From Honey Bee to Mouse Brain

M4: Massively Multilingual, Massive Machine Translation

Aditya Siddhant, Ali Dabirmoghaddam, Ankur Bapna, Colin Cherry, Dmitry Lepikhin, George Foster, Isaac Caswell, James Kuczmarski, Kun Zhang, Macduff Hughes, Mahdis Mahdieh, Manisha Jain, Markus Freitag, Maxim Krikun, Melvin Johnson, Mia Chen, Naveen Arivazhagan, **Orhan Firat**, Roee Aharoni, Sébastien Jean, Sneha Kudugunta, Thang Luong, Wei Wang, Wolfgang Macherey, Yanping Huang, Yonghui Wu, Yuan Cao, Zhifeng Chen

May 8, 2020

Virtual Talk @ UPC-Barcelona

Agenda

1. **Goal & Motivations**
2. Project Phases
3. Lessons Learned

Our goal

Develop a universal machine translation model (i.e. one model for all languages and domains)

*"Perhaps the way [of translation] is to descend, from each language, down to the common base of human communication -- the real but as yet **undiscovered universal language** -- and then re-emerge by whatever particular route is convenient."*

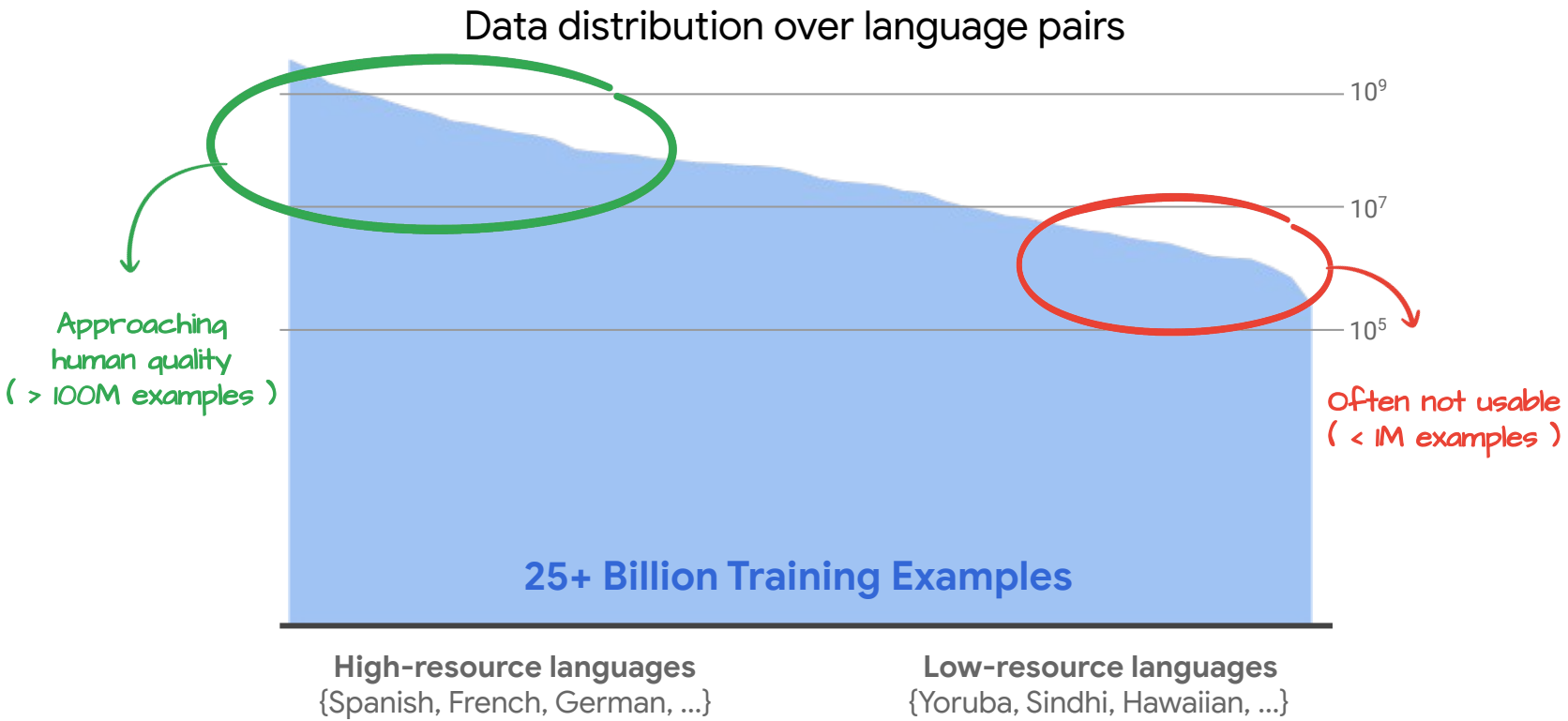
Warren Weaver (1949)

Google AI Blog

The latest news from Google AI

Exploring Massively Multilingual, Massive Neural Machine Translation

Motivation 1:
Improve translation quality for all language pairs



Motivation 2:
Expand language coverage

In the world, there are...	7,000+ Total languages	2,000+ African languages	700+ Native Am. languages[1]
But Translate only supports...	103 Total languages	11 African languages	0 Native Am. languages

1.. Estimate is 766 Native Am languages globally: 383 in South America (source); 176 in central America (source); 207 in North America (source)

Google

how many neurons in the brain

All Images Videos News Maps More Settings Tools

About 39,600,000 results (0.58 seconds)

100 billion neurons

Neurons in the Human Brain

According to many estimates, the human brain contains around **100 billion neurons** (give or take a few billion). Jun 11, 2019

How Many Neurons Are in the Brain? - Verywell

https://www.verywellmind.com/how-many-neurons-are-in-the-b

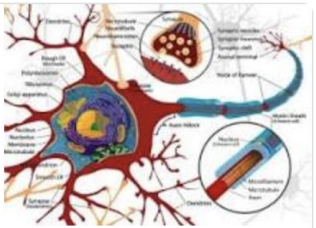
Google

how many connections in the brain

All Images Videos News Shopping More Settings Tools

About 160,000,000 results (0.54 seconds)

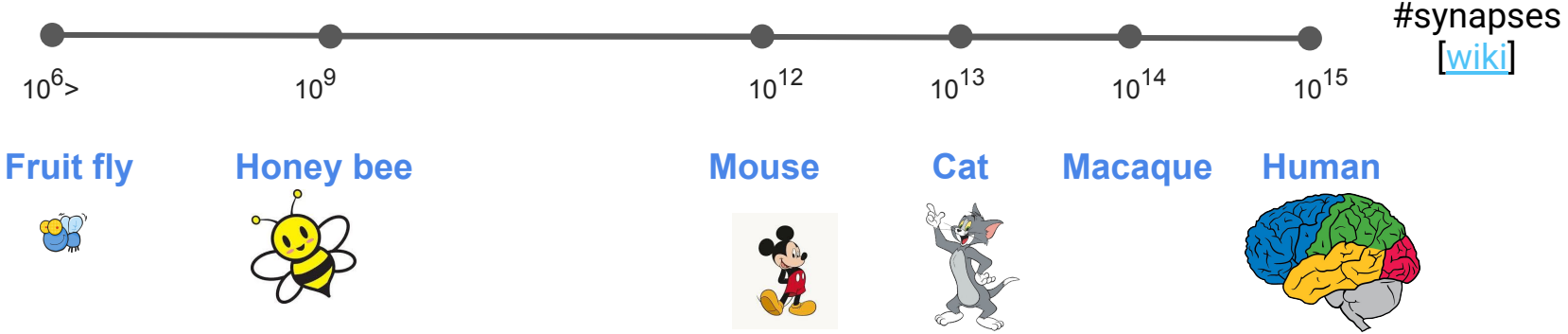
Each individual neuron can form thousands of links with other neurons in this way, giving a typical **brain** well over 100 trillion synapses (up to 1,000 trillion, by some estimates). Functionally related neurons connect to each other to form neural networks (also known as neural nets or assemblies).



Neurons & Synapses - Memory & the Brain - The Human Memory

www.human-memory.net/brain_neurons.html

Number of Synapses



Number of Synapses

NMT with Attention
Resnet50
[25-50M]

Fruit fly

Honey bee

Mouse

Cat

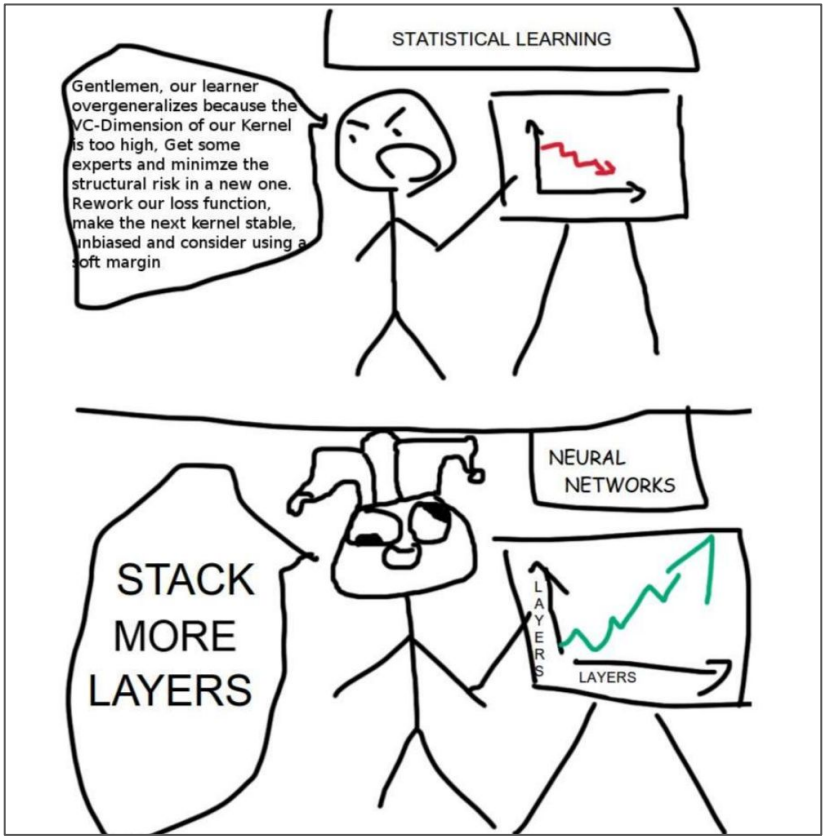
Macaque

Human

Unstoppable Force

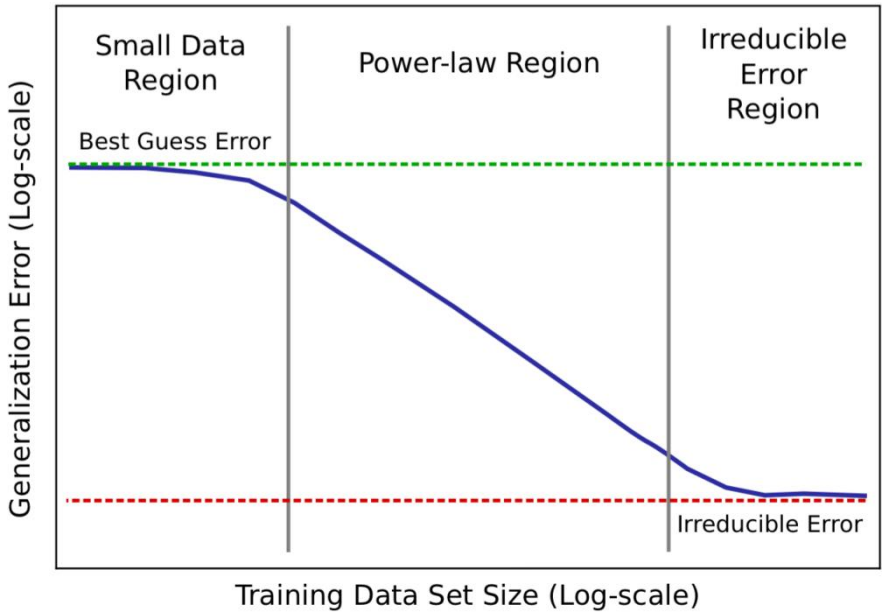
Unstoppable Force vs Immovable Object

Scaling Up Neural Networks



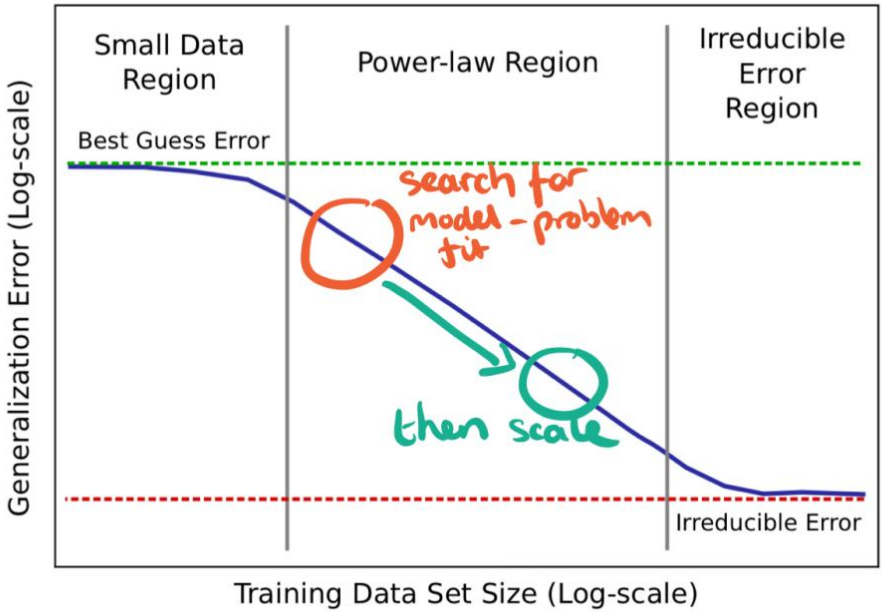
Machine Learning Memes for Convolutional Teens

Motivation 3:
Neural network scaling and the new understanding of generalization



Hestness et al. 2018

Motivation 3:
Neural network scaling and the new understanding of generalization



Hestness et al. 2018

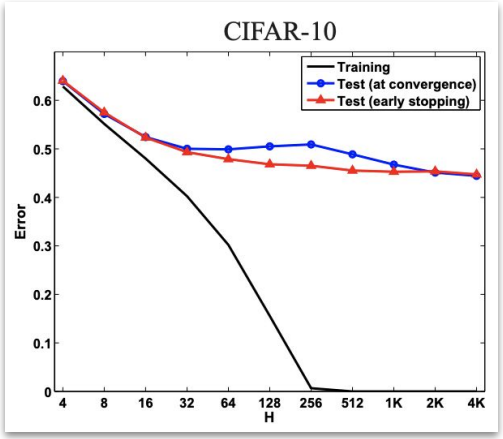
Motivation 3:
Overparameterization and Generalization Theory

 # parameters >> training examples

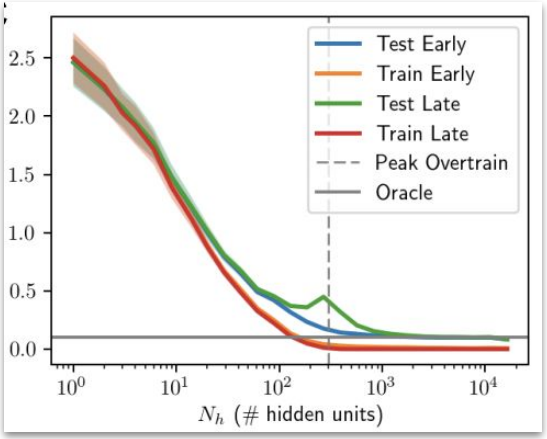
Why do large nets generalize better?
(predict well on unseen data)

reducing approximation error ⇒ reduces estimation error
(increase expressivity) (increase generalization)

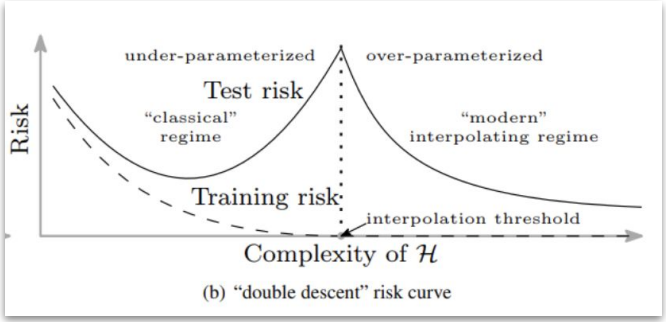
Motivation 3:
Overparameterization and Generalization Theory
Deep Double Descent



Neyshabur et al. 2015

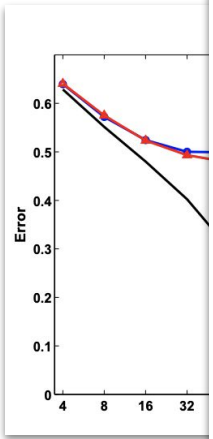


Advani and Saxe 2017

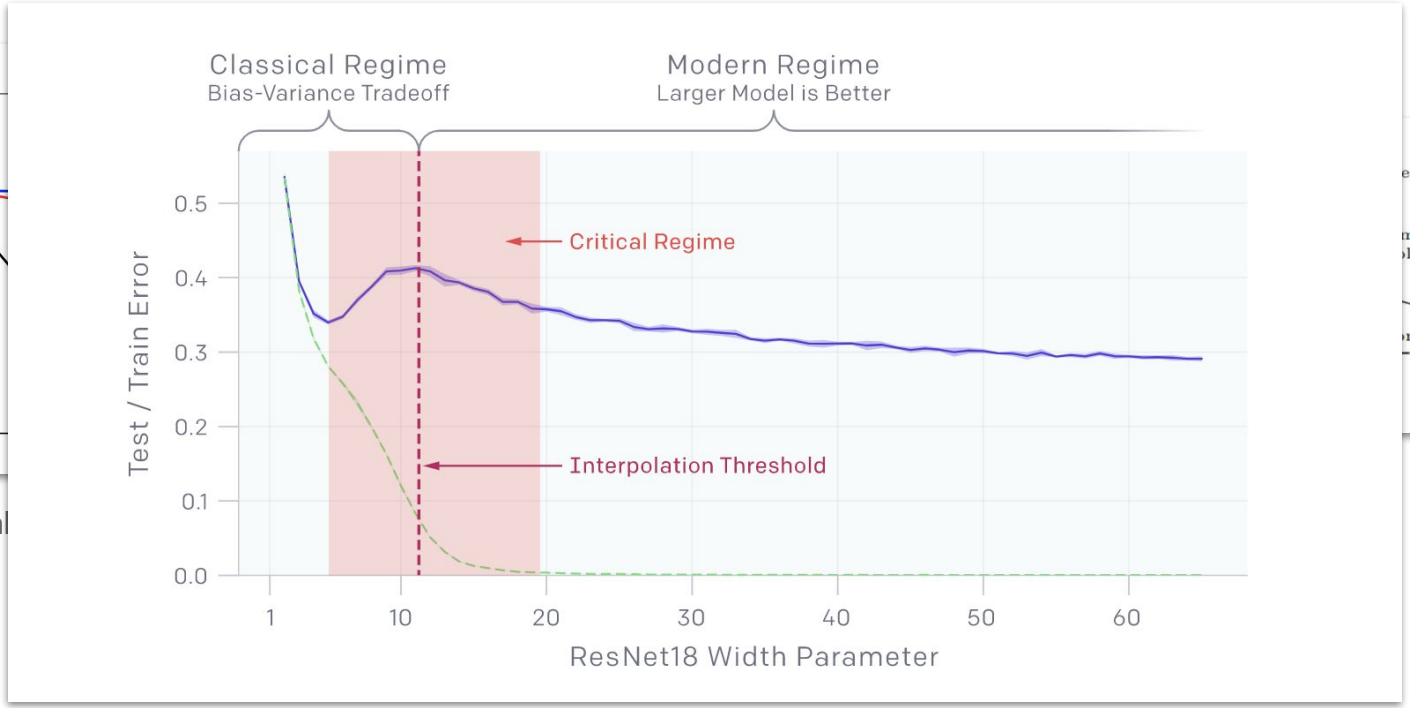


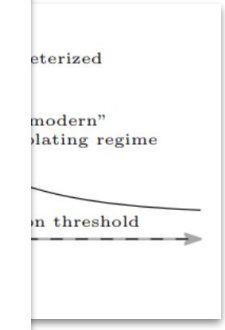
Belkin et al. 2018

Motivation 3:
Overparameterization and Generalization Theory
Deep Double Descent



Neyshal





Nakkiran et al. 2020

Motivation 4:
This is a compelling test bed for ML research

Massive multilinguality requires advances in :

- Multi-task learning
- Meta-learning
- Continual learning

To achieve massive multilinguality, we need massive scale, requires advances in:

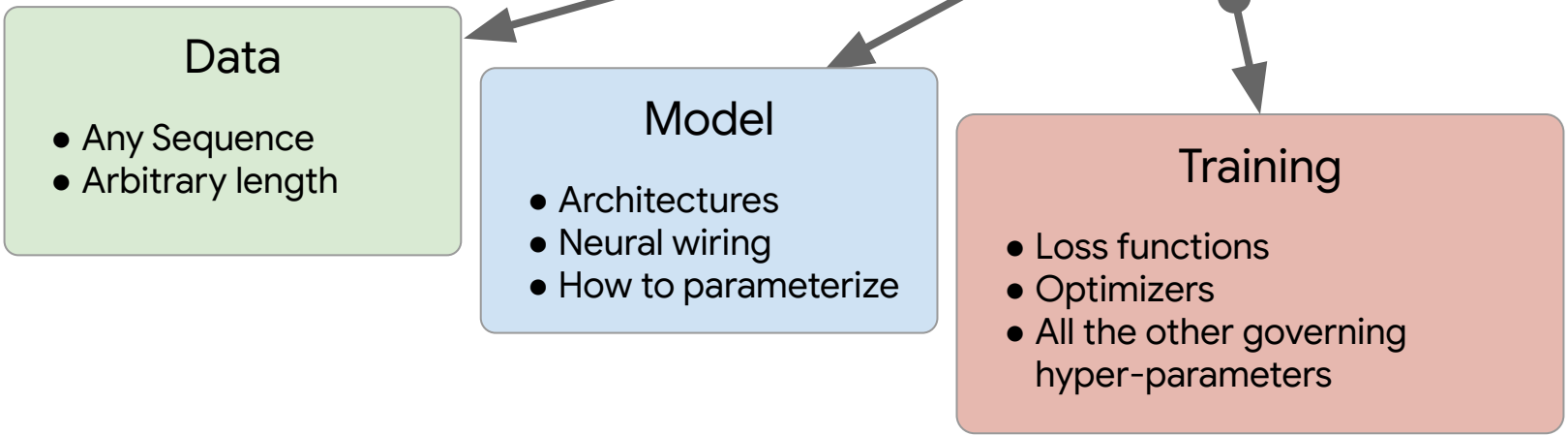
- Model capacity
- Trainability and optimization
- Efficiency improvements

Agenda

1. Goal & Motivation
2. **Project Phases**
3. Lessons Learned

To achieve our goal, we knew we had to dramatically scale capacity; but first, we had to enumerate all relevant research challenges

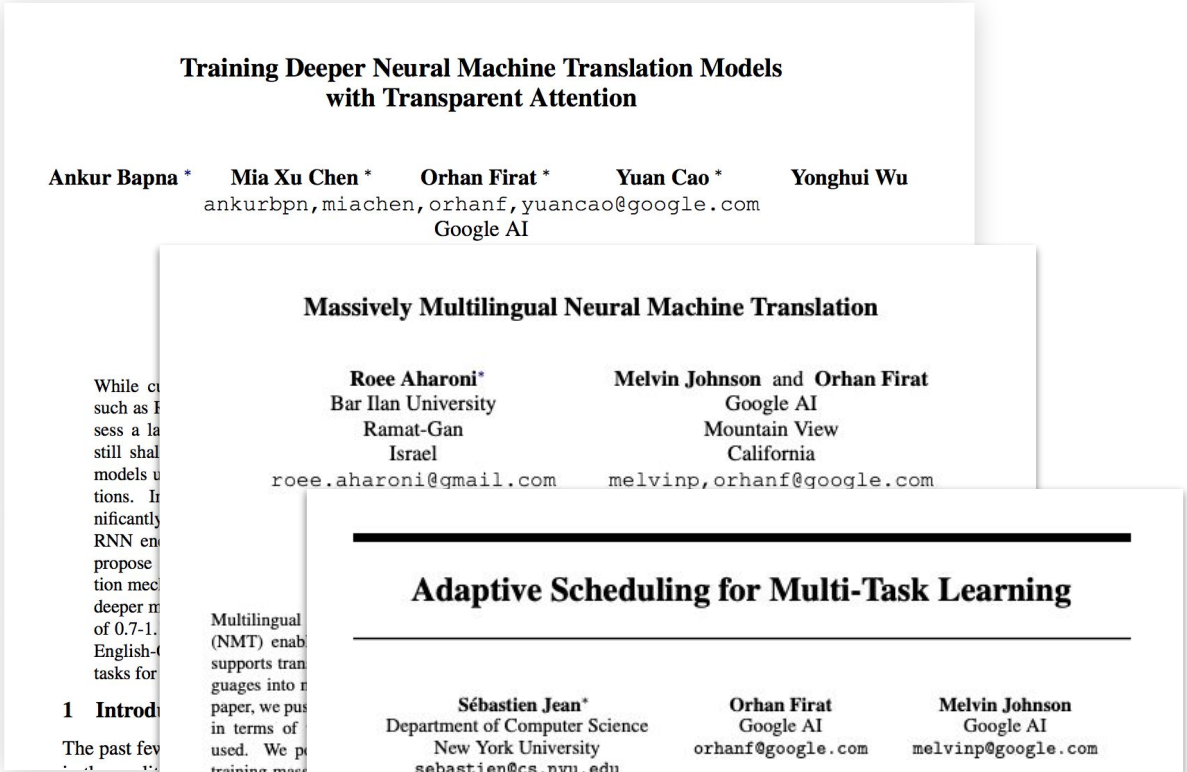
$$\text{quality} = f(X, \theta, \mu)$$



To achieve our goal, we knew we had to dramatically scale capacity; but first, we had to enumerate all relevant research challenges

Conference Publications

- EMNLP
- NAACL
- NeurIPS



Training Deeper Neural Machine Translation Models with Transparent Attention

Ankur Bapna * **Mia Xu Chen** * **Orhan Firat** * **Yuan Cao** * **Yonghui Wu**
ankurbpn,miachen,orhanf,yuancao@google.com
Google AI

Massively Multilingual Neural Machine Translation

Roee Aharoni* Bar Ilan University Ramat-Gan Israel roee.aharoni@gmail.com

Melvin Johnson and **Orhan Firat** Google AI Mountain View California melvinp,orhanf@google.com

Adaptive Scheduling for Multi-Task Learning

Sébastien Jean* Department of Computer Science New York University sebastien@cs.nyu.edu

Orhan Firat Google AI orhanf@google.com

Melvin Johnson Google AI melvinp@google.com

After our pilot studies, we moved to realistic scenario: M4 in the wild

RESEARCH PRIORITIES

- Develop baselines
- Learn, given data imbalance
- Increase model capacity

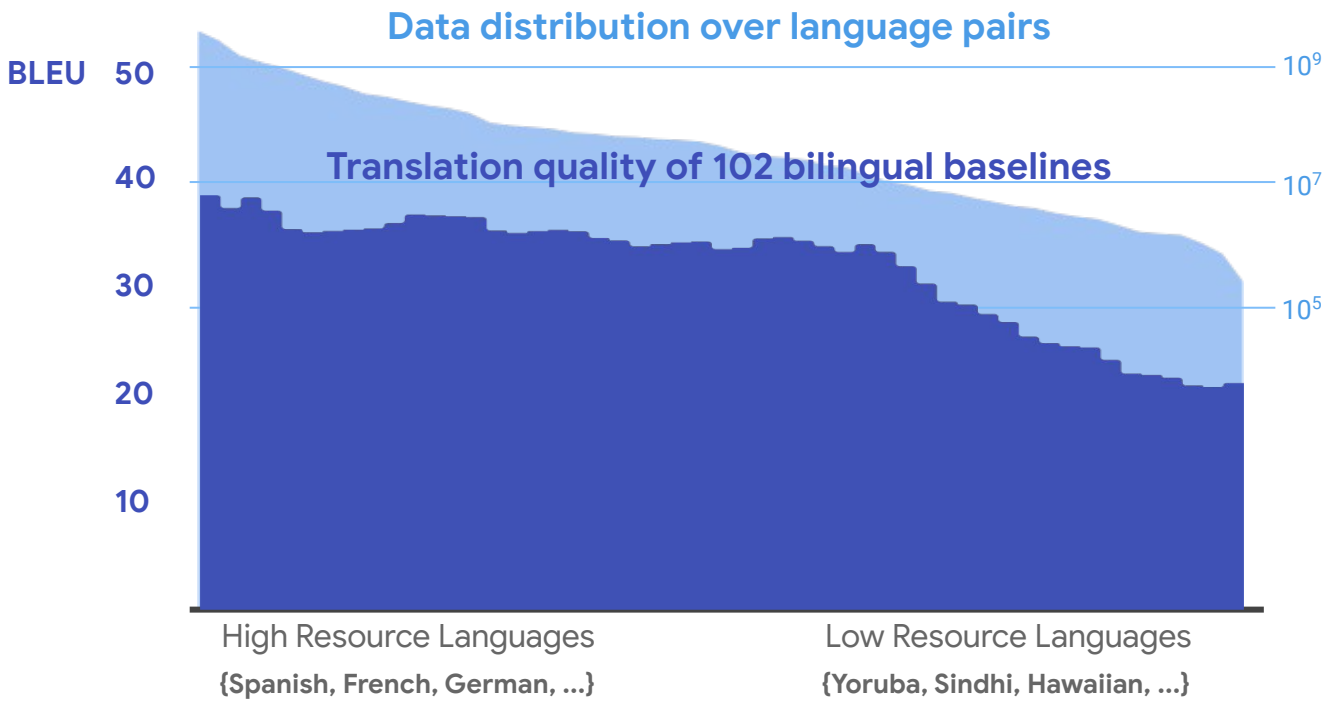
Let's go into detail on each priority

GOAL

Train a 103-language model; attain parity with baselines

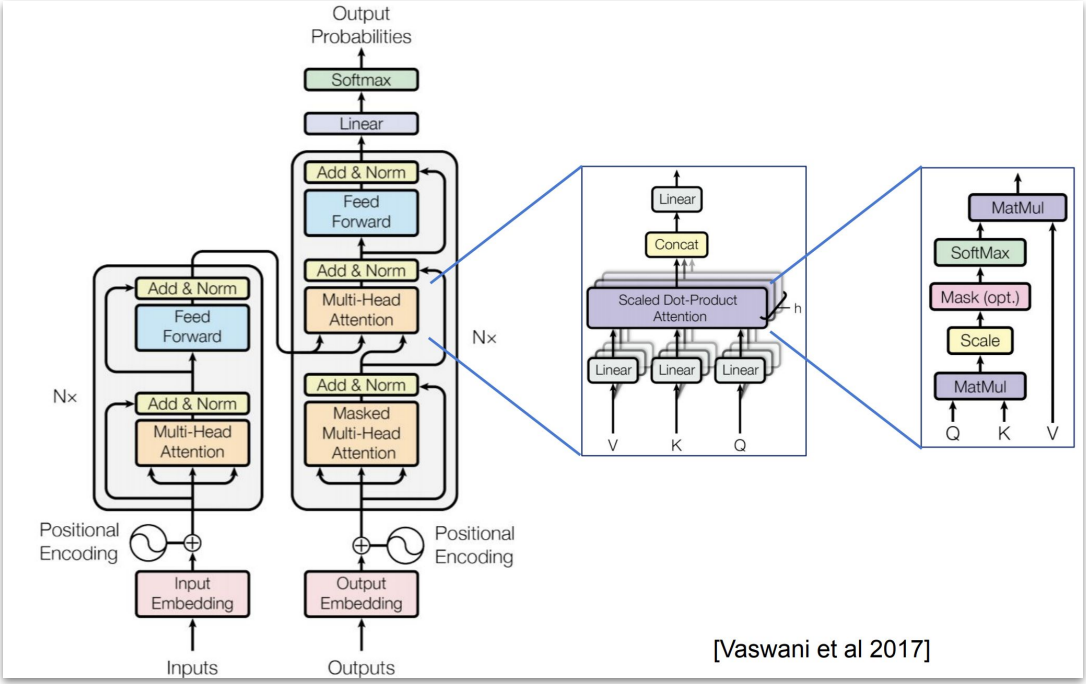
1/ Develop baselines:

We trained and evaluated new bilingual models as controls



1/ Develop baselines:

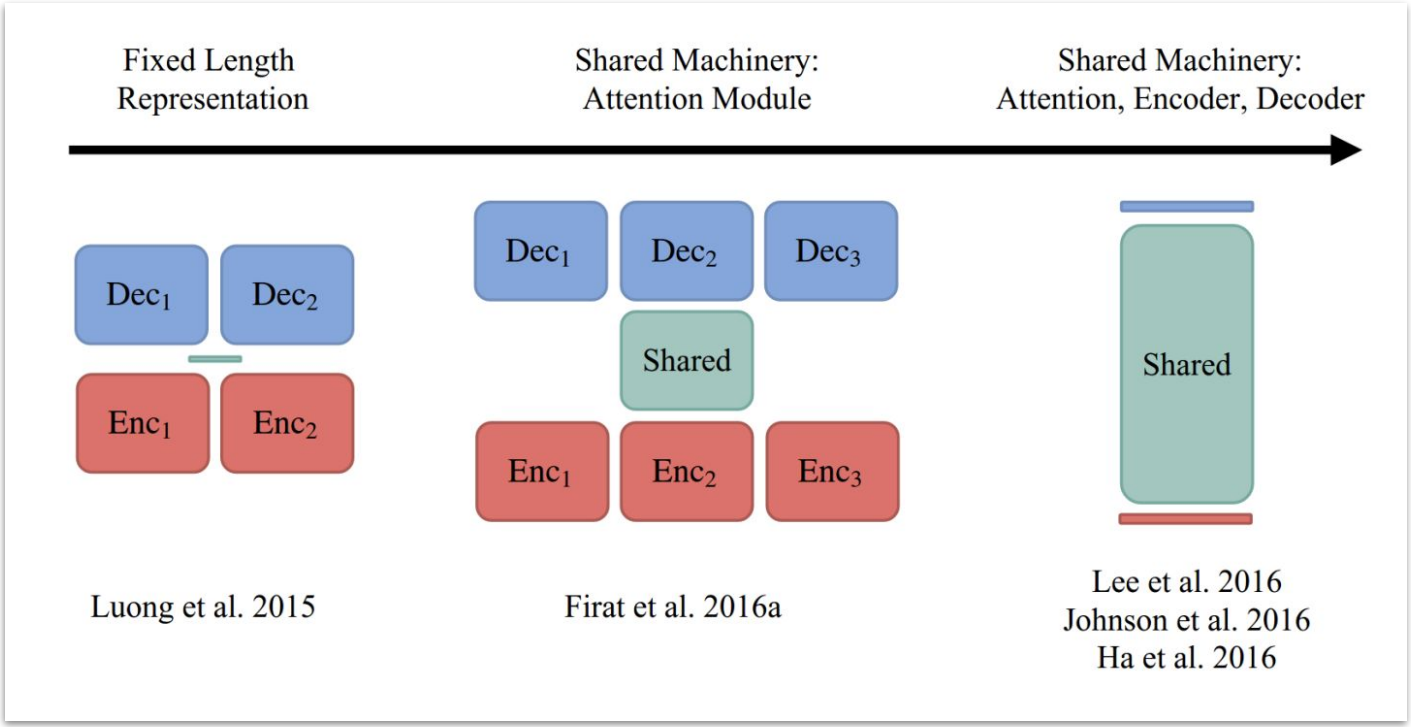
Models
Wiring: Transformer as explained in [1][2] and [3]



[1] Vaswani et al. 2017, “Attention is All You Need”
[2] Chen et al. 2018, “The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation”
[3] Bapna et al. 2018, “Training Deeper Neural Machine Translation Models with Transparent Attention”

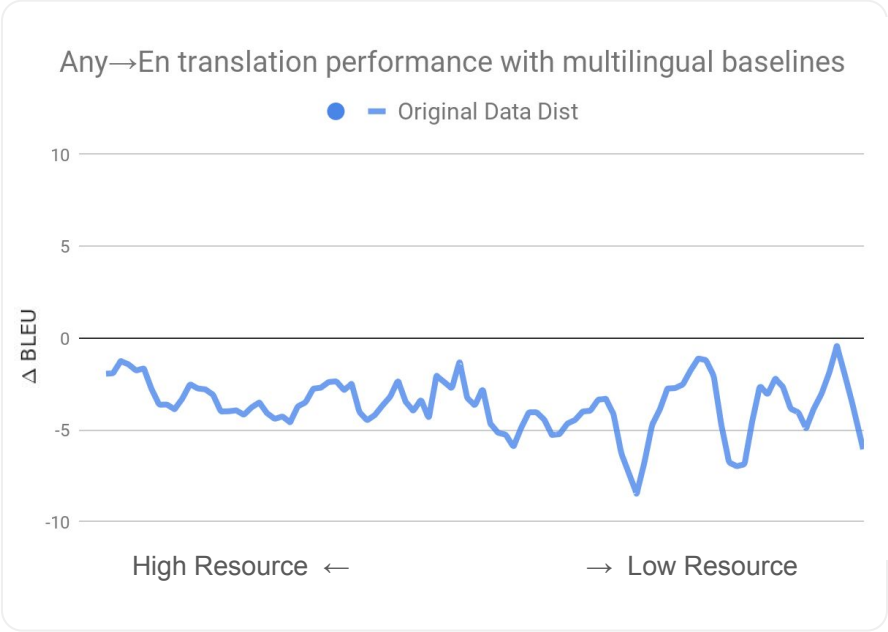
1/ Develop baselines:

Models
Parameter Sharing: Full (rightmost)

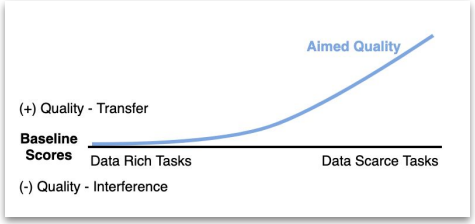


2/ Learn, given data imbalance:

Initial baseline on Any→Any model

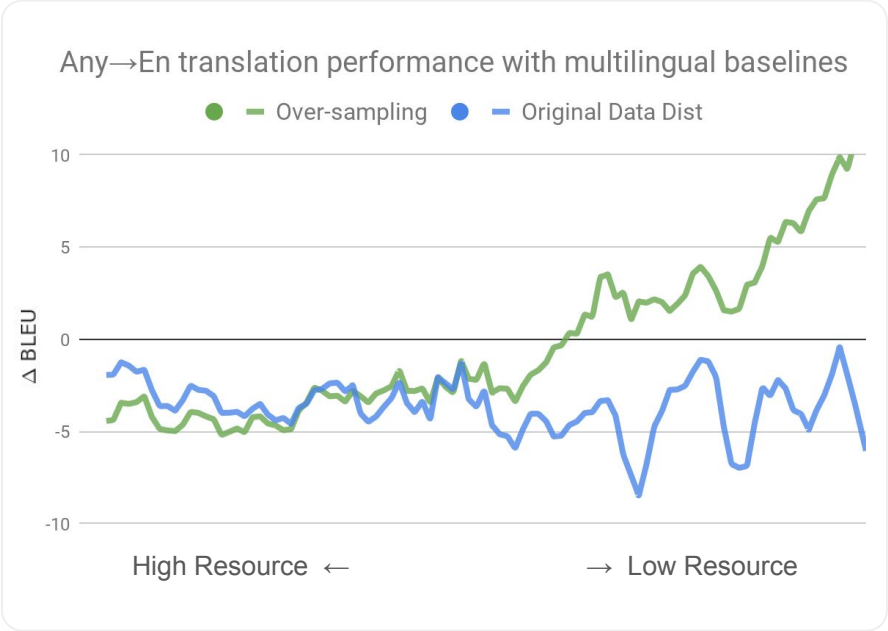


Any to English

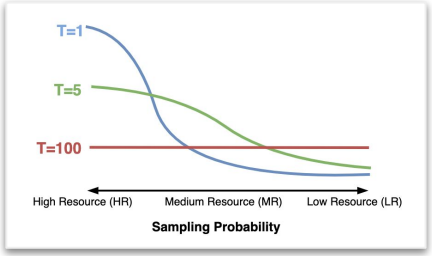


2/ Learn, given data imbalance:

Importance of re-balancing data

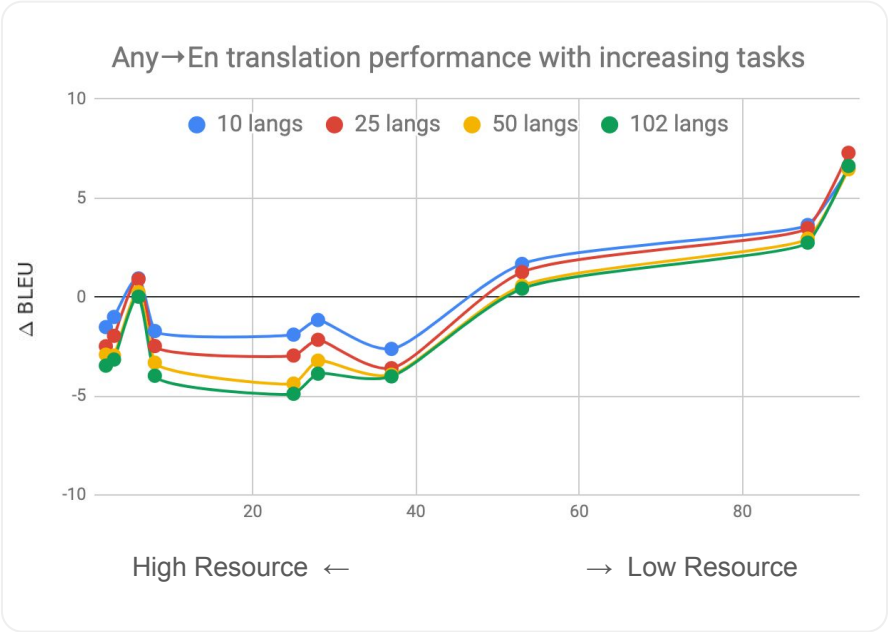


Any to English



2/ Learn, given data imbalance:

Increasing the number of languages at the same capacity results in worse quality due to interference, especially in En→Any



Any to English

To reduce quality losses, we needed greater model capacity

Relevant research questions:

- Do deep or wide networks drive greater quality gains?
- How deep or wide should we go?
- What quality boost do we attain by sacrificing En→Any (i.e. half of our tasks)?

Massive Neural Networks

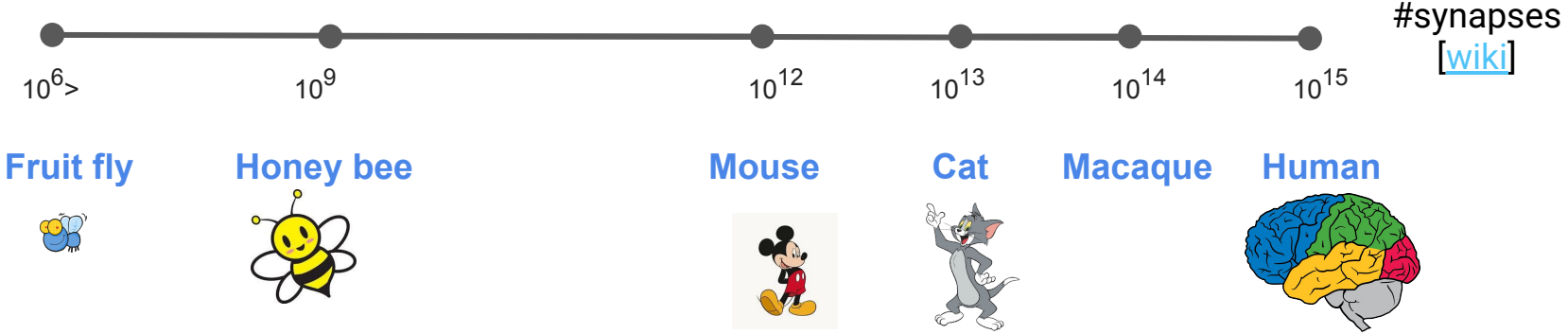
Training Deeper Neural Machine Translation Models with Transparent Attention, Bapna et al. EMNLP 2018

GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism, Huang et al. NeurIPS 2019

Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, Shazeer et al. ICLR 2017

Controlling Computation versus Quality for Neural Sequence Models, Bapna et al. 2020

Number of Synapses



Number of Synapses

NMT with Attention
Resnet50
[25-50M]

Fruit fly

Honey bee

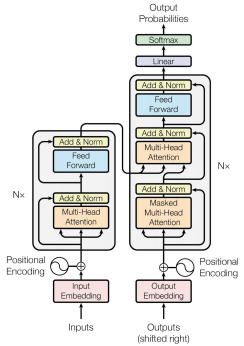
Mouse

Cat

Macaque

Human

Number of Synapses



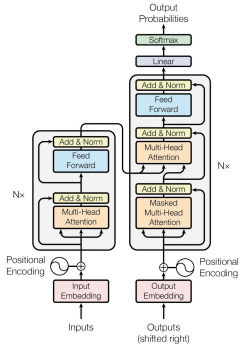
Transformer [400M]

10^{6}> 10^{9} 10^{12} 10^{13} 10^{14} 10^{15}

#synapses [wiki]

Fruit fly | Honey bee | Mouse | Cat | Macaque | Human

Number of Synapses



Facebook ResNeXt101
Open AI GPT2
MSR ZeRO
NVidia Megatron-LM
Google-T5
[1-10B]

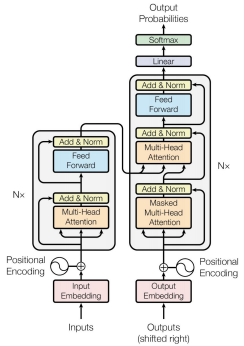
Transformer
[400M]

#synapses
[wiki]

10^{6}> | 10^{9} | 10^{12} | 10^{13} | 10^{14} | 10^{15}

Fruit fly | **Honey bee** | **Mouse** | **Cat** | **Macaque** | **Human**

Number of Synapses



Transformer
[400M]

Facebook ResNeXt101
Open AI GPT2
MSR ZeRO
NVidia Megatron-LM
Google-T5
[1-10B]

M4:
Massively Multilingual Massive Machine Translation
[80-600B]

10^{6}> 10^{9} 10^{12} 10^{13} 10^{14} 10^{15}

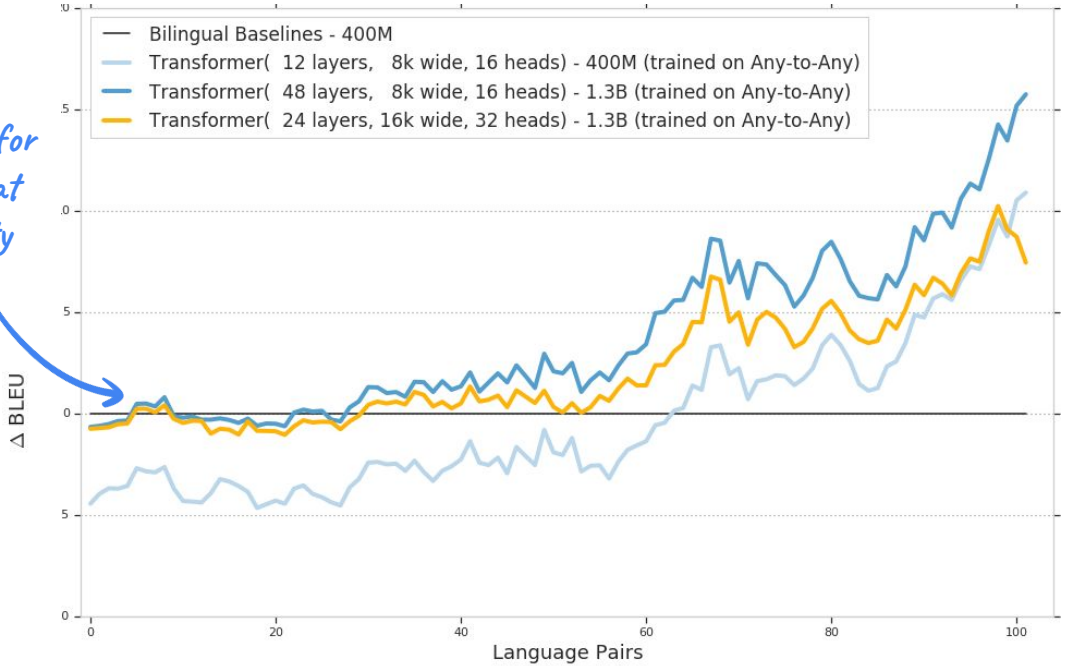
#synapses [wiki]

Fruit fly **Honey bee** **Mouse** **Cat** **Macaque** **Human**

3/ Increase model capacity:

Do deep or wide networks drive greater quality gains?

Any→En translation performance with model size



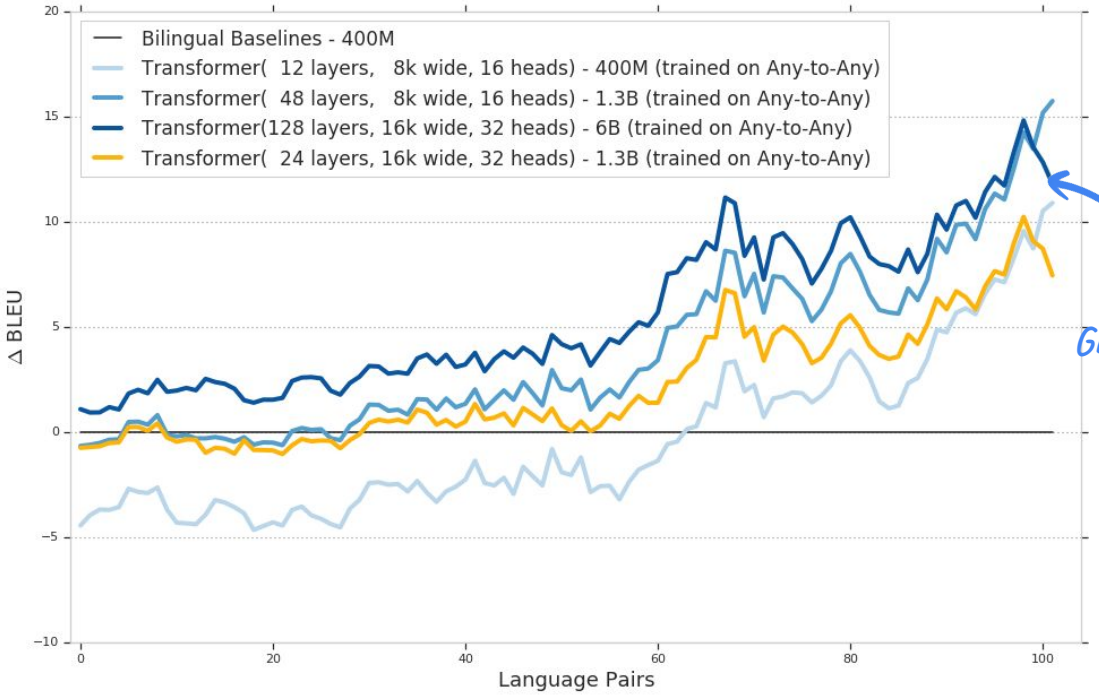
Higher quality for deep vs. wide at same capacity

← Bilingual Baselines

3/ Increase model capacity:

How deep or wide should we go?

All-to-English Translation Quality for M4 Models.



Greatest quality with 128-layer GPipe

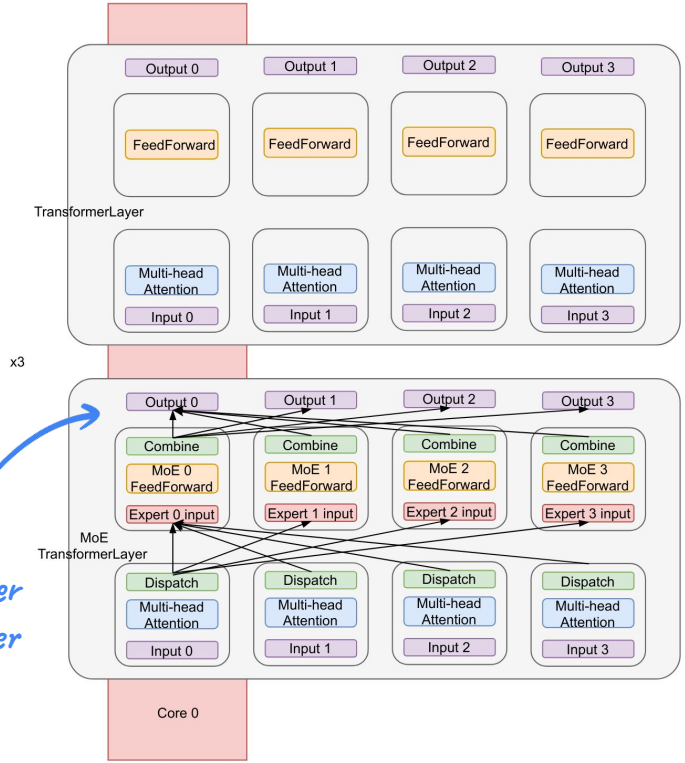
Dense Scaling

- GPipe allowed us to break single-TPU memory limit *efficiently* with pipelining.

Sparse Scaling

- ***Conditional computation***, where only part of the network is active for a given example became crucial after certain point.

How can we scale to 1k chips and beyond?
Conditional computation and Mixture-of-Experts with 50B+ weights



Replaces every other Transformer feed-forward layer with MoE

As we increase number of experts, training and inference scales sublinearly:

- FLOPS $\propto$ batch_size * avg. gated subnetwork size

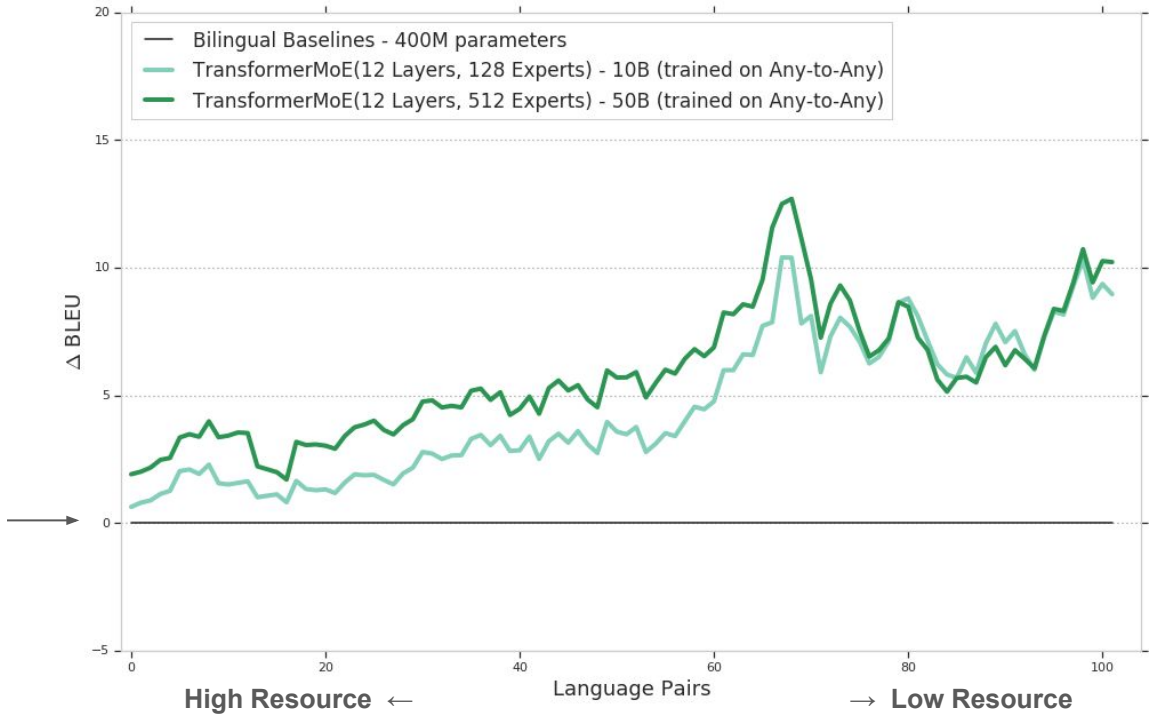
Tradeoffs of 512 token-level experts & 1 expert/core:

- Don't need MoE gradient aggregation
- MoE weight update broadcast is not required
- However, an extra network is required to dispatch activations

3/ Increase model capacity:

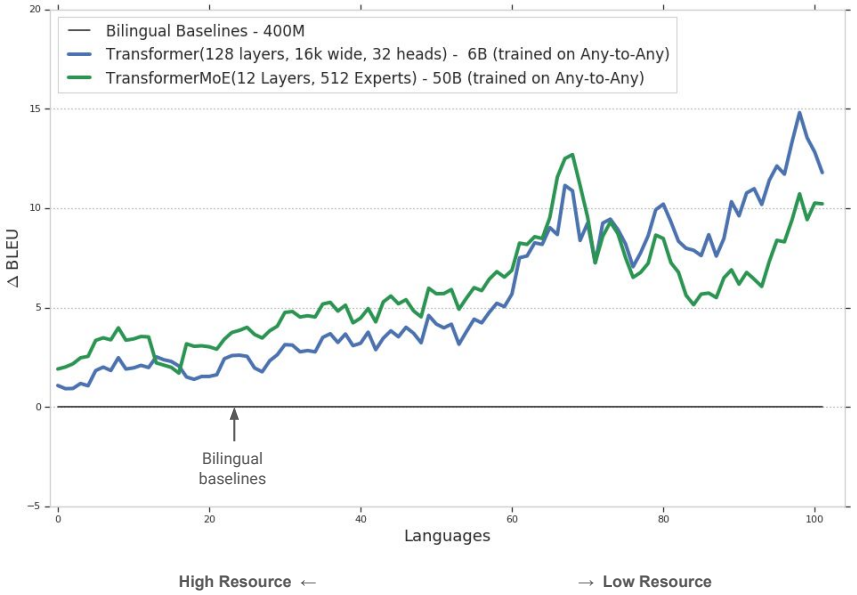
With MoE, bigger is better: 50B 512-MoE > 10B 128-MoE

Bilingual baselines

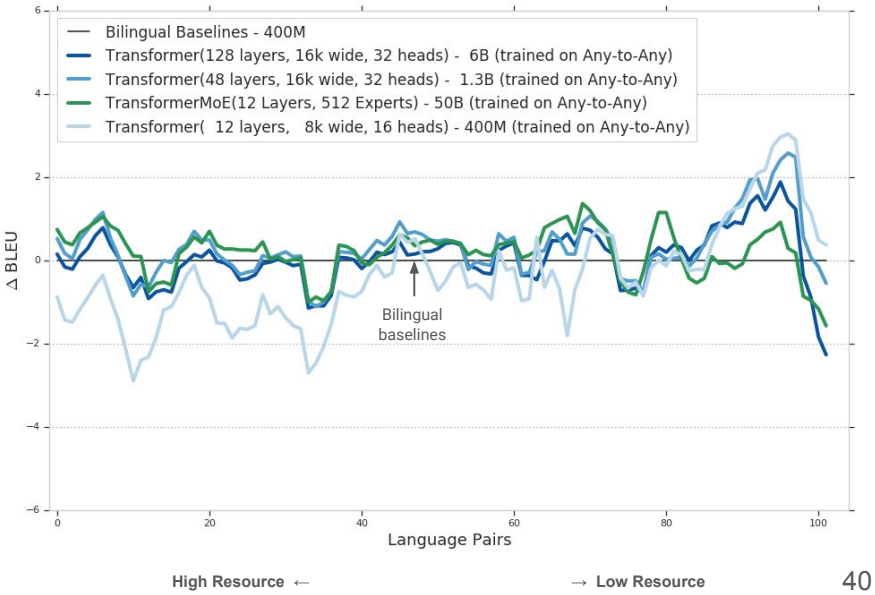


At the end of the first phase, two additional insights helped inform research questions for the next phases

1/ We need a future model that can get the "best of both worlds" across MoE and Deep Transformer



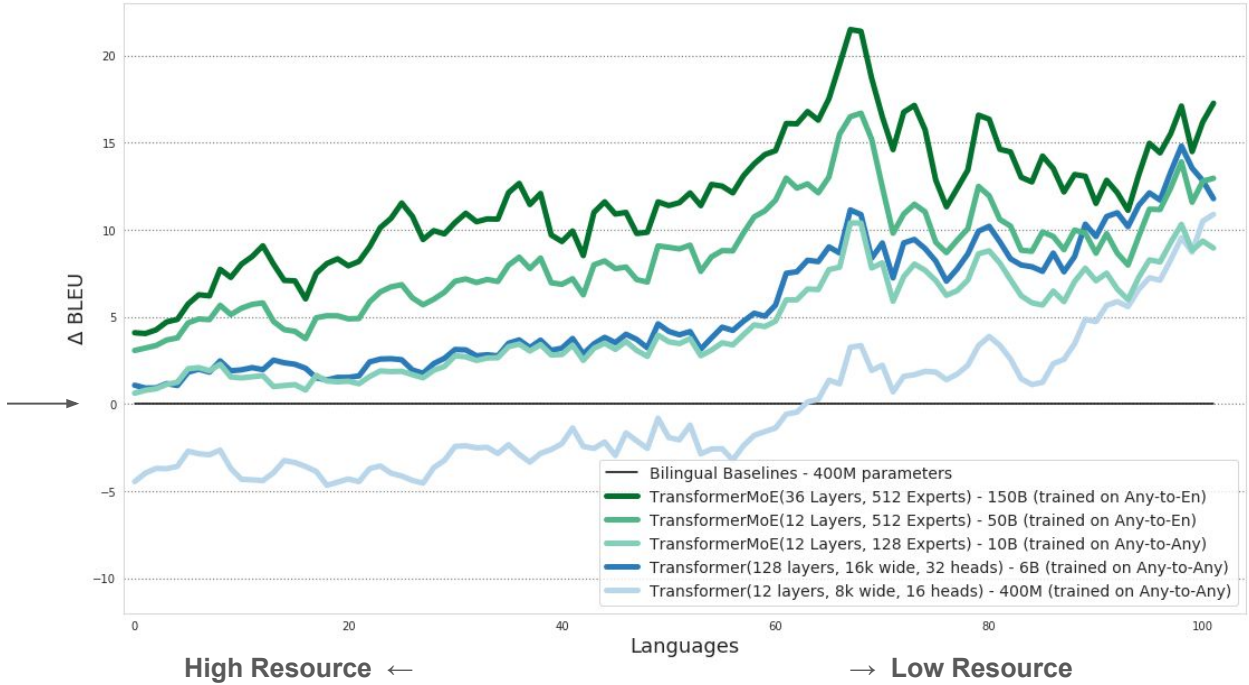
2/ Scaling up model capacity doesn't immediately improve performance on En→Any



3/ Increase model capacity:

Final Phase 1 results

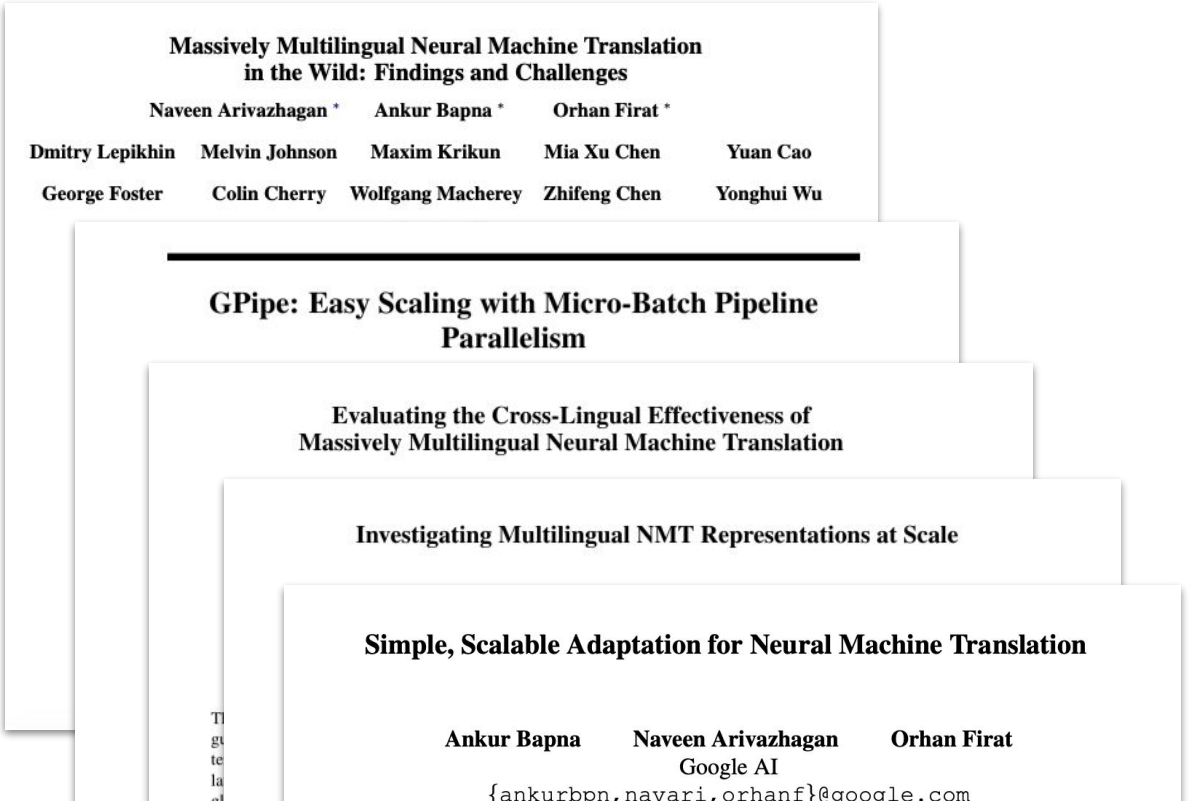
Bilingual baselines



Summary of recent publications

Conference Publications

- NeurIPS
- EMNLP
- AAAI



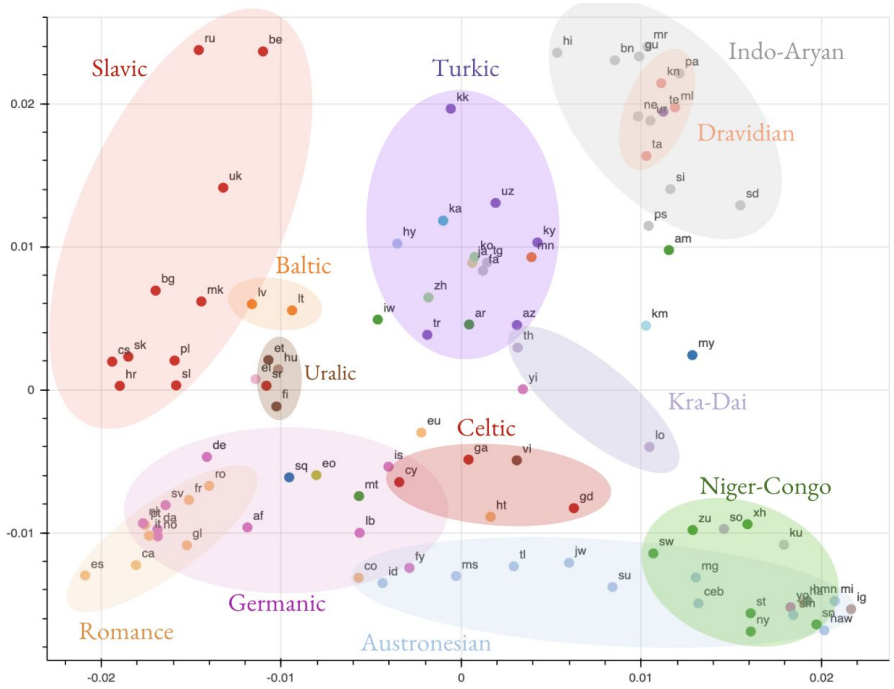
Drawing the Map of Languages

Investigating Multilingual NMT Representations at Scale, Kudugunta et al. - EMNLP'19

Under the hood ...

* Rexona commercial

If we plot M4 language representations, they cluster based on linguistic similarity

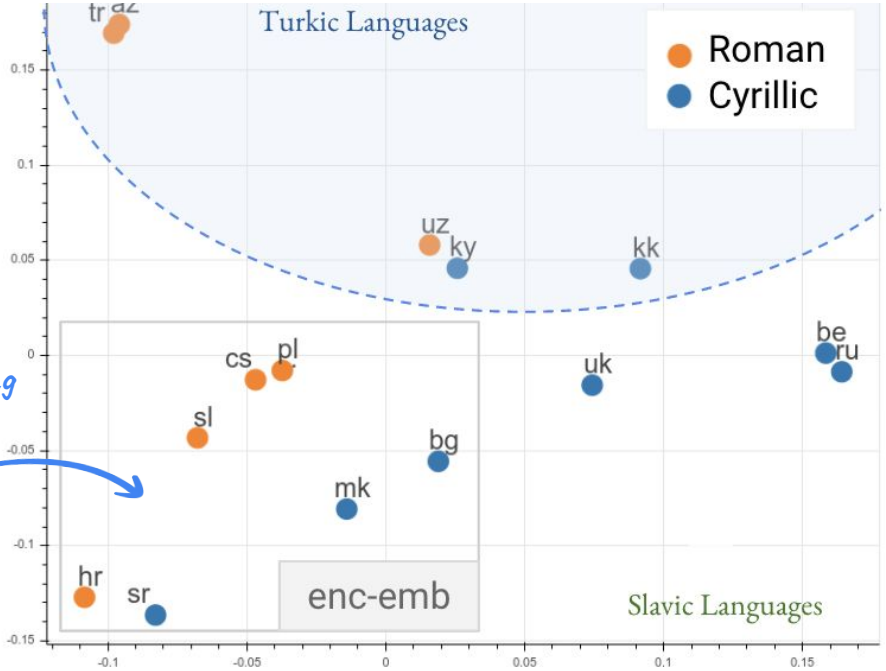


Representations of all languages (top encoder layer)

More details: Investigating Multilingual NMT Representations at Scale, Kudugunta et al. - EMNLP'19 [SLIDES]

Linguistic similarity determines the representation affinity, especially in higher layers

Representations of Slavic and Turkic languages with Roman and Cyrillic scripts

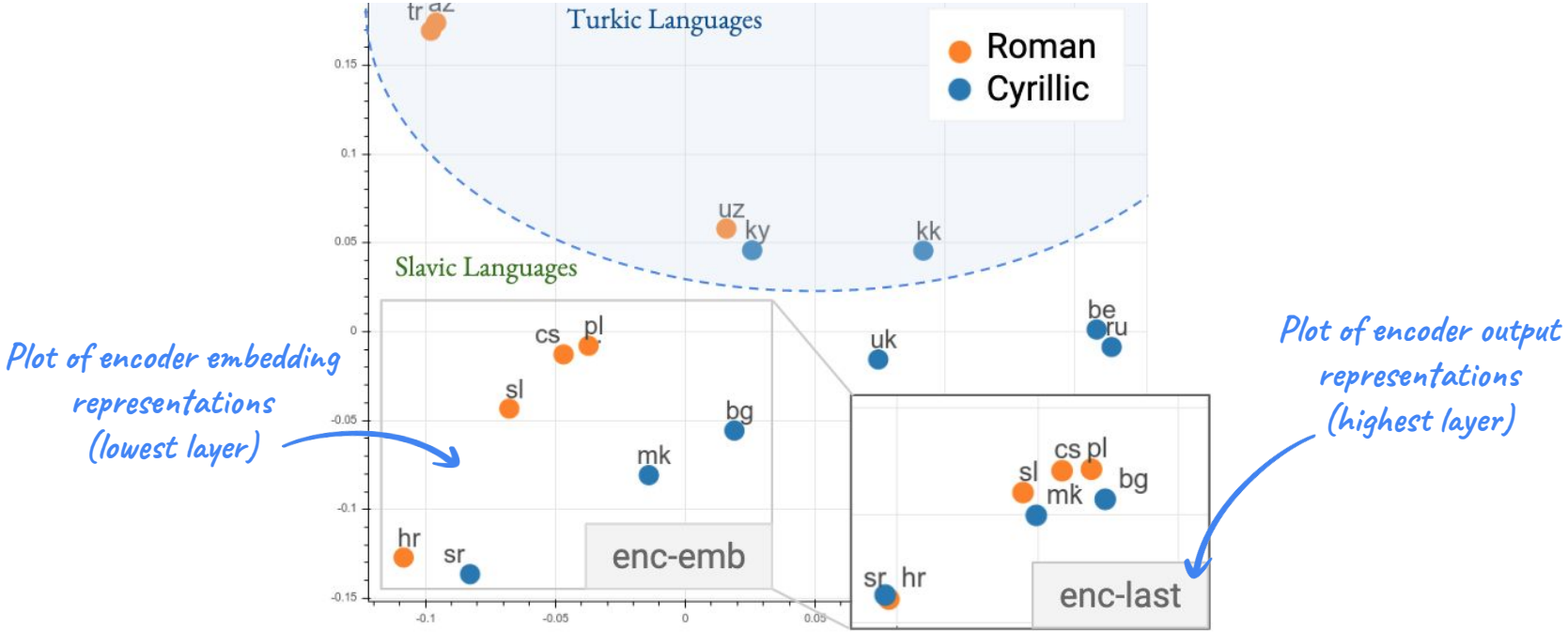


Plot of encoder embedding representations (lowest layer)

More details: Investigating Multilingual NMT Representations at Scale, Kudugunta et al. - EMNLP'19 [SLIDES]

Linguistic similarity determines the representation affinity, especially in higher layers

Representations of Slavic and Turkic languages with Roman and Cyrillic scripts



More details: Investigating Multilingual NMT Representations at Scale, Kudugunta et al. - EMNLP'19 [SLIDES]

Takeaways

1. *Which factors determine the extent of overlap in the learned representations?*
   Linguistic similarity, not just lexical overlap.

2. *Is the extent of representational overlap similar throughout the model?*
   It changes, depending on the type of task.

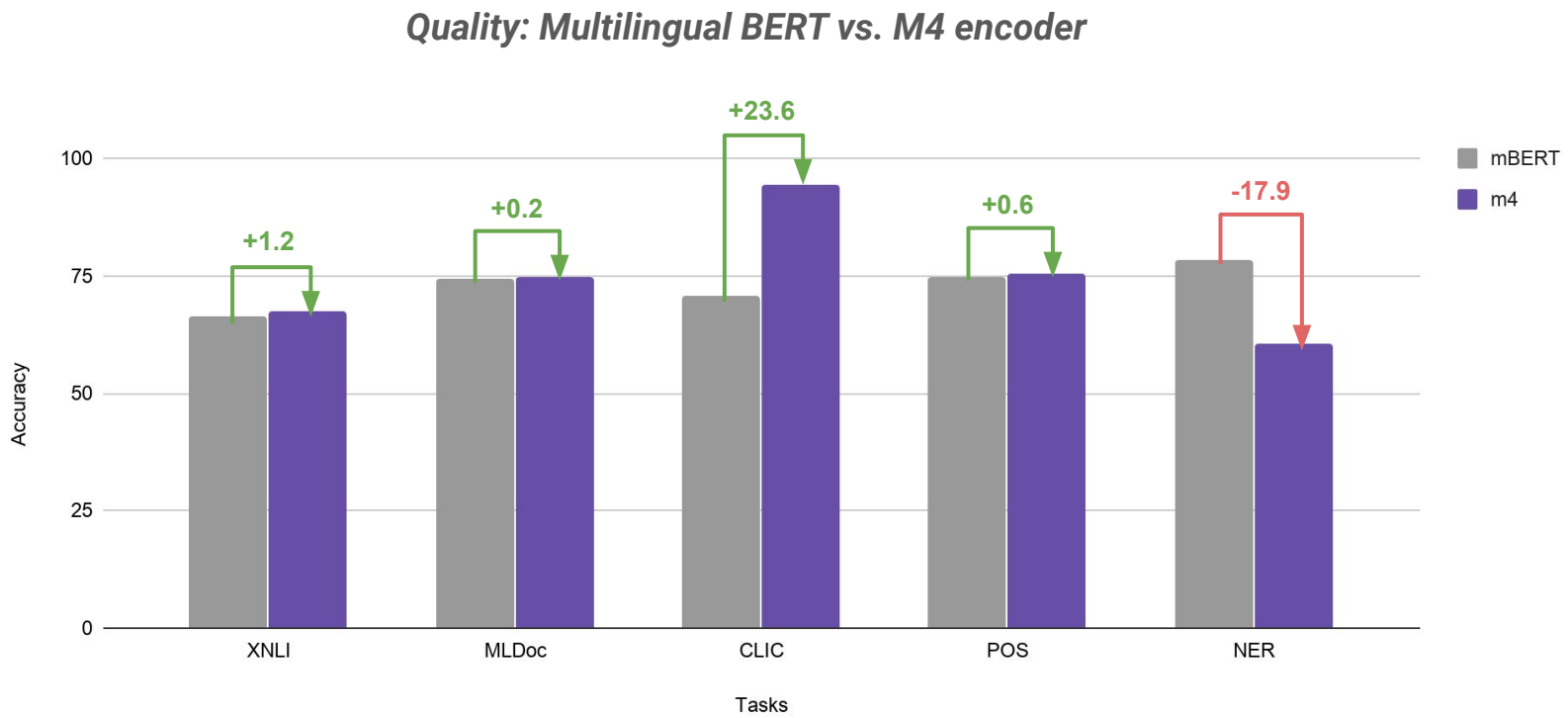
3. *How robust are multilingual NMT representations?*
   (to fine-tuning on an another language)
   Depends on resource size and linguistic similarity to finetuning language.

Cross-lingual Downstream Transfer

Evaluating the Cross-Lingual Effectiveness of Massively Multilingual Neural Machine Translation, Siddhant et al. AAAI 2020

XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization, Hu et al. ICML 2020

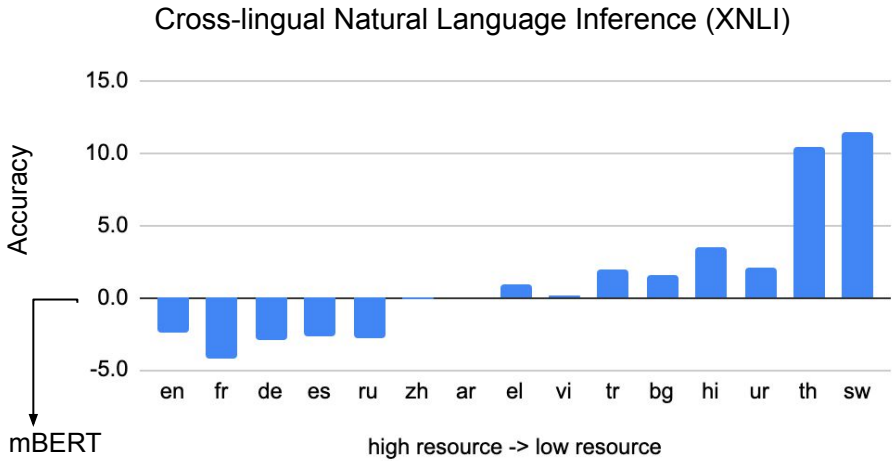
We evaluated M4 representations on downstream tasks; M4 encoder worked better than multilingual BERT in 4 out of 5 tasks

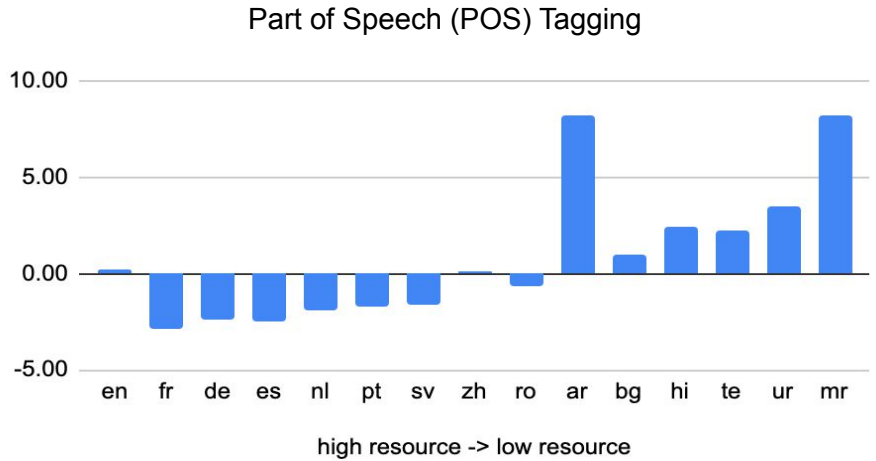


More details: Evaluating the Cross-Lingual Effectiveness of Massively Multilingual Neural Machine Translation, Siddhant et al. - AAAI 2020

M4 representations transfer to low-resource languages better than to high-resource languages

Quality: Multilingual BERT vs. M4 encoder





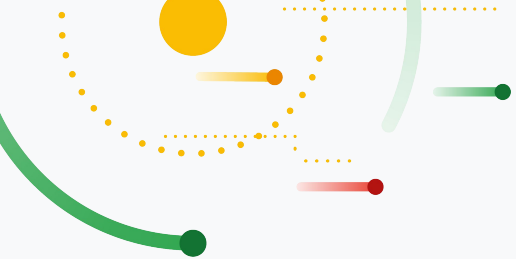
Hypotheses why transfer is better for low-resource languages:

- Translation to English from all languages implicity forces their representation to be in the same space
- High resource languages are harder to move from point of convergence on fine-tuning

Next, on more tasks and languages (9 tasks, 40 languages)
XTREME Benchmark: sites.research.google/xtreme

Model	Avg	Pair sentence		Structured prediction		Question answering			Sentence retrieval	
		XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA-GoldP	BUCC	Tatoeba
Metrics		Acc.	Acc.	F1	F1	F1 / EM	F1 / EM	F1 / EM	F1	Acc.
Cross-lingual zero-shot transfer (models are trained on English data)										
mBERT	59.6	65.4	81.9	70.3	62.2	64.5 / 49.4	61.4 / 44.2	59.7 / 43.9	56.7	38.7
XLM	55.5	69.1	80.9	70.1	61.2	59.8 / 44.3	48.5 / 32.6	43.6 / 29.1	56.8	32.6
XLM-R Large	68.1	79.2	86.4	72.6	65.4	76.6 / 60.8	71.6 / 53.2	65.1 / 45.0	66.0	57.3
MMTE	59.3	67.4	81.3	72.3	58.3	64.4 / 46.2	60.3 / 41.4	58.1 / 43.8	59.8	37.9
Translate-train (models are trained on English training data translated to the target language)										
mBERT	-	74.0	86.3	-	-	70.0 / 56.0	65.6 / 48.0	55.1 / 42.1	-	-
mBERT, multi-task	-	75.1	88.9	-	-	72.4 / 58.3	67.6 / 49.8	64.2 / 49.3	-	-
Translate-test (models are trained on English data and evaluated on target language data translated to English)										
BERT-large	-	76.5	84.4	-	-	76.3 / 62.1	72.9 / 55.3	72.1 / 56.0	-	-
In-language models (models are trained on the target language training data)										
mBERT, 1000 examples	-	-	-	87.6	77.9	-	-	58.7 / 46.5	-	-
mBERT	-	-	-	89.8	88.3	-	-	74.5 / 62.7	-	-
mBERT, multi-task	-	-	-	91.5	89.1	-	-	77.6 / 68.0	-	-
Human	-	92.8	97.5	97.0	-	91.2 / 82.3	91.2 / 82.3	90.1 / -	-	-

More details: XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization, Hu et al. ICML 202



Google AI

On training and hyper-parameters

Adaptive Schedules have the potential to overcome task-scheduling challenges

- Meta-Learning (for parallel transfer)
  - Learning to Schedule Tasks
    - Learning Task Weights [paper]
  - Explicit schedules: not scalable

$$w_i = 1 \Big/ \left(\min\left(1, \frac{s_i}{b_i}\right)^{\alpha} + \epsilon \right)$$

  - Implicit schedules: coupled with optimization

$$w_i = 1 + (\text{sign}\,(\overline{S} - S_i)) \min\left(\gamma, \left(\max_j S_j\right)^{\alpha} |S_i - \overline{S}|^{\beta}\right)$$

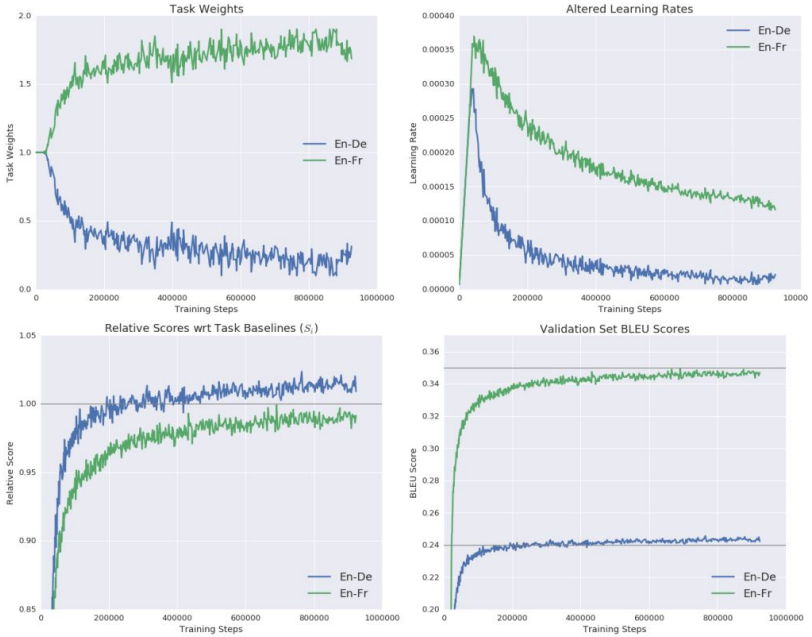


Figure 5: Implicit Validation-Based Scheduling Progress.

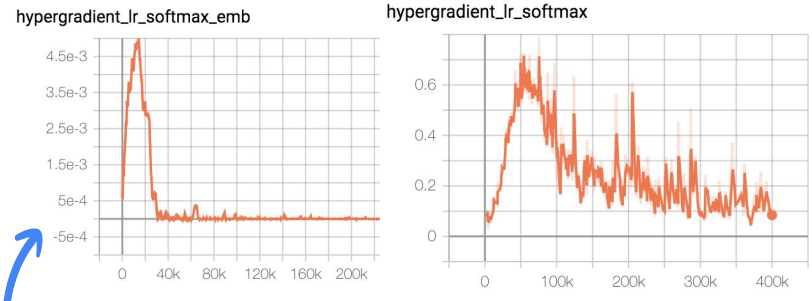
If we meta-learn some hyper-parameters, they don't require further tuning at scale

- Apply gradient descent on the learning rate (+underlying optimizer)

$$\frac{\partial f(\theta_{t-1})}{\partial \alpha} = \nabla f(\theta_{t-1}) \cdot \frac{\partial (\theta_{t-2} - \alpha \nabla f(\theta_{t-2}))}{\partial \alpha} = \nabla f(\theta_{t-1}) \cdot (-\nabla f(\theta_{t-2}))$$

$$\alpha_t = \alpha_{t-1} - \beta \frac{\partial f(\theta_{t-1})}{\partial \alpha} = \alpha_{t-1} + \beta \nabla f(\theta_{t-1}) \cdot \nabla f(\theta_{t-2})$$

- Comparison
  - Single pair (wmt'19 en-de): HG ~ Baseline
  - Multi-task (wmt en-{de,fr}): HG > Baseline
  - BERT: HG ~ Baseline



Learnt learning rate schedules (per-layer)

Hypergradients, Baydin et al. ICLR 2017

We also summarized Phase 1 findings in a Google AI blog post (link)

Google AI Blog

The latest news from Google AI

Exploring Massively Multilingual, Massive Neural Machine Translation

Friday, October 11, 2019

Posted by Ankur Bapna, Software Engineer and Orhan Firat, Research Scientist, Google Research

"... perhaps the way [of translation] is to descend, from each language, down to the common base of human communication — the real but as yet undiscovered universal language — and then re-emerge by whatever particular route is convenient." — Warren Weaver, 1949

Challenges & Open Problems

Cross-lingual Downstream Transfer Learning

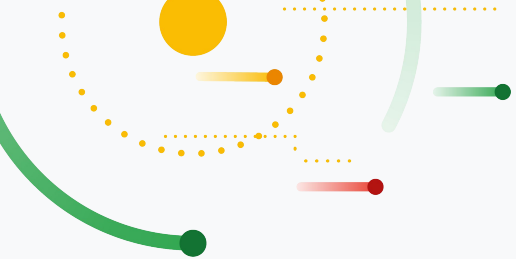
- Better learning objectives
- New tasks and languages

Unsupervised Machine Translation

- M4 as a knowledge base for unsupervised MT
- Adapting to unseen languages (modalities)

Transfer Learning

- Parallel transfer: smarter schedulers, mitigating interference
- Serial transfer: continual/meta learning, maximize transfer w/o forgetting



Next 1000 Languages

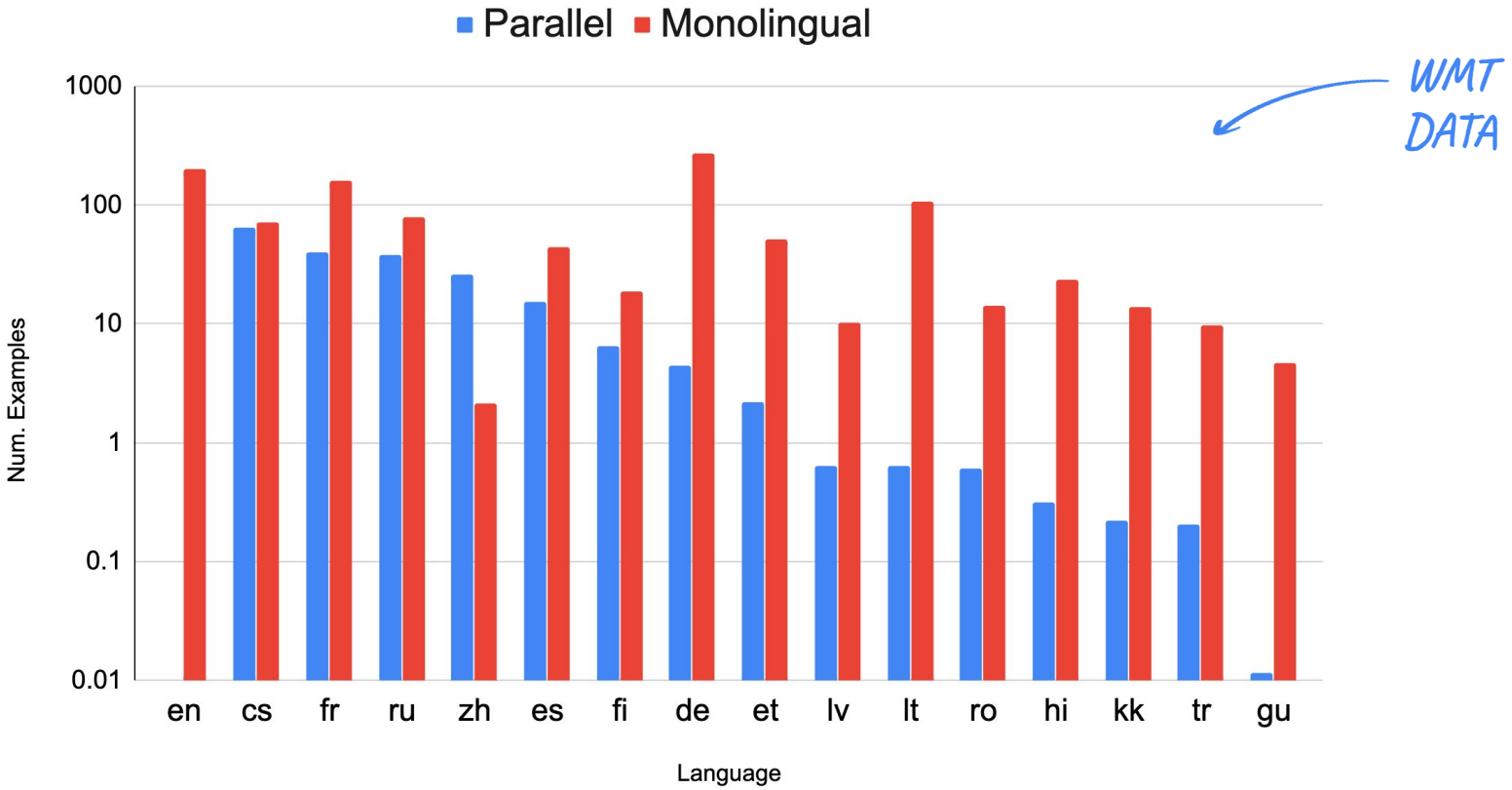
Towards universal translation

Context on data challenges

- **Multilingual** models disproportionately boost low-resource quality, getting close to so-called “interlingua”
- There’s also been tremendous progress on **unsupervised** MT, but deficiencies remain (e.g., domain mismatch, distant languages)
- We need the **confluence** of multilingual and unsupervised MT research to build a universal translation model
- But first, we need the **right data**, or just **some data**!

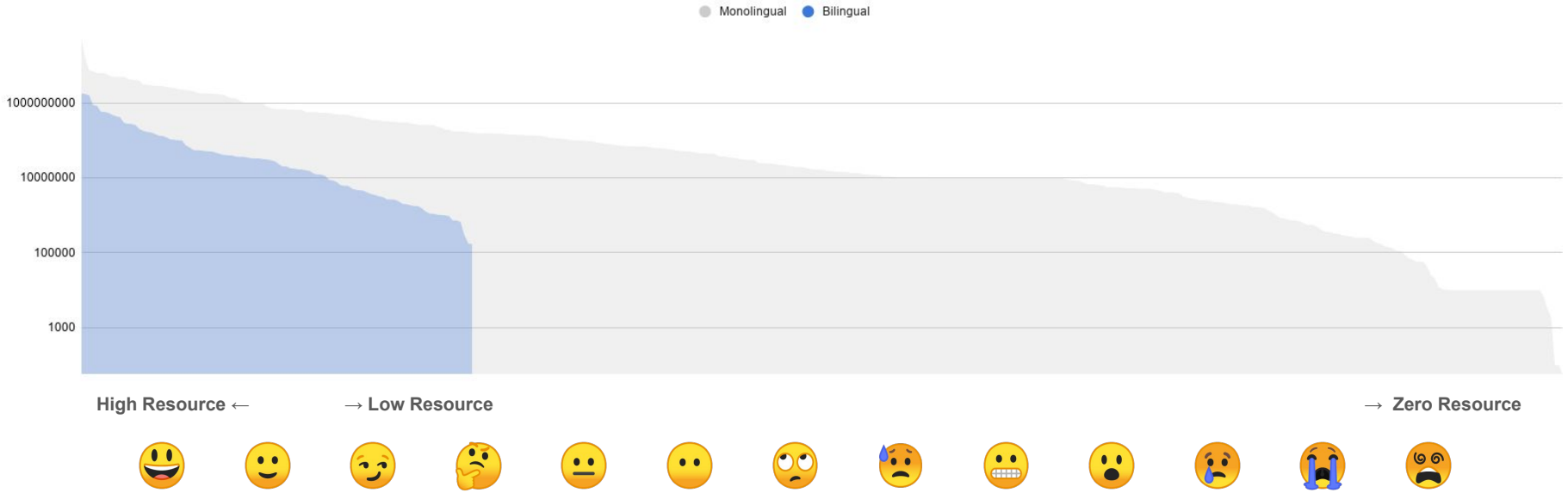
Google AI

As we move beyond 100 languages, we journey over supervised, semi-supervised and unsupervised learning



Google AI

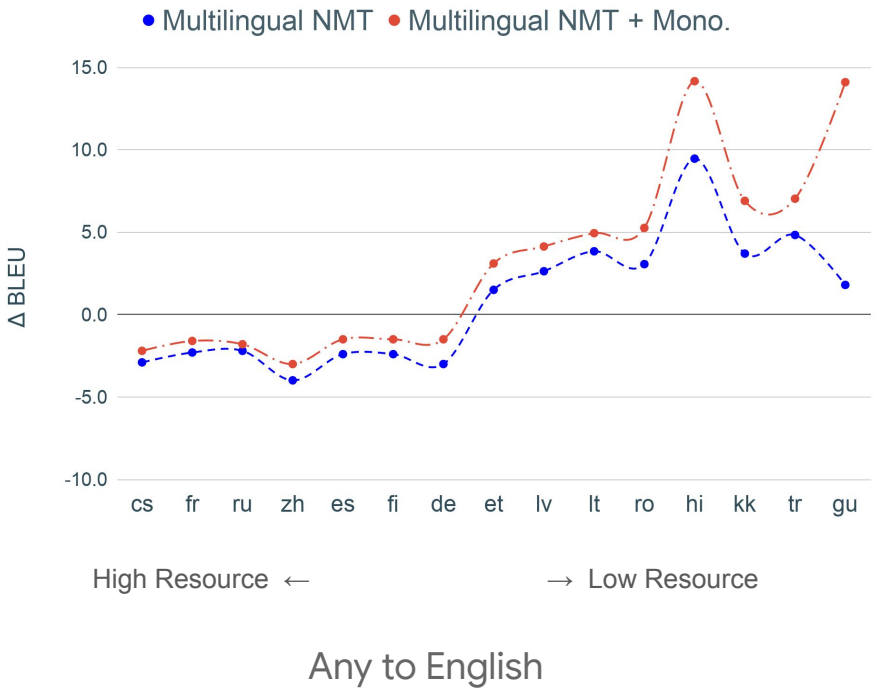
As we move beyond 100 languages, we journey over supervised, semi-supervised and unsupervised learning

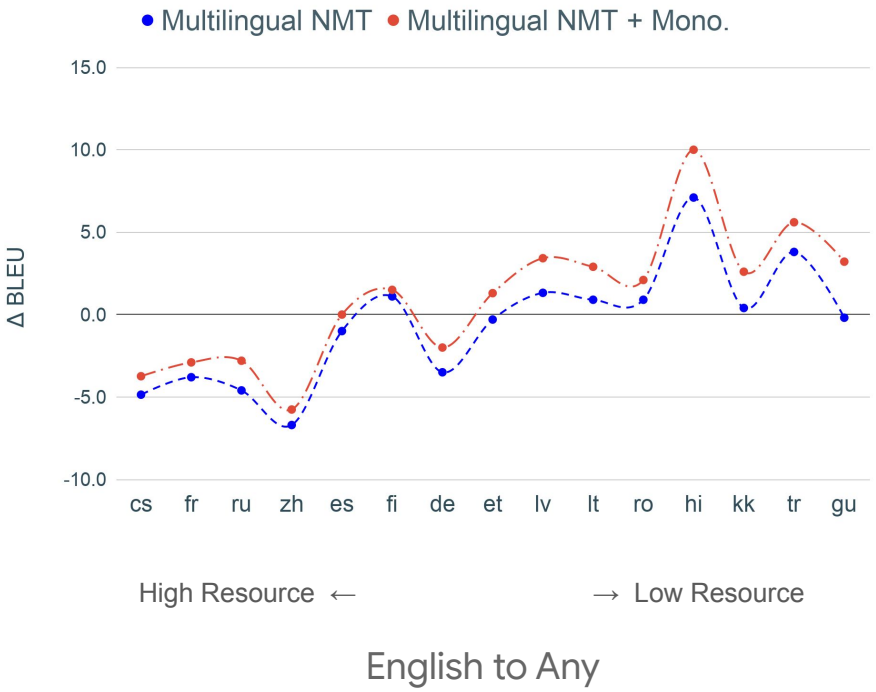


😃 🙂 😏 🤔 😐 😶 🙄 😟 😬 😮 😢 😭 😩

😉 : *Symbiosis between supervised and unsupervised MT*

Solution: Supervised + Self-supervised loss simultaneously, boosting quality even without iterative back-translation



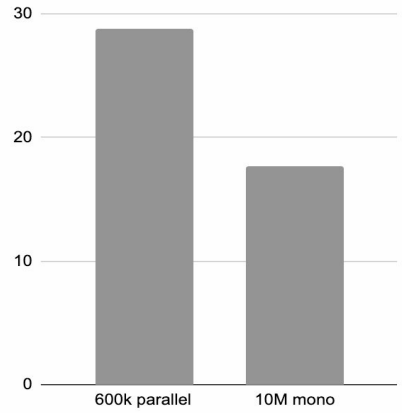


More details: Leveraging Monolingual Data with Self-Supervision for Multilingual Neural Machine Translation, Sidhant et al. - ACL'20

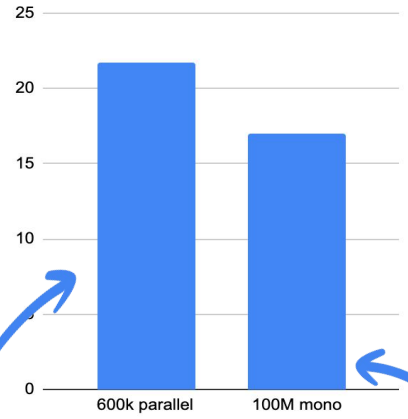
Google AI

Combining **Supervised** and **Self-Supervised** Learning, enables M4 model to learn new unseen languages **with monolingual data only**

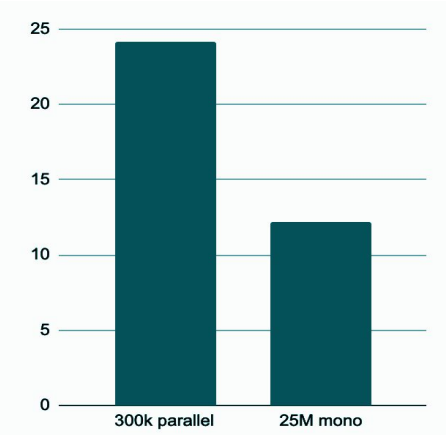
BLEU scores



Lithuanian → English



Latvian → English



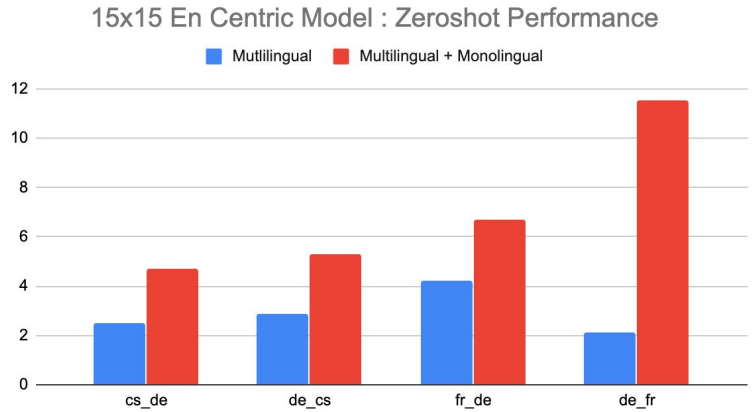
Hindi → English

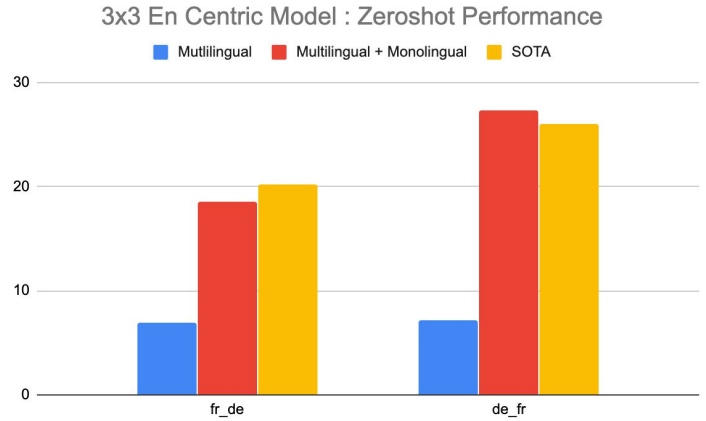
Multilingual Model with Latvian

Multilingual Model W/O Latvian + monolingual Latvian data

Google AI

Adding Monolingual data also enhances Zero-shot MT on Non-English Centric Language Pairs too





Bonus Dataset: M4-Public

About to release a benchmark on:

- Compilation of public datasets - **125 languages**
  - Europarl, Paracrawl, Ted57, JW300, Newscrawl, ...
  - Diverse set of languages, coverage
  - Multi-domain, noisy, imbalanced
  - 450M parallel data
- A new evaluation set - **60 languages**
  - Professional translations
  - Multi-way parallel
  - Web-domain
- Enables studying
  - Representation analysis
  - Out-of-domain generalization
  - Learning dynamics of massively multi-task models
  - Supervised, semi-supervised and unsupervised MT

Agenda

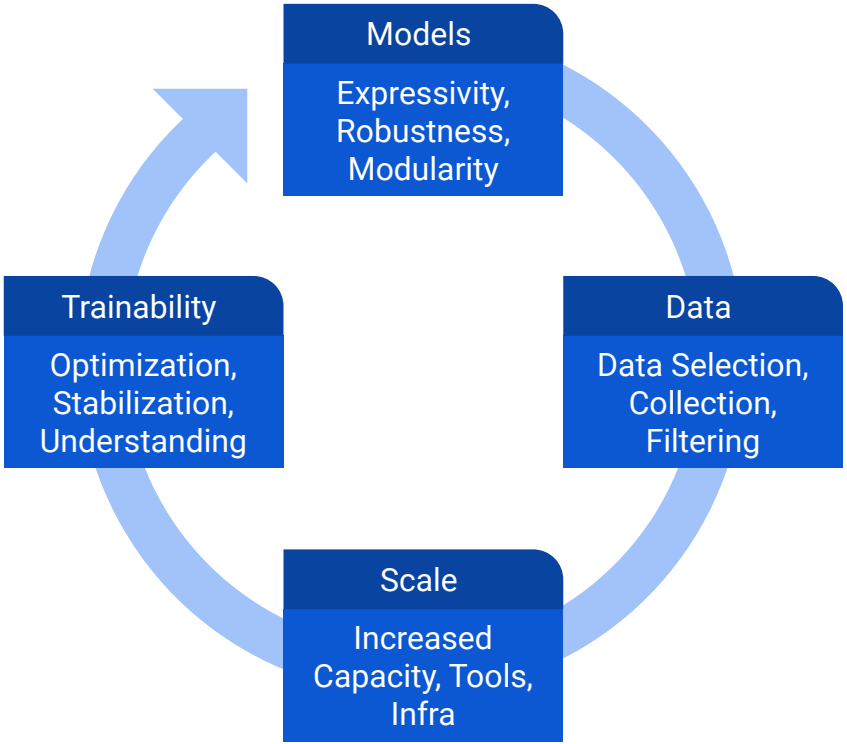
1. Motivation
2. Project Phases
3. **Lessons Learned**

Google AI

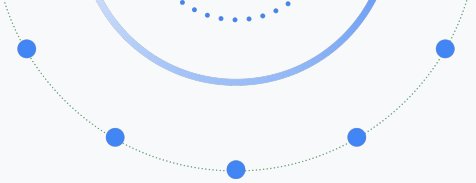
Workflow

Google AI

Workflow of AI Researchers working at Scale



Tools and Systems in order to orchestrate a giant machinery is essential for large scale ML projects



Google AI

Components of a ML System

Training

- Reads the data, computes loss and gradients, applies parameter update.
- The most compute intensive job, runs on TPU

Inference

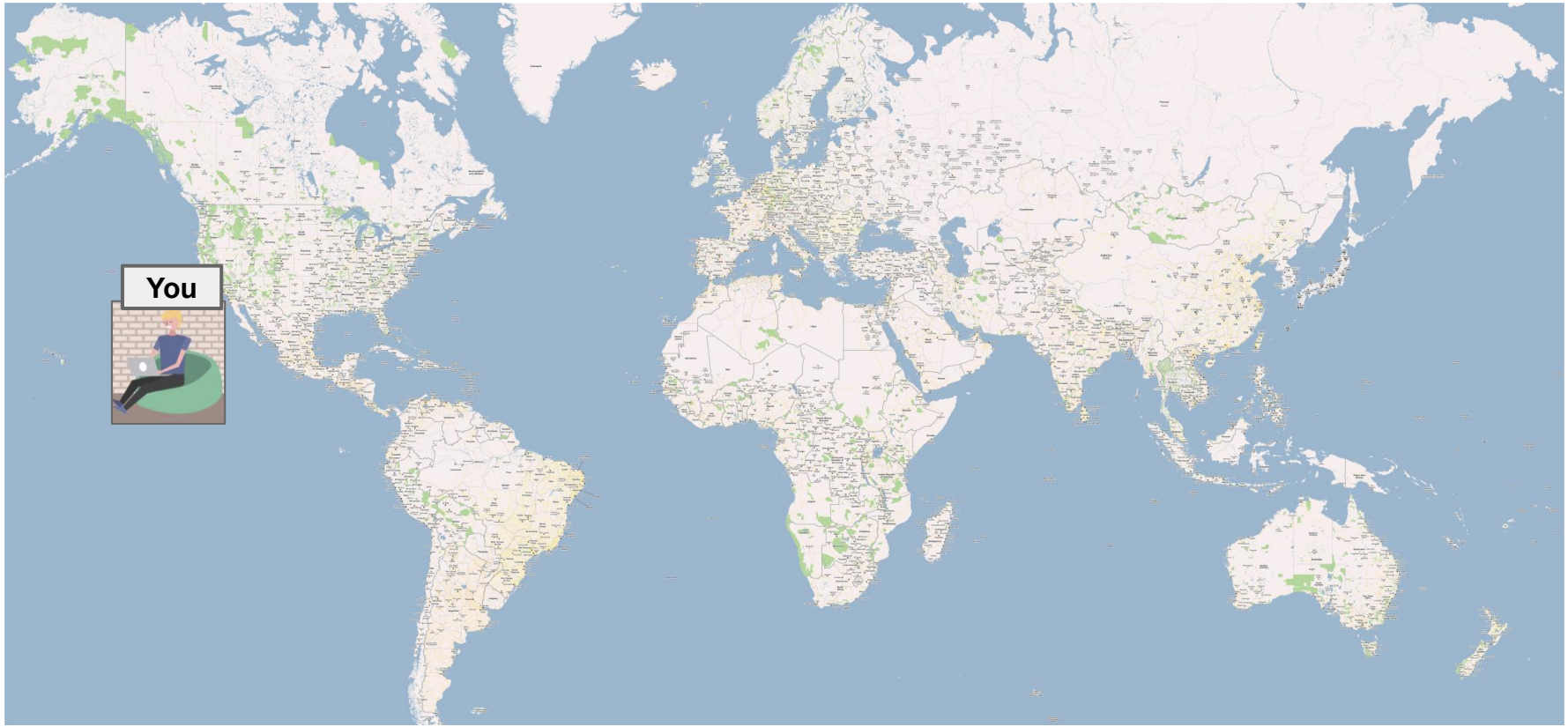
- Reads a checkpoint of the trained model, runs inference (beam-search)
- Generating output sequences, usually runs on GPU

Evaluation

- Reads a checkpoint of the trained model, computes loss on dev set.
- Used for monitoring the progress, usually runs on GPU or CPU

Under the hood

Google AI



*This is just a sketch, exact locations are inaccurate.

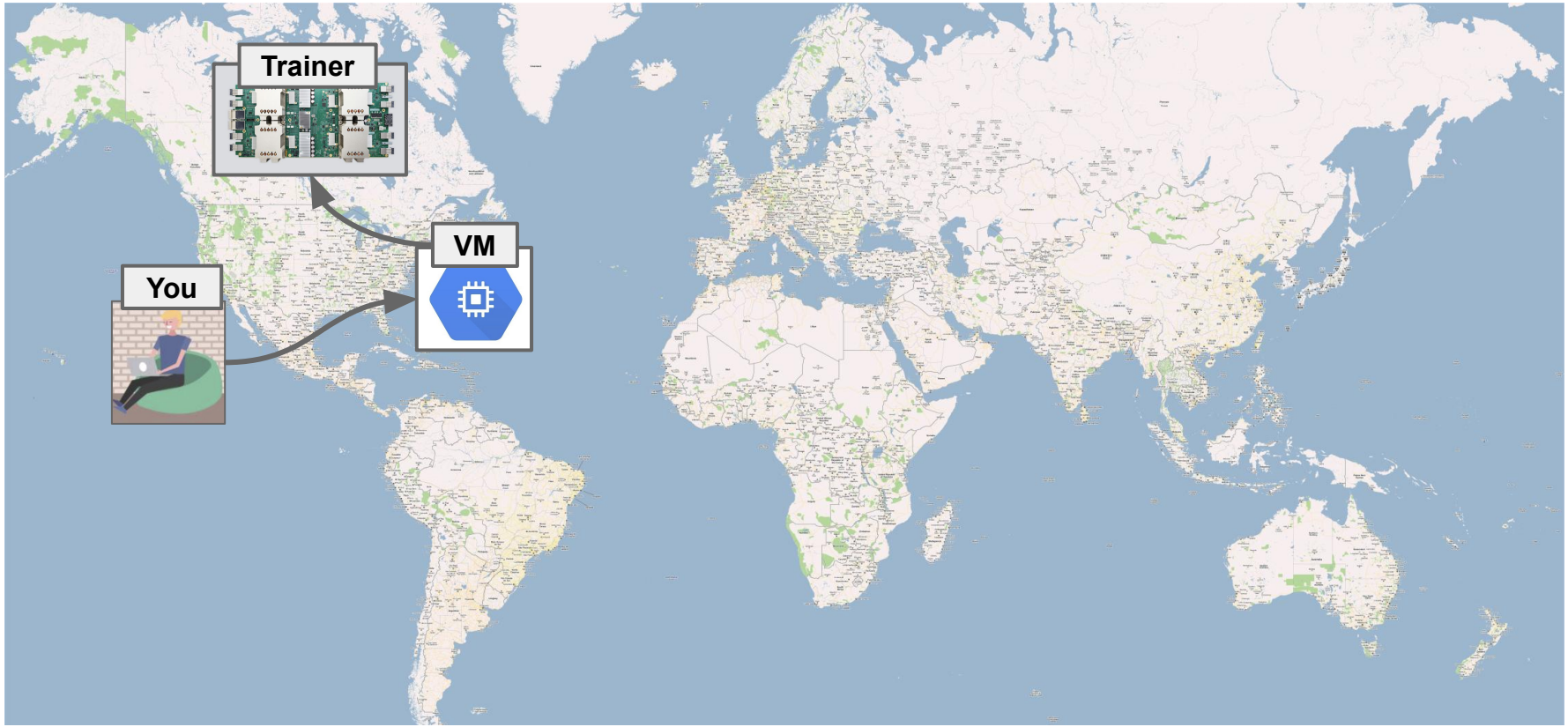
Under the hood

Google AI

*This is just a sketch, exact locations are inaccurate.

Under the hood

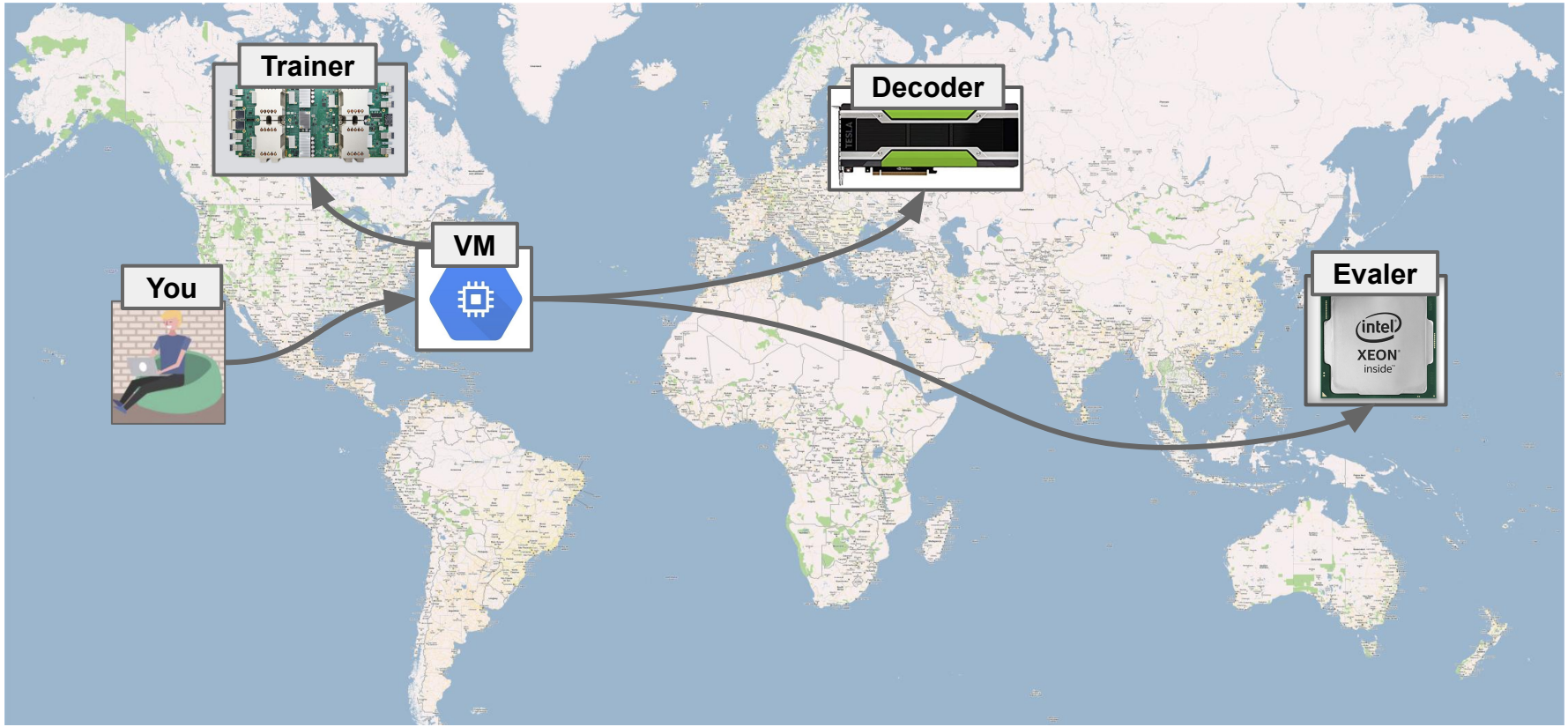
Google AI



*This is just a sketch, exact locations are inaccurate.

Under the hood

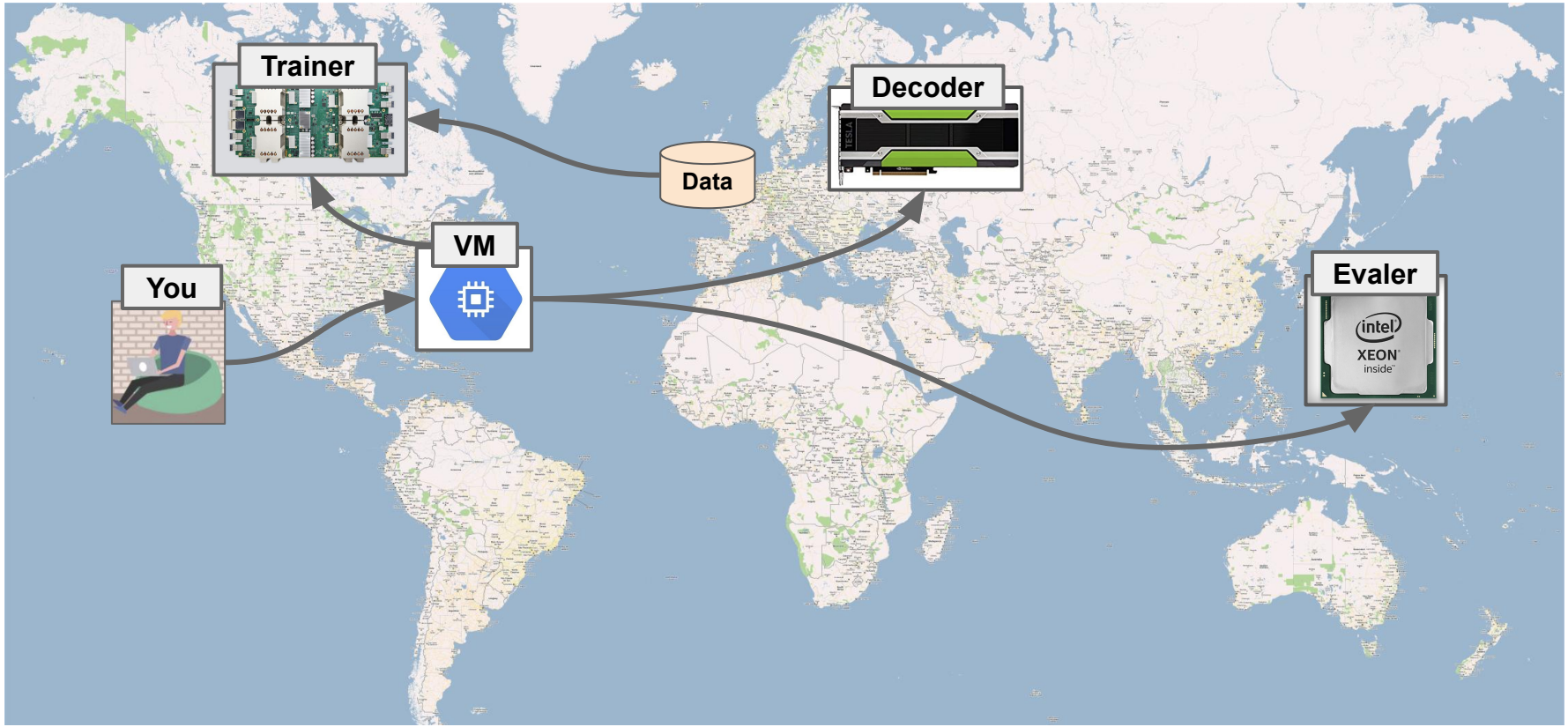
Google AI



*This is just a sketch, exact locations are inaccurate.

Under the hood

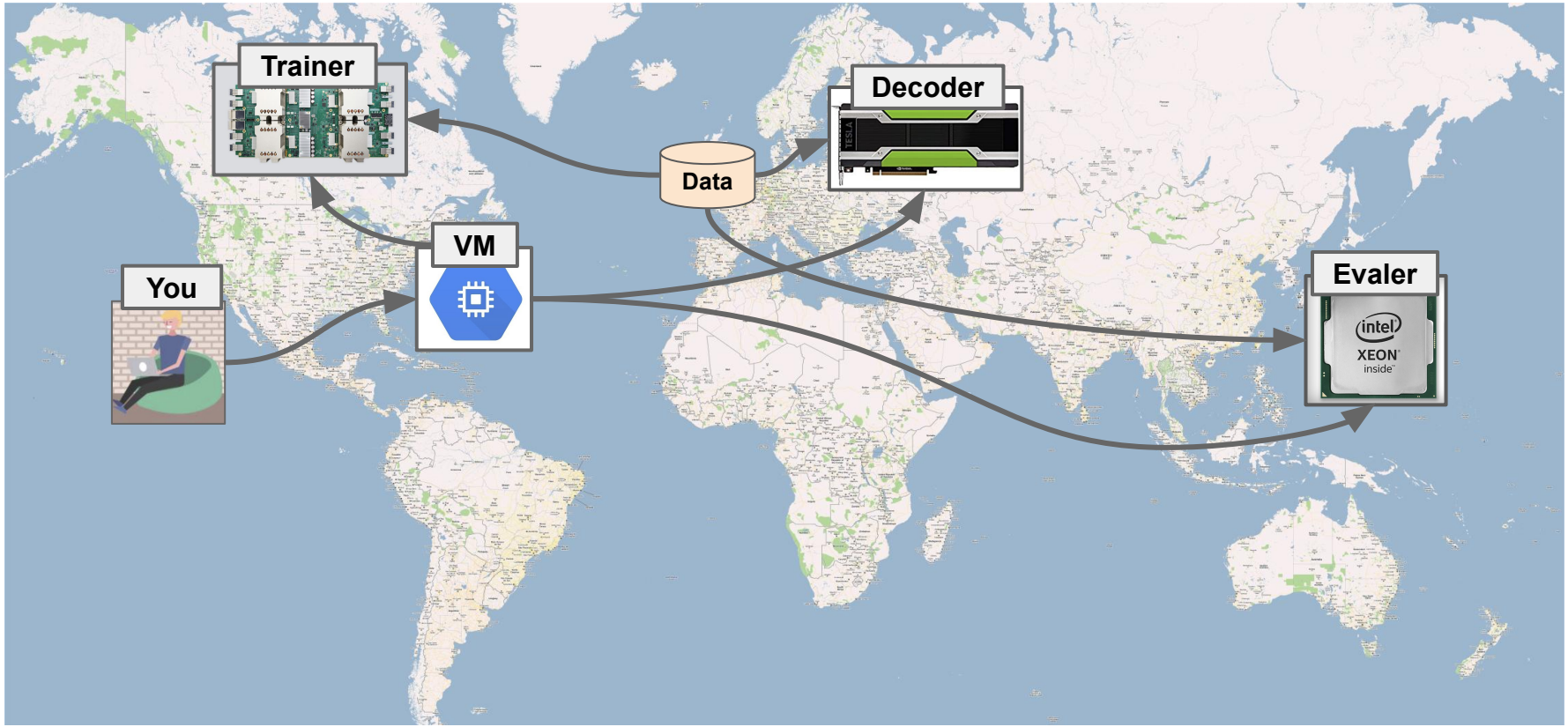
Google AI



*This is just a sketch, exact locations are inaccurate.

Under the hood

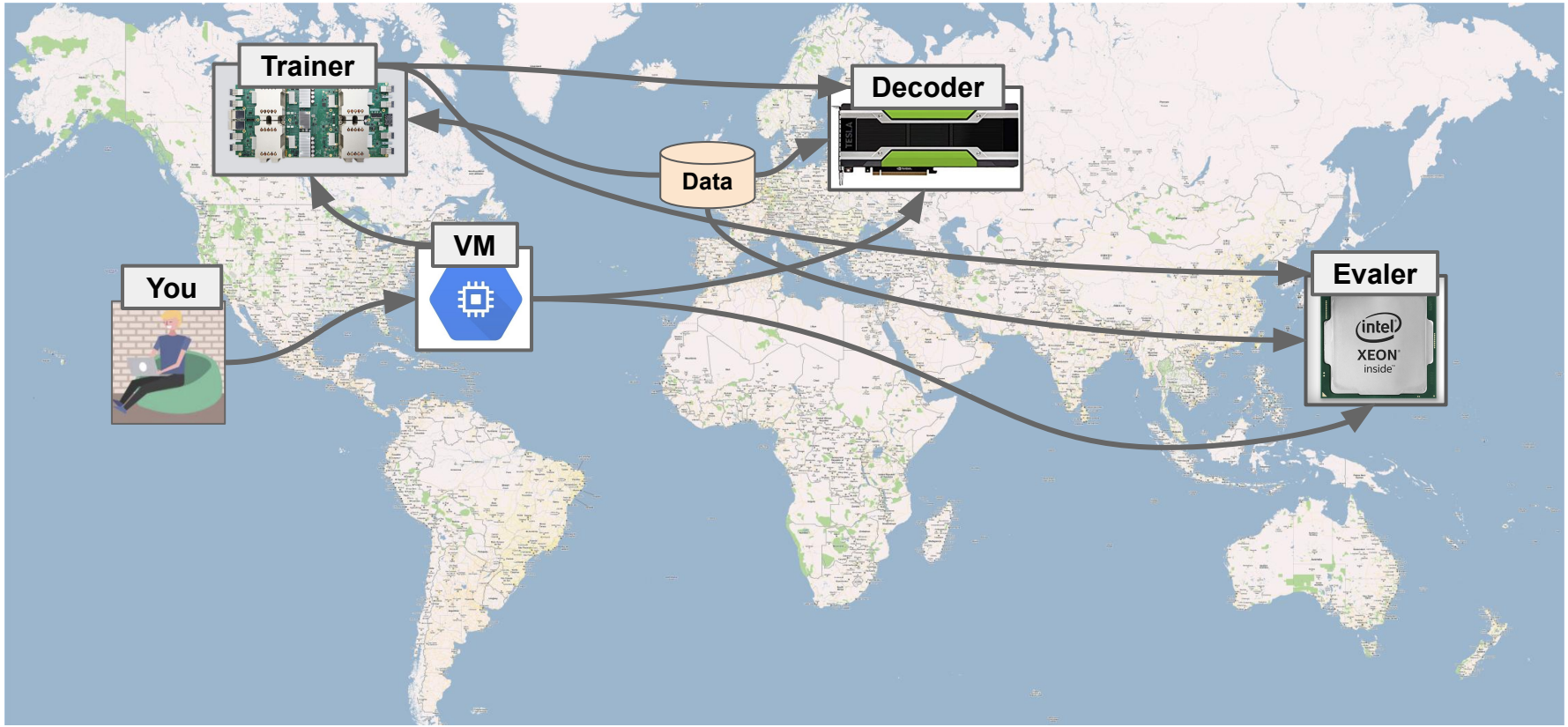
Google AI



*This is just a sketch, exact locations are inaccurate.

Under the hood

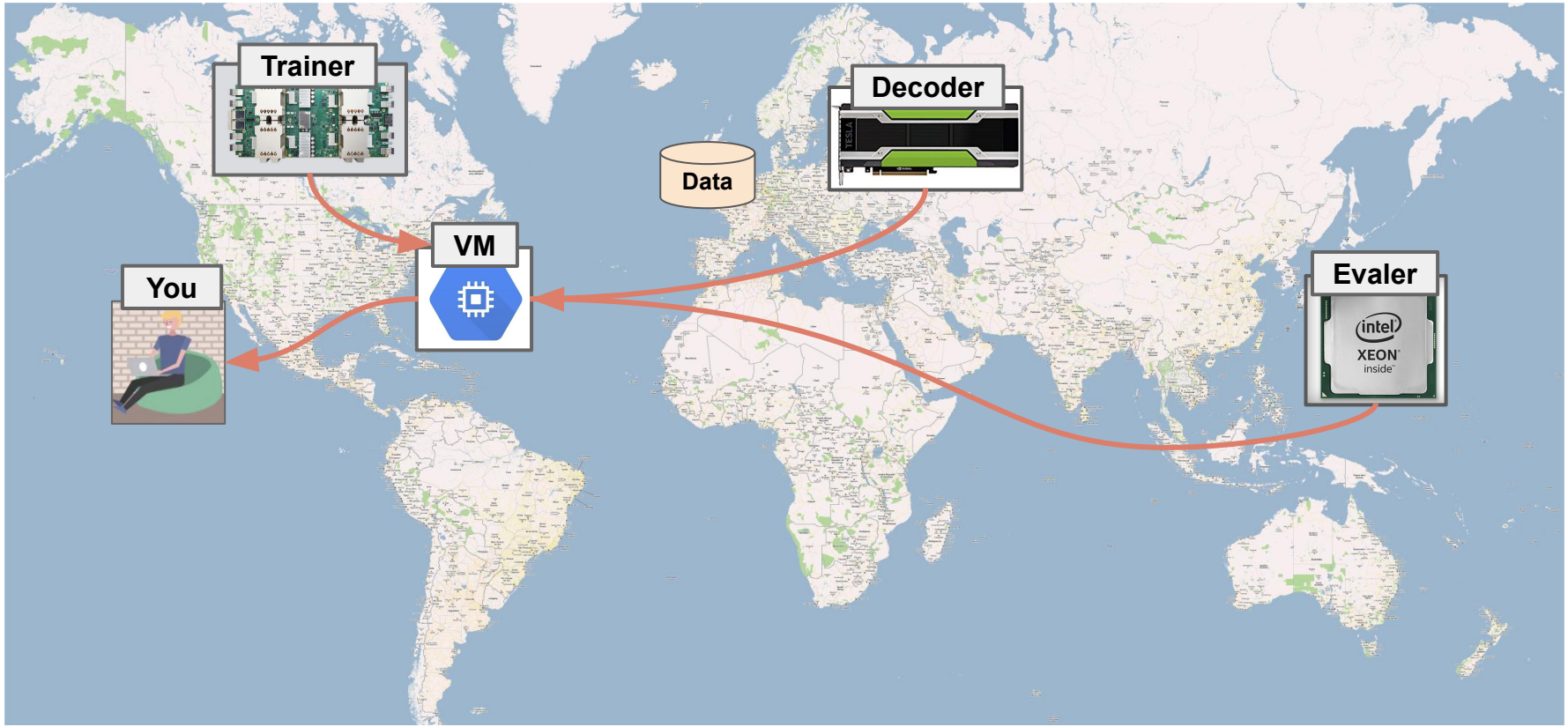
Google AI



*This is just a sketch, exact locations are inaccurate.

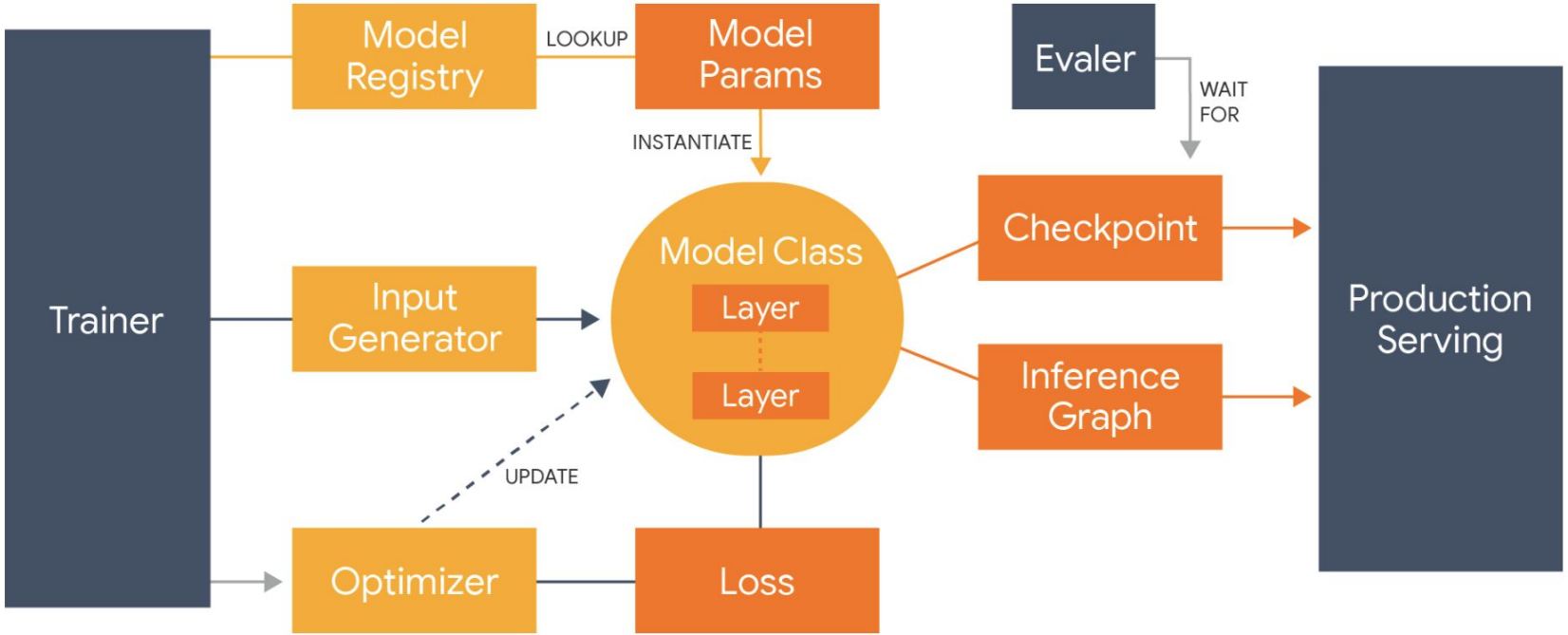
Under the hood

Google AI



*This is just a sketch, exact locations are inaccurate.

Tensorflow Lingvo: github.com/tensorflow/lingvo



Prototyping and debugging large scale neural networks may require **travelling off the beaten path**: devising new plots to monitor.

Debugging Large Scale Models

Sources of “bugs” in large scale machine learning:

1. **Regular bugs:** introduced by ML practitioner
   a. Soln. go grab a coffee

2. **Device/Infra bugs:** hideous bugs
   a. Soln. change device, data, data center

3. **Theoretical bugs:** well... this should’ve never happened in the first place
   a. Soln. brush up your ML.
   b. Look at the right thing, norm of the gradient vs norm of the weights.
   c. Isolate initialization, optimization and malicious data.

Hyper-parameter search may no longer be an option if a single model is taking weeks to train on thousands of TPUs: **meta-learning**.

Hyper-parameter Search

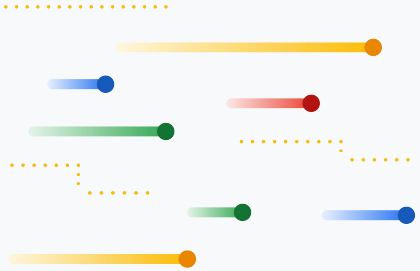
First: No one has infinite resources → the more compute we get, the larger we scale.

Some rule-of-thumbs

- All variables are interconnected: if you are changing one, expect the others to be changed
- Always start with the learning rate, then the batch-size
- Hill-climbing is as good as random search

Some tools to automate

- Vizier for Cloud
- Tune for Pytorch



Google AI

The Learning Rate Schedules

“Often the single most important hyper-parameter”

Practical recommendations for gradient-based training of deep architectures, Bengio 2012

Should always be tuned.

Importance of Configs

For large scale experiments:

- Reproducibility is more important than code reuse, cosmetics and other conventions
- Maintaining sufficient checkpoints
- Having experimental results attached to the configs

DO NOT INHERIT CONFIG CLASSES

```
@model_registry.RegisterSingleTaskModel
class WmtEnDeTransformerBase(base_model_params.SingleTaskModelParams):
  """Params for WMT'14 En->De."""

  DATADIR = '/usr/local/google/wmt14/wpm/'
  VOCAB_SIZE = 32000

  @classmethod
  def Train(cls):
    p = input_generator.NmtInput.Params()
    p.file_pattern = 'tfrecord:'+os.path.join(cls.DATADIR, 'train.tfrecords-*')
    p.tokenizer.token_vocab_filepath = os.path.join(cls.DATADIR,'wpm-ende.voc')
    p.bucket_batch_limit = ([128, 102, 85, 73, 64, 51, 42])
    return p

  @classmethod
  def Dev(cls):
    p = input_generator.NmtInput.Params()
    p.file_pattern = 'tfrecord:' + os.path.join(cls.DATADIR, 'dev.tfrecords')
    p.tokenizer.token_vocab_filepath = os.path.join(cls.DATADIR, 'wpm-ende.voc')
    p.bucket_batch_limit = [16] * 8 + [4] * 2
    return p

  @classmethod
  def Test(cls):
    p = input_generator.NmtInput.Params()
    p.file_pattern = 'tfrecord:' + os.path.join(cls.DATADIR, 'test.tfrecords')
    p.tokenizer.token_vocab_filepath = os.path.join(cls.DATADIR, 'wpm-ende.voc')
    p.bucket_batch_limit = [16] * 8 + [4] * 2
    return p

  @classmethod
  def Task(cls):
    p = base_config.SetupTransformerParams(
        model.TransformerModel.Params(),
        name='wmt14_en_de_transformer_base',
        vocab_size=cls.VOCAB_SIZE,
        model_dim=512,
        hidden_dim=2048,
        num_heads=8,
        num_layers=6,
        residual_dropout_prob=0.1,
        input_dropout_prob=0.1,
        learning_rate=3.0,
        warmup_steps=40000)
    p.eval.samples_per_summary = 7500
    return p
```

https://github.com/tensorflow/lingvo/blob/master/lingvo/tasks/mt/params/wmt14_en_de.py

Bonus

Platform independent frameworks

- TF ⟶ CPU/GPU/TPU/Mobile/Browser, use the right tool for the job

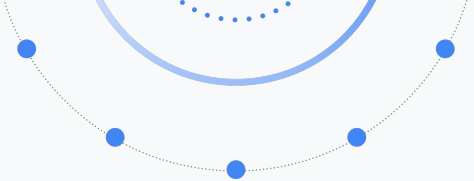
Don’t get lost in the nuances,

- Ask yourself, which research question I’m trying to answer all the time
- There is no end in optimization, so don’t over-optimize things

Working with larger teams

- Async approaches to sync, a simple spreadsheet can help a lot

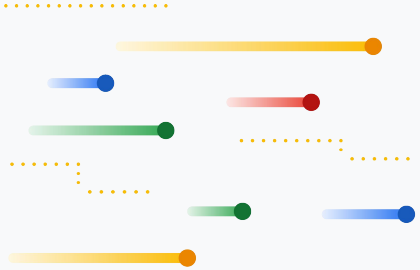
We need something post-silicone, given the current consumption of chips!



Google AI

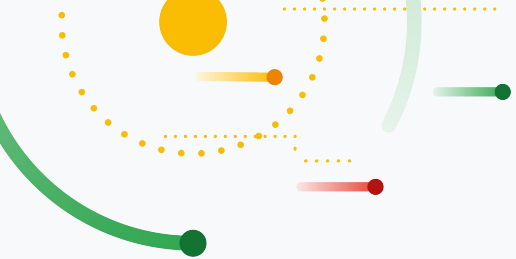
THANKS!

orhanf@google.com



Google AI

Appendix



Google AI

More on Scaling

Hardware, GPipe and Large Batches

GPipe: Easy Scaling with Pipeline Parallelism - Huang et al., 2019

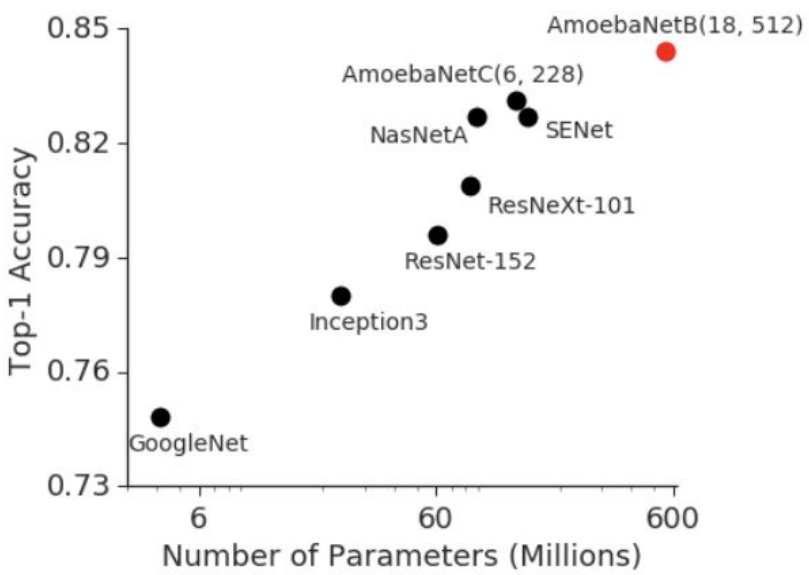
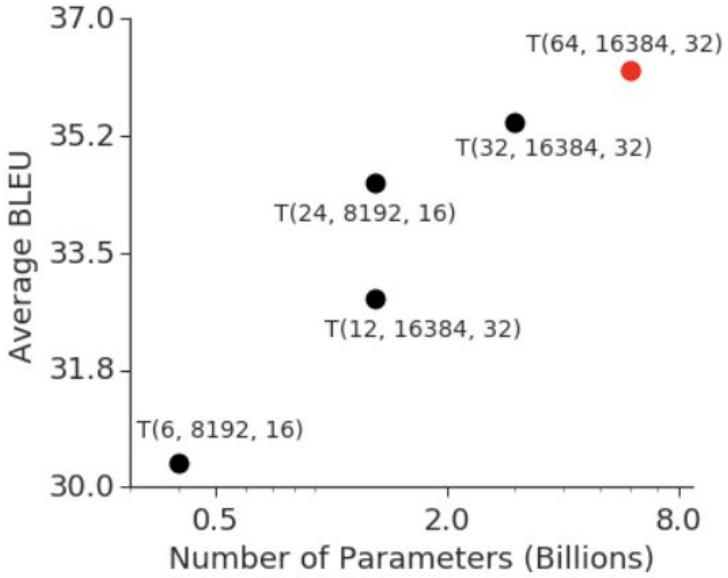


Image-Net



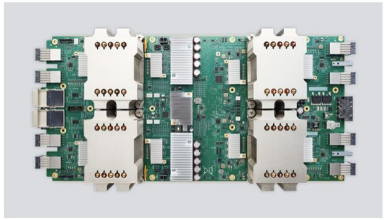
Machine Translation

Compute and Machine Learning Systems

- Training on 1024 TPU-v3 chips
- Bfloat16 (Brain Floating Point)
- GPipe: Micro-Batch Pipeline Parallelism (Huang et al., 2019)
  - Rematerialization (gradient checkpointing)
  - Large batches (4M examples)

TPU v3 Pod in its habitat

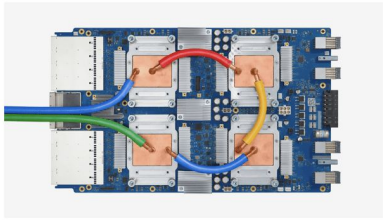
Compute - Tensor Processing Units



Cloud TPU v2

180 teraflops

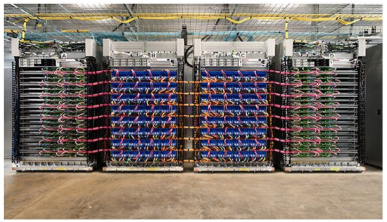
64 GB High Bandwidth Memory (HBM)



Cloud TPU v3

420 teraflops

128 GB HBM

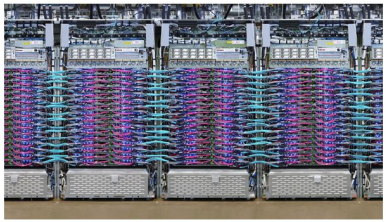


Cloud TPU v2 Pod (beta)

11.5 petaflops

4 TB HBM

2-D toroidal mesh network



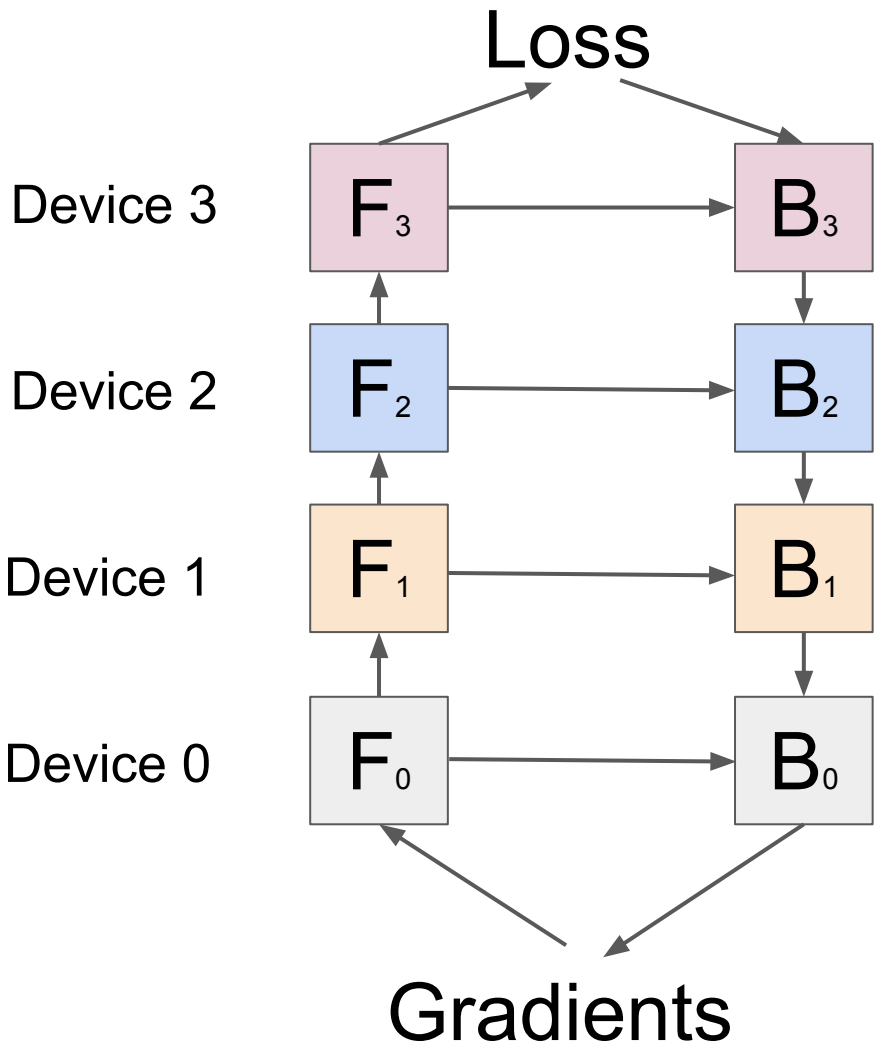
Cloud TPU v3 Pod (beta)

100+ petaflops

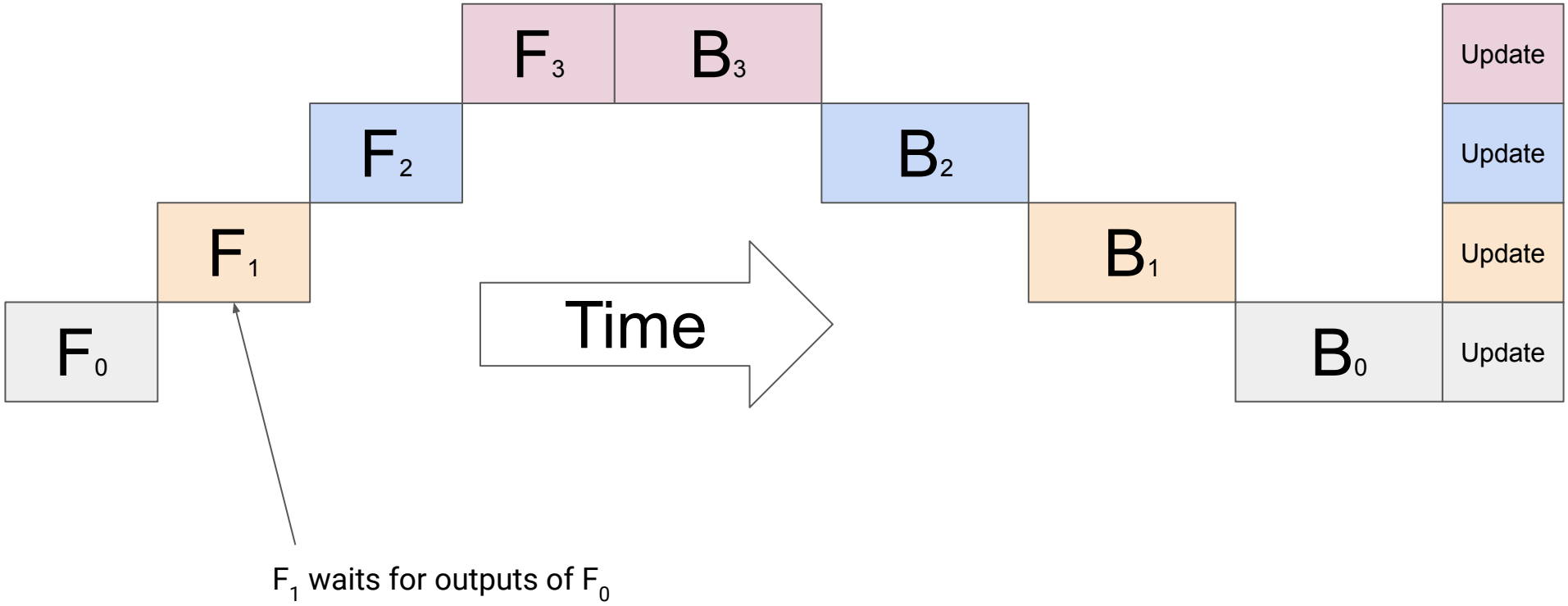
32 TB HBM

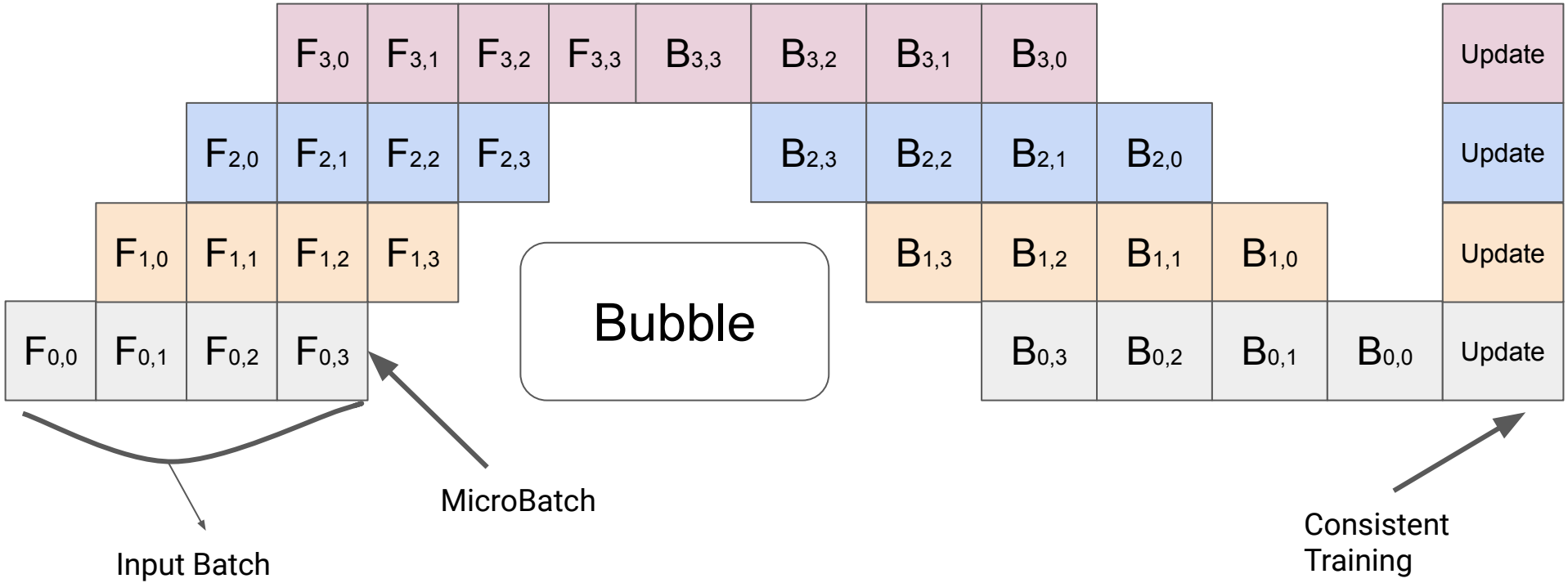
2-D toroidal mesh network

https://cloud.google.com/tpu/


Loss
Device 3
Device 2
Device 1
Device 0
F3
B3
F2
B2
F1
B1
F0
B0
Gradients

Only one accelerator is active when the model is distributed across the accelerators




F3,0
F3,1
F3,2
F3,3
B3,3
B3,2
B3,1
B3,0
Update
F2,0
F2,1
F2,2
F2,3
B2,3
B2,2
B2,1
B2,0
Update
F1,0
F1,1
F1,2
F1,3
B1,3
B1,2
B1,1
B1,0
Update
F0,0
F0,1
F0,2
F0,3
B0,3
B0,2
B0,1
B0,0
Update
Bubble
Input Batch
MicroBatch
Consistent
Training

Performance

Sublinear scaling with or without high speed interconnect.

TPU	AmoebaNet			Transformer		
$K =$	2	4	8	2	4	8
$M = 1$	1	1.13	1.38	1	1.07	1.3
$M = 4$	1.07	1.26	1.72	1.7	3.2	4.8
$M = 32$	1.21	1.84	3.48	1.8	3.4	6.3

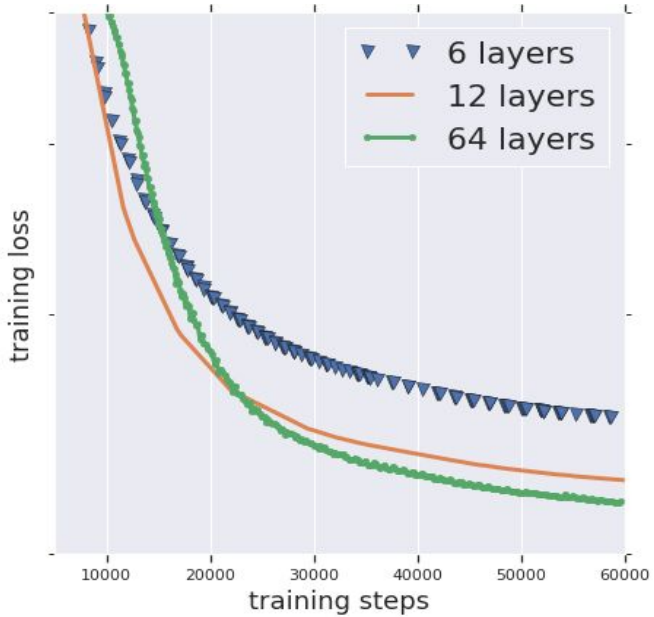
Sub-linear scaling,
6.3x throughput with
8x more chips

K: # of model partitions.
M:# of batch splitting

3/ Increase model capacity:

Validating recent neural-net theory at scale.

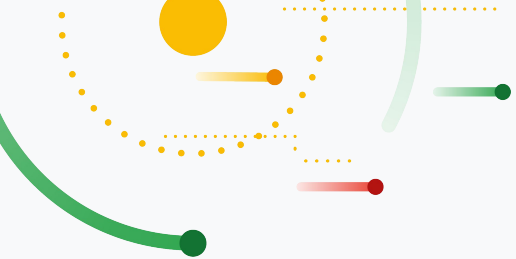
- Implicit acceleration by overparameterization.
- Deeper models are sample efficient.



- Effect of very large batch-sizes.
- Still noise in the 1st order approximation.

Batch Size	260K	1M	4M
BLEU	30.92	31.86	32.71
Loss (NLL)	2.58	2.51	2.46

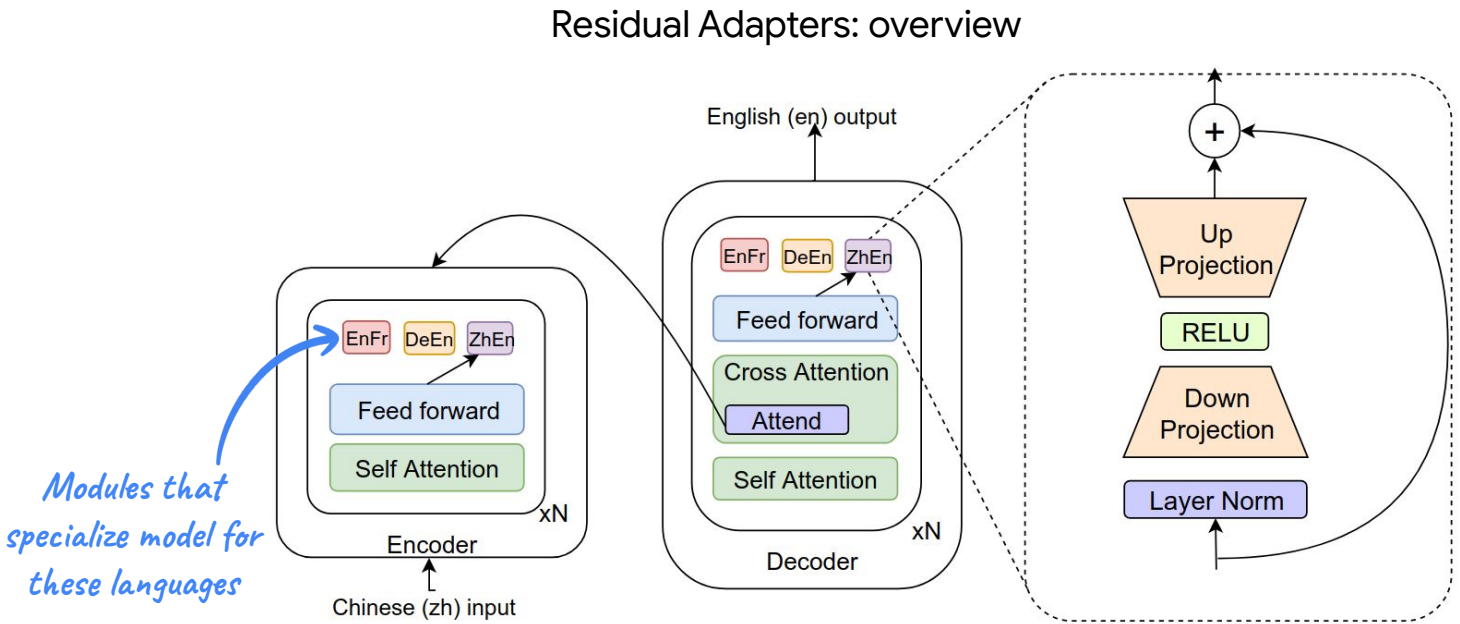
- Way to go for large datasets,
  - Risk of injecting too many noisy examples



Making M4 Practical: Simple Adaptation

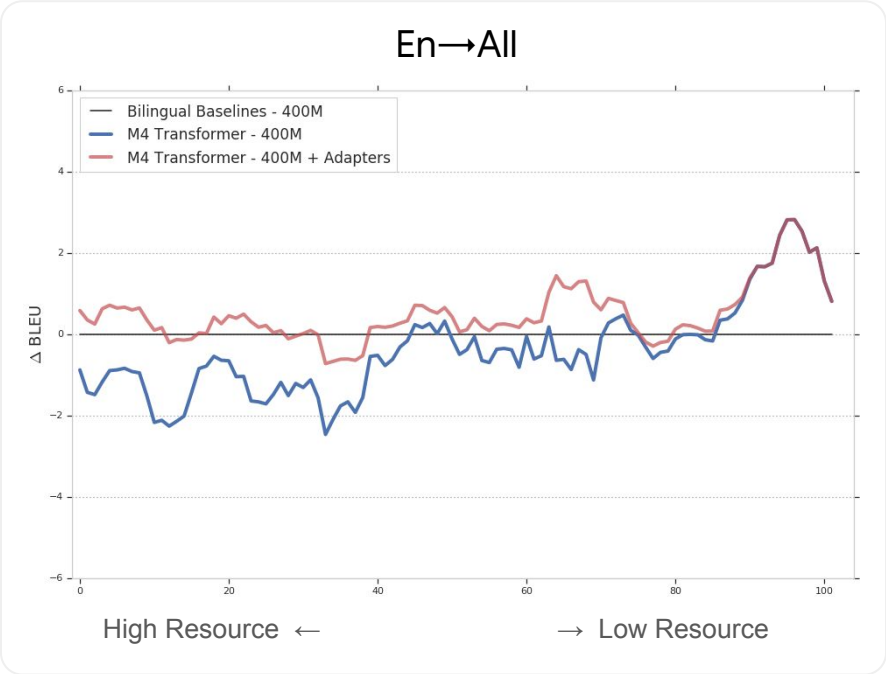
Simple, Scalable Adaptation for Neural Machine Translation, Bapna et al. - EMNLP'19

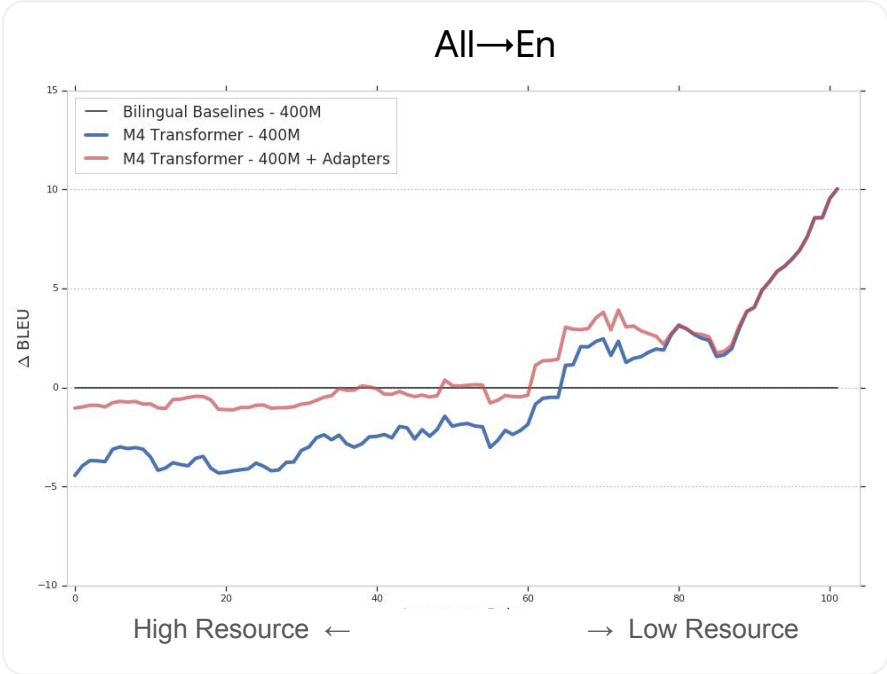
Residual Adapters allow for improving quality and for specializing models on languages or domains



Relevant paper: Simple, Scalable Adaptation for Neural Machine Translation (Bapna et al. - EMNLP'19)

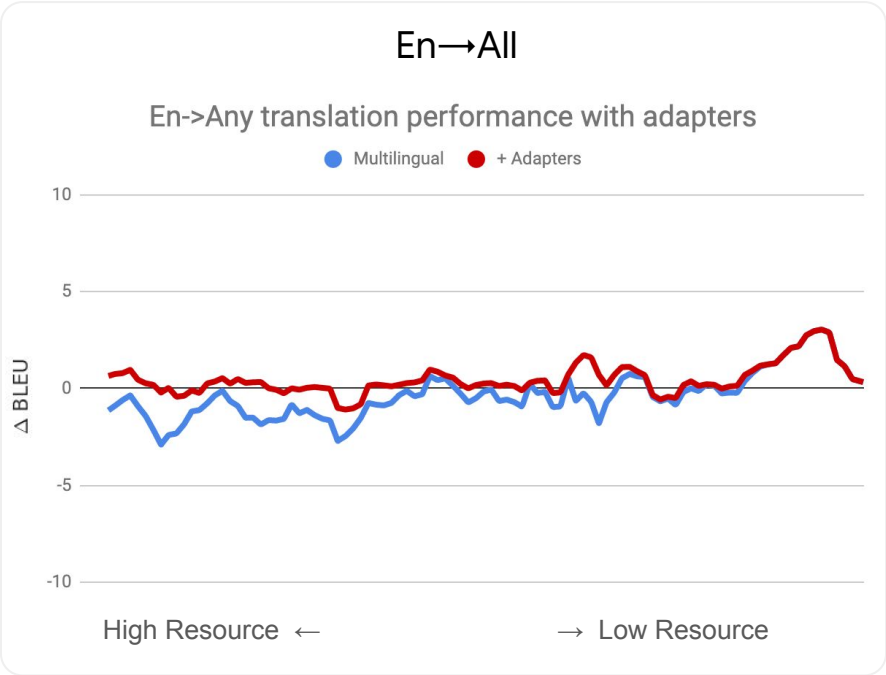
With Residual Adapters, we regain the quality drop in high-to-mid resource languages and enable domain-adaptation

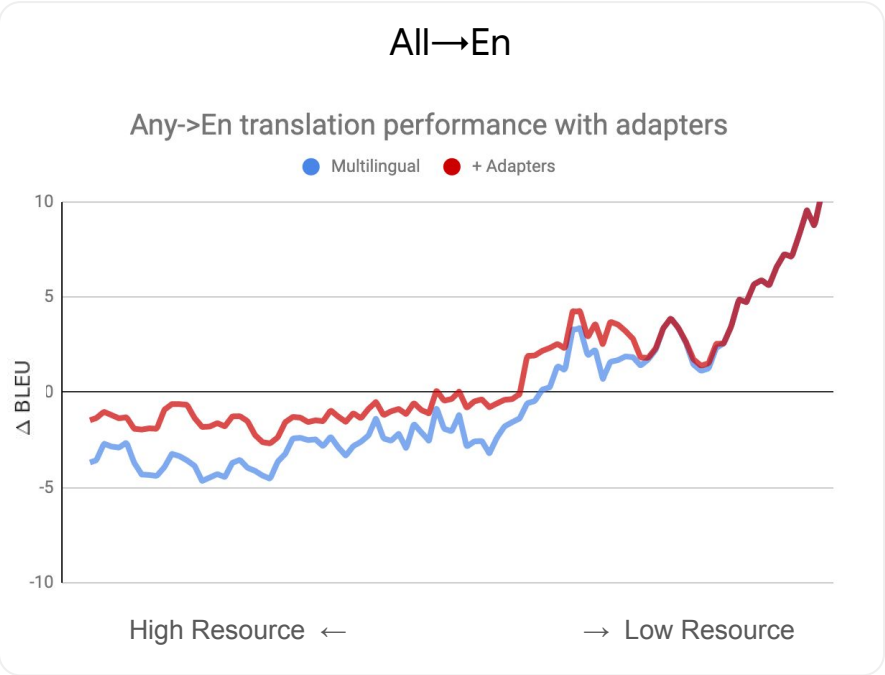




Relevant paper: Simple, Scalable Adaptation for Neural Machine Translation (Bapna et al. - EMNLP'19)

Restore quality on high-to-mid resource languages with 13% extra parameters per language

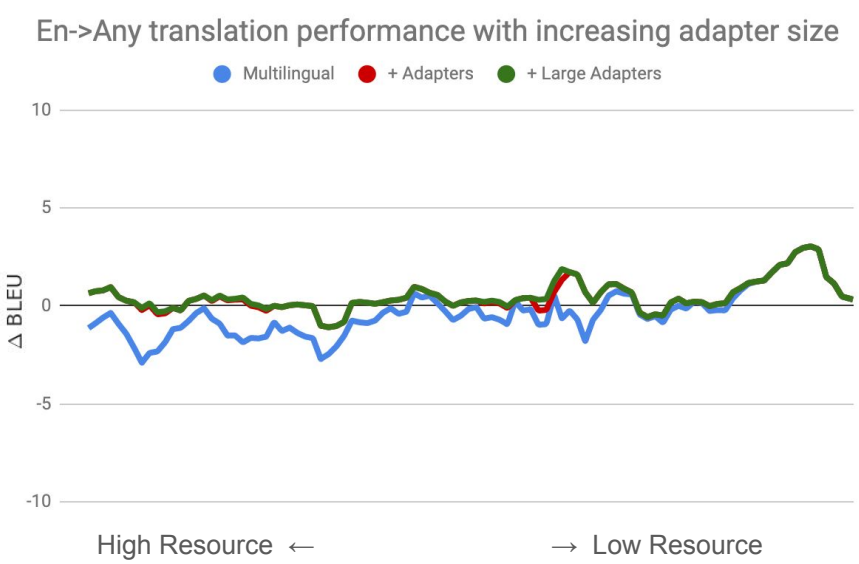




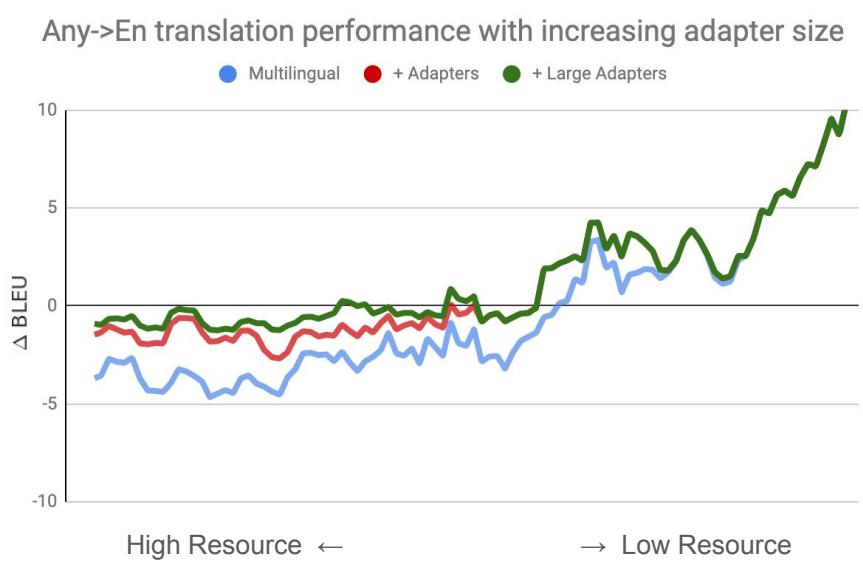
Relevant paper: Simple, Scalable Adaptation for Neural Machine Translation (Bapna et al. - EMNLP'19)

For high resource languages we see further gains by increasing adapter capacity, although the returns are diminishing

En→All



All→En



Relevant paper: Simple, Scalable Adaptation for Neural Machine Translation (Bapna et al. - EMNLP'19)

Enhancing Zero-Shot Translation:
The Missing Ingredient

The missing ingredient in zero-shot Neural Machine Translation, Arivazhagan et al. 2019.

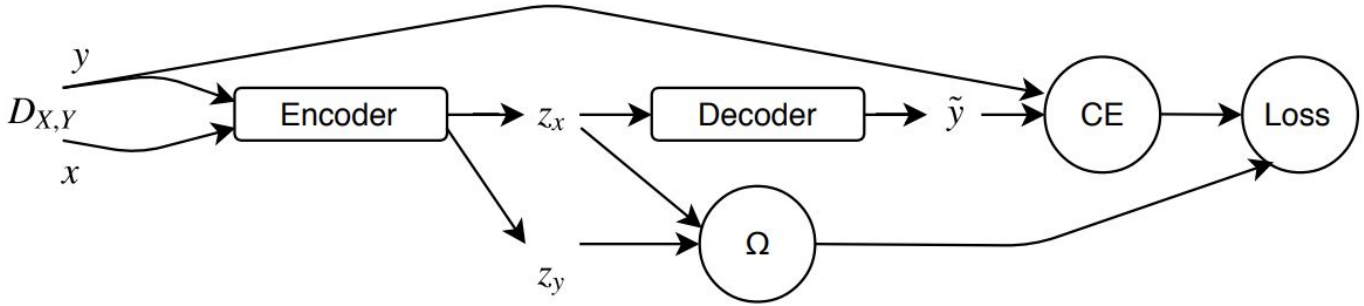
Enhancing Zero-Shot Translation: The Missing Ingredient

Bridging the gap between pivoting and zero-shot

- auxiliary losses on the NMT encoder
- impose representational invariance via loss
- easy scalability to multiple languages

Group	**vanilla**		**align(cosine)**
	direct	pivot	direct
$en \leftrightarrow xx$ (8)	30.11	-	29.95
$xx \leftrightarrow yy$ (12)	16.73 (zs)	17.76	17.72 (zs)
All (20)	22.2	22.81	22.72

Table 5: Average BLEU scores for multilingual model on IWSLT-2017; Zero-Shot results are marked (zs).



[The missing ingredient in zero-shot Neural Machine Translation](), Arivazhagan et al. 2019.