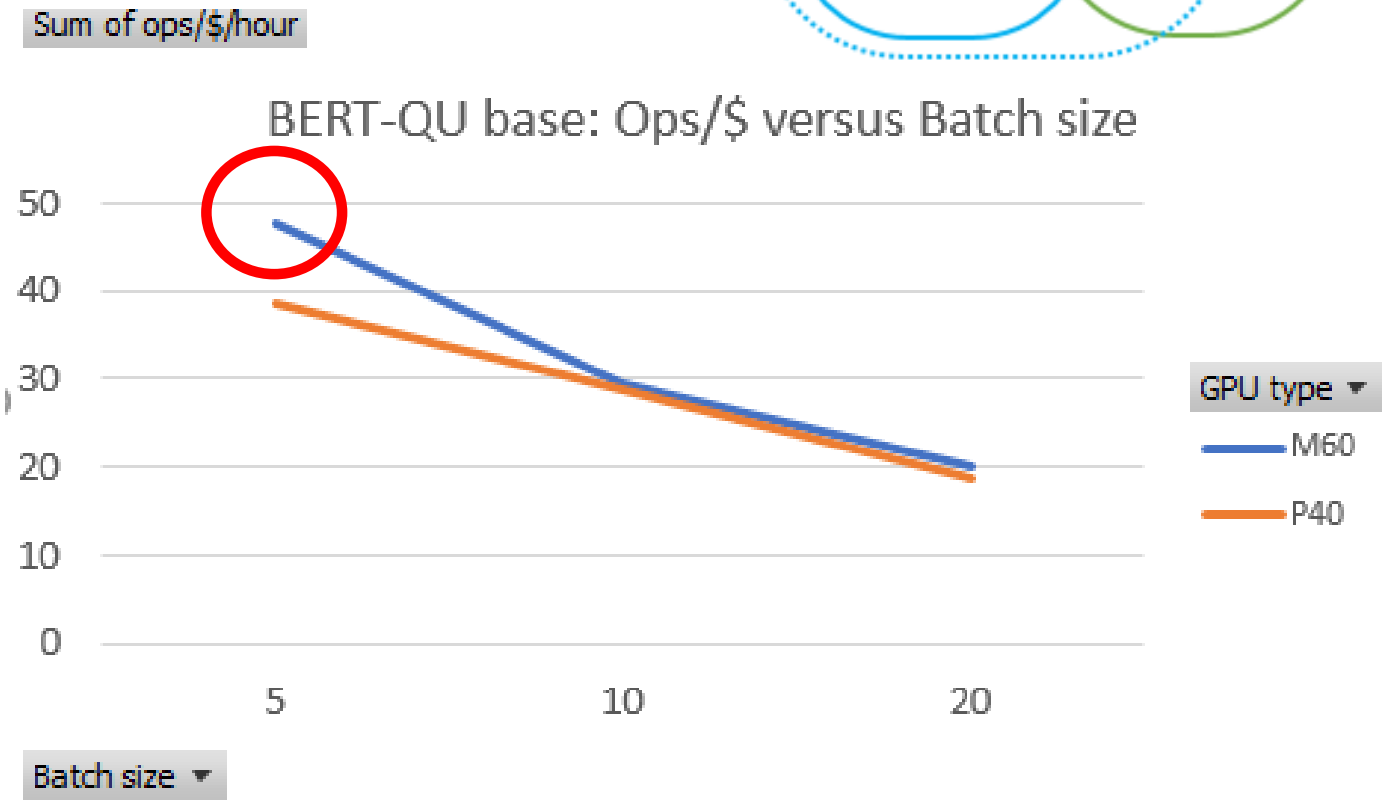
- BERT-base based model
  - 12-layers
  - Input size = 20
  - Batch = 10
- HW
  - Nvidia Tesla M60
    - Azure VM = Standard_NV6
  - Nvidia Tesla P40
    - Azure VM = Standard_ND6
  - GTX 1070



BERT base - ops/$ vs. HW type



- BERT-base based model
  - 12-layers
  - Input size = 20
  - Batch = 5, 10, 20
- HW
  - Nvidia Tesla M60
    - Azure VM = Standard_NV6
  - Nvidia Tesla P40
    - Azure VM = Standard_ND6
  - (Azure prices in US West 2)
- Conclusion
  - P40 more efficient for batch=5



Agenda

- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

Model fit technique #1: Temporal/spatial breakdown

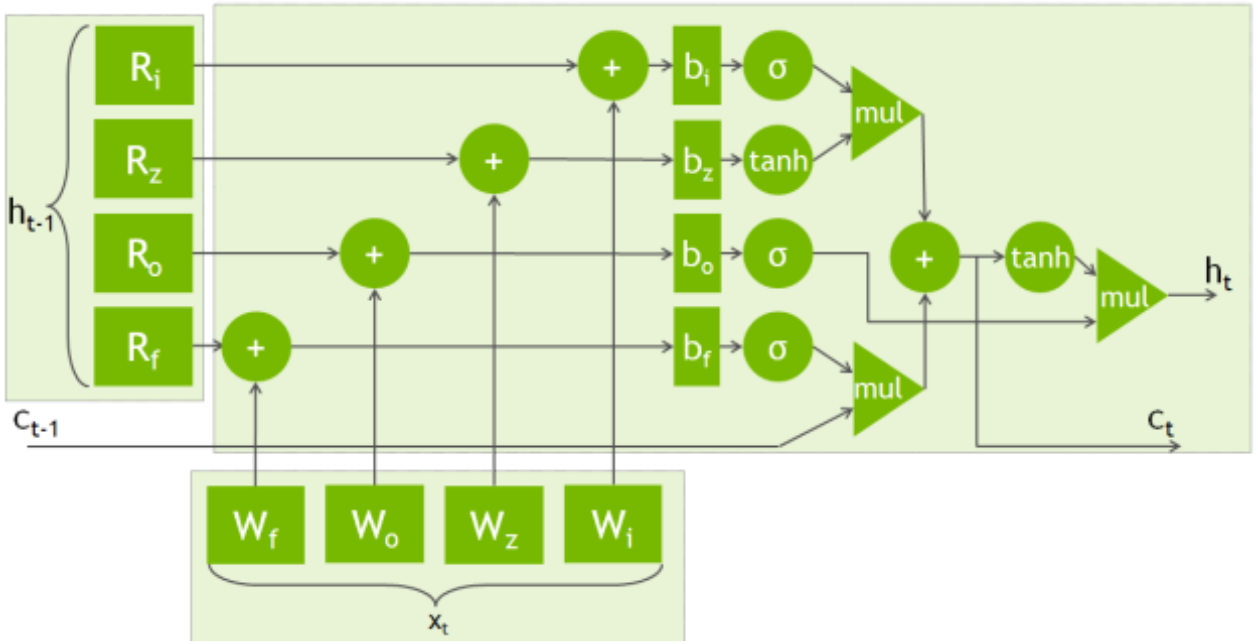


Diagram source: https://devblogs.nvidia.com/optimizing-recurrent-neural-networks-cudnn-5

Case study: Typical word-based RNNs

- Hidden dimensions = ~100
- Input size = 100 (or more)
- Output size = 100
- Bi-directional
- Time steps = 100+ (one timestep per word in a sentence/paragraph)

Problem:

- Native TensorFlow execution is slow
  - 20K kernel invocations
  - 5 us/invocation
  - ~100 ms delay
- We need a custom implementation [1]
  - TensorFlow custom op for RNN execution
  - Collaboration with Nvidia

[1] - S81039 – GTC 2018: Accelerating Machine Reading Comprehension Models Using GPUs

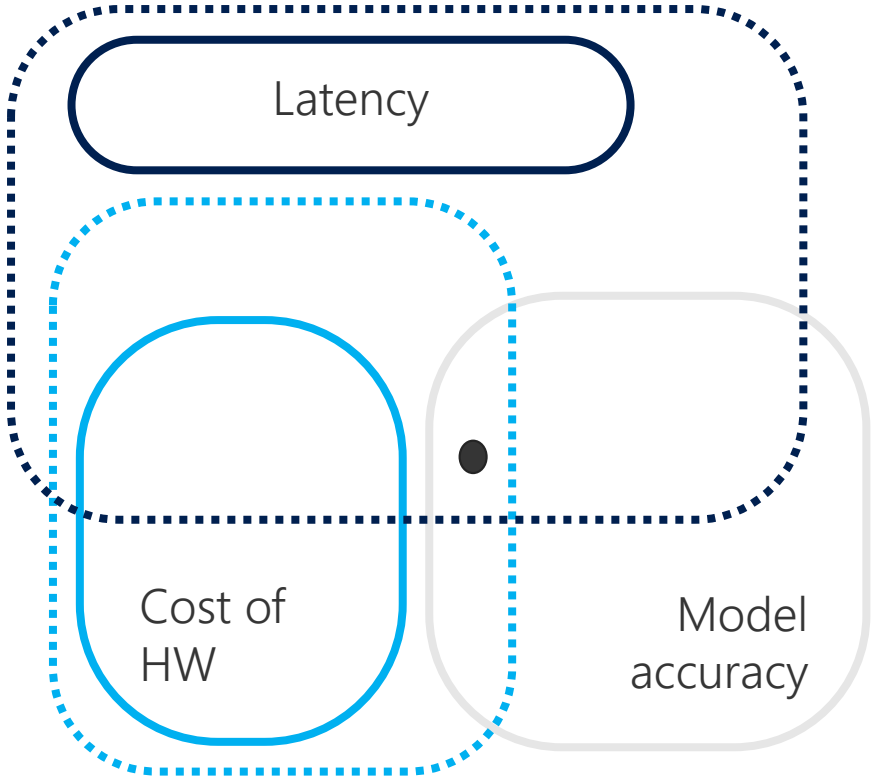
Block-persistent RNNs – perf results

- Time is averaged over back-to-back 100 iterations (different input data)
- 40 timesteps

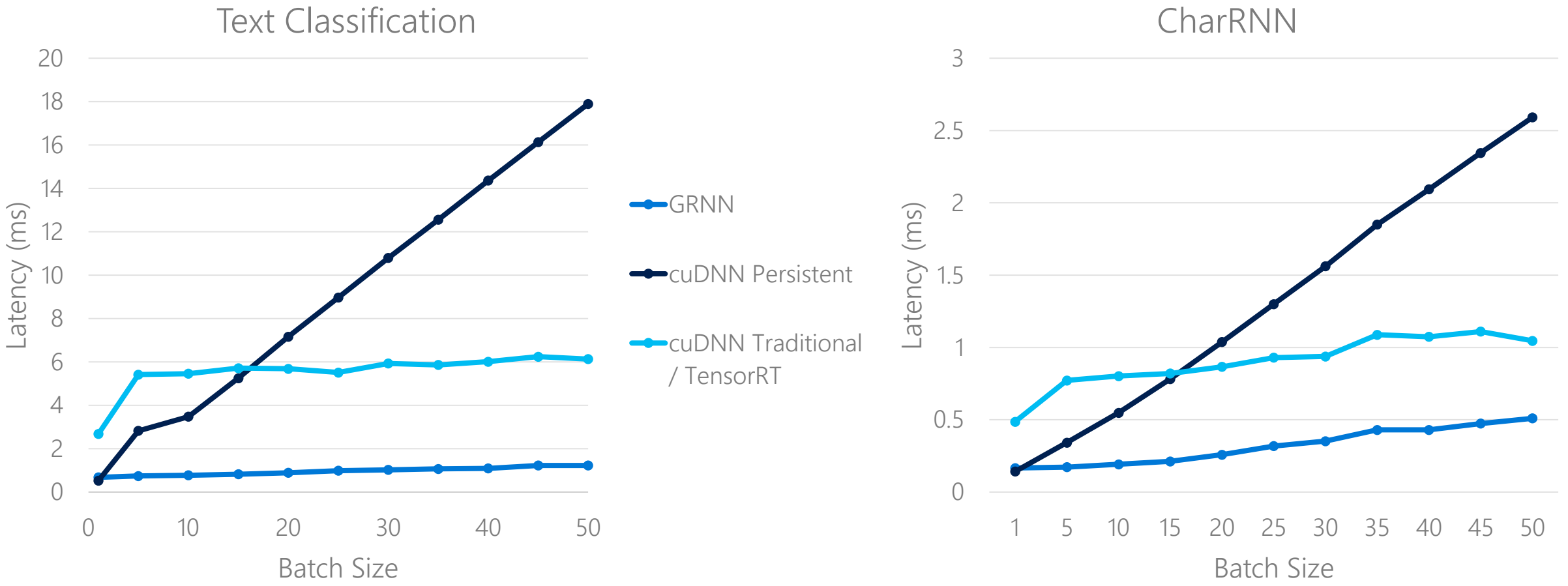
1) For K80 and M40 we used CUDNN_RNN_ALGO_STANDARD, since grid persistent approach is supported starting from Pascal.
2) For P4 we chose CUDNN_RNN_ALGO_PERSIST_STATIC, and it is more than 2x faster than block persistent.

Block-persistent RNNs results – joint collaboration with NVidia

	cuDNN		Blockpersistent	
	Batch 1	Batch 5	Batch 1	Batch 5
K80	9.30	6.52	1.77	1.87
M40	2.78	2.90	0.95	1.00
P4	0.25	1.06	0.68	0.72



GPU Inference Optimization - GRNN



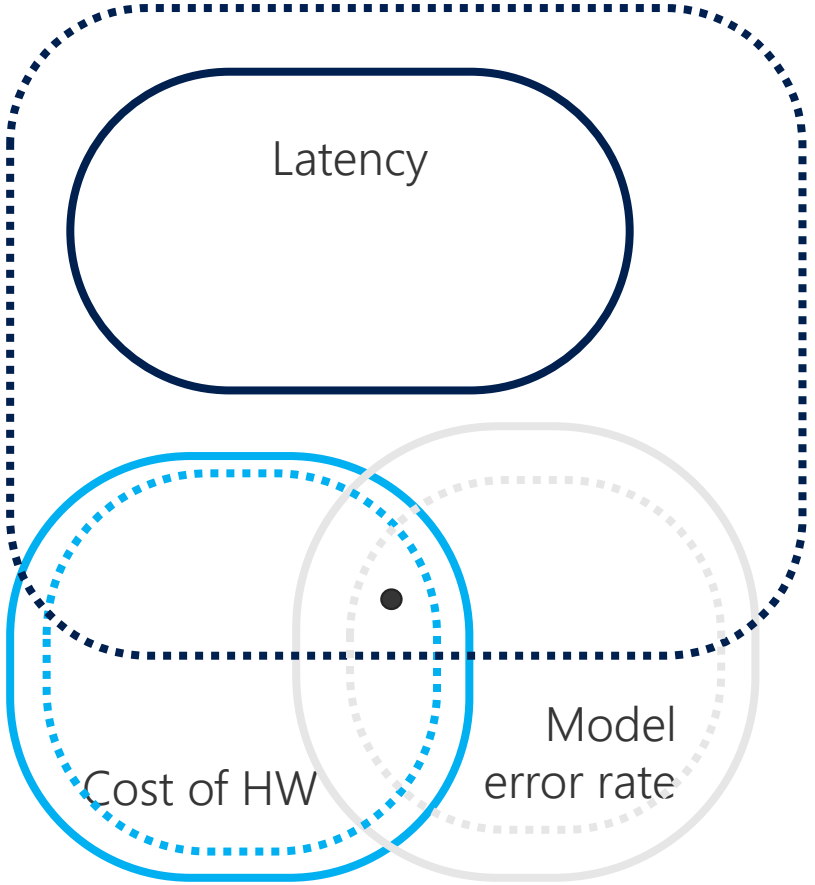
* GRNN: Low-Latency and Scalable RNN Inference on GPUs. Connor Holmes, Daniel Mawhirter, Yuxiong He, Feng Yan, Bo Wu. To Appear @ Eurosys 2019.

Agenda

- Introduction
- Key Challenges
- Trade-offs – A Visual Taxonomy
- Case Studies
  - Cost to serve: BERT
  - Model fit: Temporal/spatial breakdown: word-based RNNs
  - Model fit: Quantization: ELMO

Model fitting technique #2: quantization

- Case study: ELMO [1]
  - Deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy).
- Characteristics
  - Main block: Very large LSTM, 4-layer
  - Hidden states = 4096
  - Hard to fit in memory
- We can improve model fit through quantization
  - Latency greatly improved
  - We might need more expensive HW (INT8 support is only in V100)
  - Model accuracy suffers



[1] - https://allennlp.org/elmo

ELMO - INT8 quantization details

Projection bi-LSTM from the last stage dominates the execution time
2 layer bi-LSTM, 512 input, 4096 hidden, 512 projection

Batch 1, 100 timesteps, takes about 70 ms on P40 in FP32 when using cuDNN

Reduce precision to INT8 to meet latency requirements

We choose conservative quantization approach, keep as much as possible in FP32

Custom implementation (Joint collaboration with Nvidia):

1) Single cuBLAS call per layer for input GEMMs for all timesteps (done in FP32)
2) Elementwise ops kernel in FP32
3) Custom MatVec kernel for recurrent part (at batch 1)
    - Input / Output vectors are FP32
    - Recurrent weights are quantized once
    - All math is FP32, so older GPU architectures can be used as well

ELMO - INT8 – Results

- Only hidden and projection weights are quantized, input weights and all bias are in FP32
- In the quantization scheme (e.g. https://arxiv.org/abs/1609.08144v2), each row has a separate scale factor
  rowMax = max(abs(row))
  scale = 127.0 / rowMax
  quantRow = int(round(row * scale))

Results at batch 1

	cuDNN FP32, ms	Custom INT8, ms
P40, ECC ON	70.0	20.7
V100, ECC ON	41.9	11.9

Conclusions

- Complex engineering tradeoffs when shipping models in production
- Use proper metrics when deciding inference platform
  - Ops/$/hour
  - Hardware acceleration choices (CPU vs. GPU, choice of HW)
- Achieving model fit is critical for agility
  - Custom ops: Temporal/spatial breakdown
  - Quantization

Q&A

Guhan Suriyanarayanan, guhans@microsoft.com
Adi Oltean, adi.oltean@microsoft.com

Please don’t forget to rate our talk!

Thank You

Microsoft