Google Research

Scaling Laws for Multilingual Neural Machine Translation

Patrick Fernandes

Joint Work:

Behrooz Ghorbani, Xavier Garcia, Markus Freitag, Orhan Firat

Accepted at ICML 2023

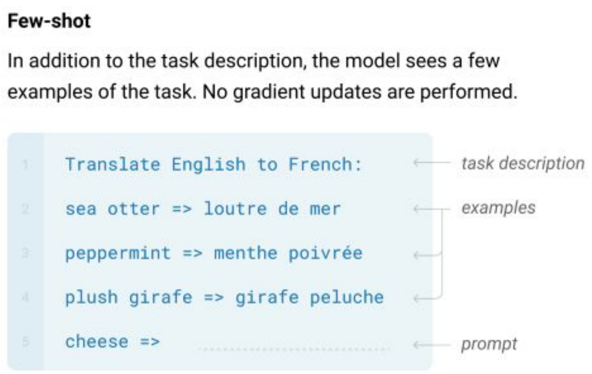
The unreasonable effectiveness of **Scaling**

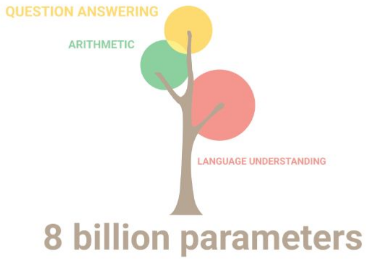
- Scaling up model and data size is an effective way to improve performance of NNs!

The unreasonable effectiveness of **Scaling**

- Scaling up model and data size is an effective way to improve performance of NNs!

- Current *state-of-the-art* models have (many) billions of parameters





Brown et al. (2020) "Language Models are Few-Shot Learners"
Chowdhery et al. (2022) "PaLM: Scaling Language Modeling with Pathways"

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute** (***C***)

- Use **scaling law** to quantify the relationship between these quantities and performance

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

- Use **scaling law** to quantify the relationship between these quantities and performance
  - Assuming *(almost)-infinite* data and compute, the *loss* of a model is given by

$$\mathcal{L}(N) = \beta N^{-\alpha} + L_\infty$$

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

- Use **scaling law** to quantify the relationship between these quantities and performance
  - Assuming *(almost)-infinite* data and compute, the *loss* of a model is given by

$$\mathcal{L}(N) = \underbrace{\beta}_{\text{Model Scaling Multiplier}} N^{-\alpha} + L_\infty$$

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

- Use **scaling law** to quantify the relationship between these quantities and performance
  - Assuming *(almost)-infinite* data and compute, the *loss* of a model is given by

$$\mathcal{L}(N) = \beta N^{-\alpha} + L_\infty$$

Model Scaling Exponent

Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

- Use **scaling law** to quantify the relationship between these quantities and performance
  - Assuming *(almost)-infinite* data and compute, the *loss* of a model is given by

$$\mathcal{L}(N) = \beta N^{-\alpha} + L_\infty$$

<u>Irreducible Loss</u>
(Limitations of Transformers)
(Intrinsic Variance in the Task)

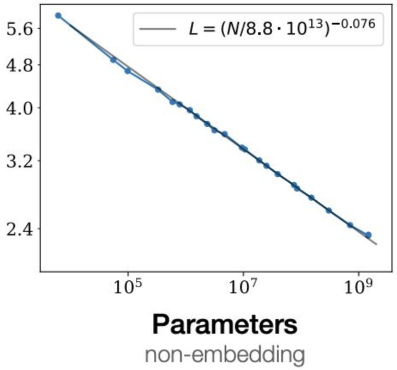
Predicting performance with **Scaling Laws**

- Growing body of literature on predicting the performance of models as we scale
  - In **capacity (*N*)**
  - In **dataset size (*D*)**
  - In **compute (*C*)**

- Use **scaling law** to quantify the relationship between these quantities and performance
  - Assuming *(almost)-infinite* data and compute, the *loss* of a model is given by

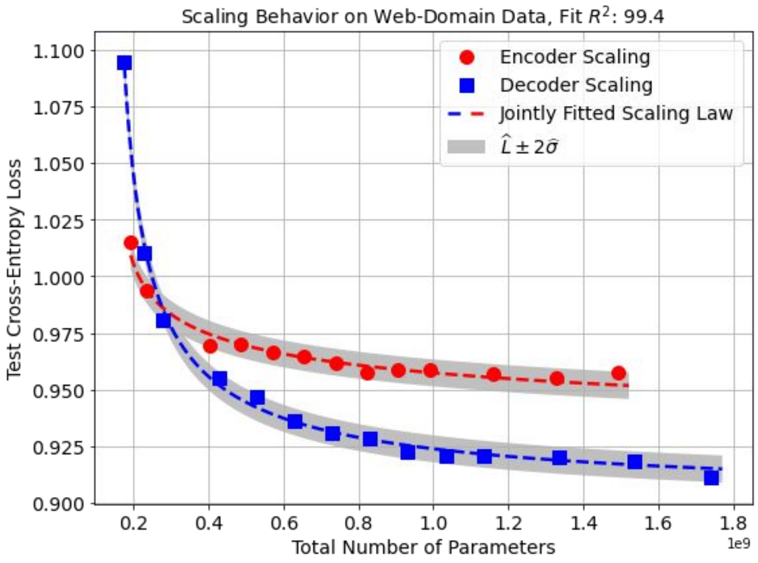
$$\mathcal{L}(N) = \beta N^{-\alpha} + L_\infty$$

- Follows a **power-law** relationship



Scaling Laws for Machine Translation

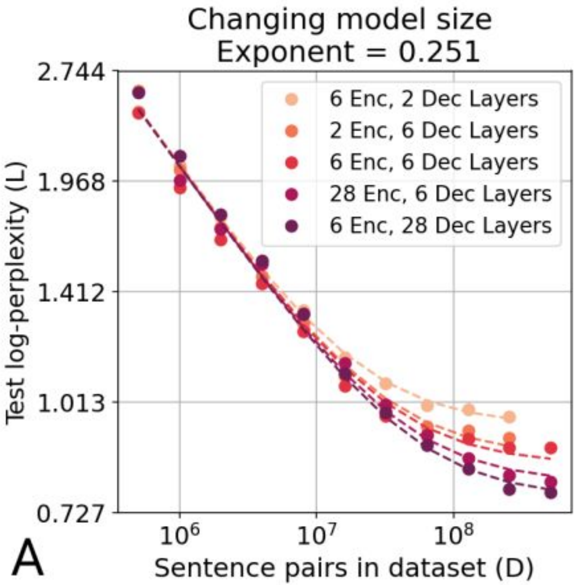
- Performance of Machine Translation models also seems to follow a power-law

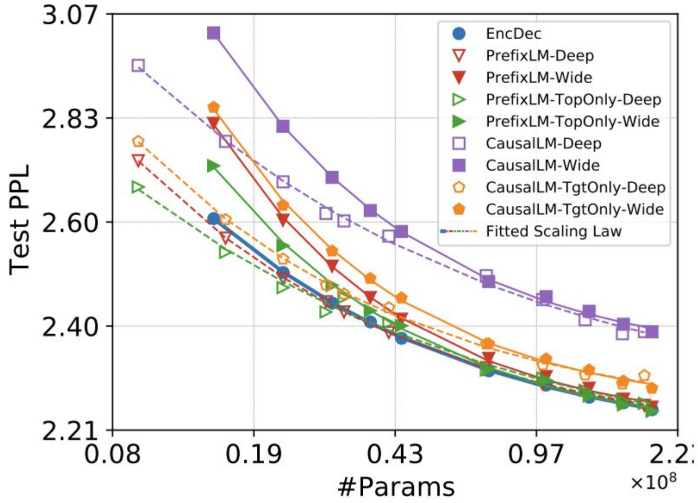


Ghorbani et al (2021). “Scaling Laws for Neural Machine Translation”

Scaling Laws for Machine Translation

- Performance of Machine Translation models also seems to follow a power-law
    - Similar laws for data scaling and different architectures





Bansal et al (2022). “Data Scaling Laws in NMT: The Effect of Noise and Architecture”
Zhang et al (2022). “Examining Scaling and Transfer of Language Model Architectures for Machine Translation”

Multilinguality + Scaling = ❤

- Previous works has mostly consider scaling laws **for a single language/task**

Multilinguality + Scaling = ❤

- Previous works has mostly consider scaling laws **for a single language/task**

- One important capability that highly benefits from scale is **multilinguality**
  - The ability to solve a task in multiple languages

Multilinguality + Scaling = ❤

- Previous works has mostly consider scaling laws **for a single language/task**

- One important capability that highly benefits from scale is **multilinguality**
    - The ability to solve a task in multiple languages

- Massive multilingual models are <u>crucial</u> to break the language barrier in NLP

Multilinguality + Scaling = ❤

- Challenges in developing massive multilingual models that current scaling laws don’t help

Multilinguality + Scaling = ❤

- Challenges in developing massive multilingual models that current scaling laws don’t help
    - How should I **weigh** each language in the training set?

Multilinguality + Scaling = ❤

- Challenges in developing massive multilingual models that current scaling laws don't help
  - How should I **weigh** each language in the training set?
  - Can't easily do hyperparameter optimization with 10's/100's billions of parameters

Multilinguality + Scaling = ❤

- Challenges in developing massive multilingual models that current scaling laws don't help
  - How should I **weigh** each language in the training set?
  - Can't easily do hyperparameter optimization with 10's/100's billions of parameters

*Can we empirically derive scaling laws for multitask/multilingual models the predict their performance for **any** weighting of the languages in the training set?*

Background: Multi-Task Optimization

- Suppose we want to train a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ for K tasks
  - Task *i* with loss $\mathcal{L}_i(\boldsymbol{\theta})$

Background: Multi-Task Optimization

- Suppose we want to train a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ for K tasks
    - Task *i* with loss $\mathcal{L}_i(\boldsymbol{\theta})$
- Multi-task models are often trained by minimizing the **scalarized** loss

$$\hat{\boldsymbol{\theta}}(\boldsymbol{w}) = \arg\min \sum_{i=1}^{K} \boldsymbol{w}_i \mathcal{L}_i(\boldsymbol{\theta}) \quad \text{where} \quad \boldsymbol{w} > 0, \quad \sum_{i=0}^{K} \boldsymbol{w}_i = 1$$

    - **w** is a fixed vector of task weights

Background: Multi-Task Optimization

- Suppose we want to train a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ for K tasks
  - Task *i* with loss $\mathcal{L}_i(\boldsymbol{\theta})$
- Multi-task models are often trained by minimizing the **scalarized** loss

$$\hat{\boldsymbol{\theta}}(\boldsymbol{w}) = \arg\min \sum_{i=1}^{K} \boldsymbol{w}_i \mathcal{L}_i(\boldsymbol{\theta}) \quad \text{where} \quad \boldsymbol{w} > 0, \quad \sum_{i=0}^{K} \boldsymbol{w}_i = 1$$

  - **w** is a fixed vector of task weights
- **Scalarization** perform on par/better than more complex multi-task optimizers

Xin et al (2022). "Do Current Multi-Task Optimization Methods in Deep Learning Even Help?

Background: Multi-Task Optimization

- Suppose we want to train a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ for K tasks
    - Task *i* with loss $\mathcal{L}_i(\boldsymbol{\theta})$
- Multi-task models are often trained by minimizing the **scalarized** loss

$$\hat{\boldsymbol{\theta}}(\boldsymbol{w}) = \arg\min \sum_{i=1}^{K} \boldsymbol{w}_i \mathcal{L}_i(\boldsymbol{\theta}) \quad \text{where} \quad \boldsymbol{w} > 0, \quad \sum_{i=0}^{K} \boldsymbol{w}_i = 1$$

    - **w** is a fixed vector of task weights
- **Scalarization** perform on par/better than more complex multi-task optimizers
- Typically implemented *implicitly*
    - Sample observations from each task according to its weight on the loss

Xin et al (2022). "Do Current Multi-Task Optimization Methods in Deep Learning Even Help?

Data-Rich Multi-Task Optimization

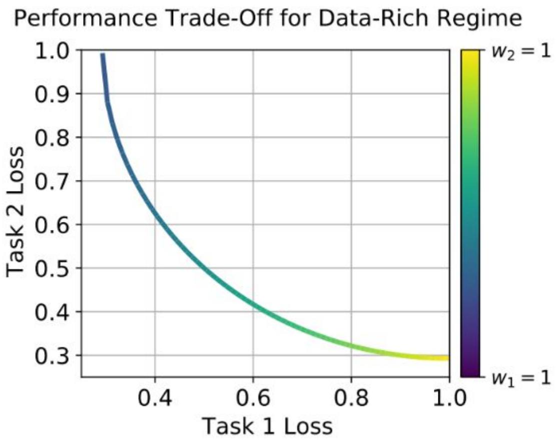
- In the presence of sufficient data for each, there is a *performance trade-off frontier*

Xin et al (2022). “Do Current Multi-Task Optimization Methods in Deep Learning Even Help?

Google

Data-Rich Multi-Task Optimization

- In the presence of sufficient data for each, there is a *performance trade-off frontier*

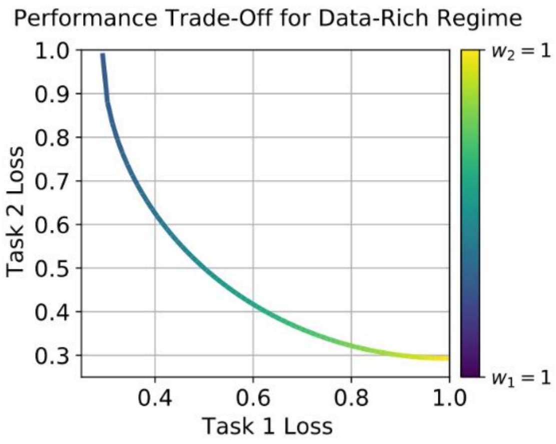


Xin et al (2022). "Do Current Multi-Task Optimization Methods in Deep Learning Even Help?

Google

Data-Rich Multi-Task Optimization

- In the presence of sufficient data for each, there is a *performance trade-off frontier*



*Can we empirically derive scaling laws for multilingual models in the **data-rich** scenario?*

Xin et al (2022). "Do Current Multi-Task Optimization Methods in Deep Learning Even Help?

Setting: Interference

- We attempt to first fully understand **data-rich** scenario

Google

Setting: Interference

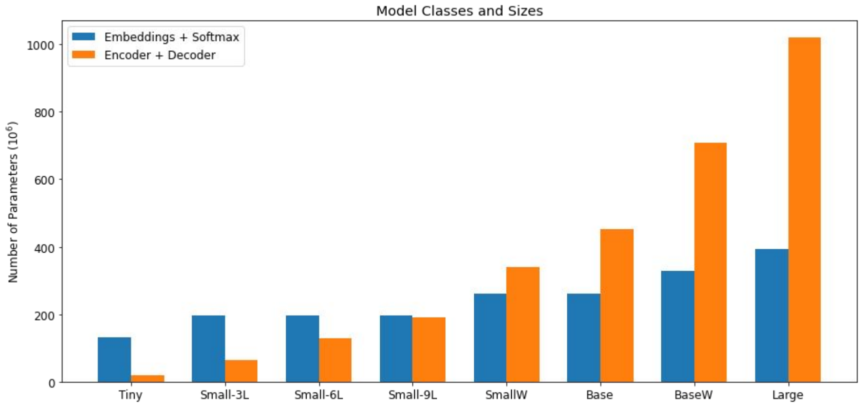
- We attempt to first fully understand **data-rich** scenario
- We train models for three language-pair combinations
    - English→German+Chinese, English→German+French and German+Chinese→English
    - 600M sentences of production data for each language pair (1.2B for each model)

Setting: Interference

- We attempt to first fully understand **data-rich** scenario
- We train models for three language-pair combinations
  - English→German+Chinese, English→German+French and German+Chinese→English
  - 600M sentences of production data for each language pair (1.2B for each model)
- We train up to 8 model sizes (from 10M to 1B non-embedding parameters)

Setting: Interference

- We attempt to first fully understand **data-rich** scenario
- We train models for three language-pair combinations
  - English→German+Chinese, English→German+French and German+Chinese→English
  - 600M sentences of production data for each language pair (1.2B for each model)
- We train up to 8 model sizes (from 10M to 1B non-embedding parameters)



Setting: Interference

- We attempt to first fully understand **data-rich** scenario
- We train models for three language-pair combinations
  - English→German+Chinese, English→German+French and German+Chinese→English
  - 600M sentences of production data for each language pair (1.2B for each model)
- We train up to 8 model sizes (from 10M to 1B non-embedding parameters)
- We vary the task **weight/probability** for each language (different mixture probabilities)

$$p_1 \in [0,\ 0.01,\ 0.05,\ 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 0.95,\ 0.99,\ 1] \qquad p_2 = 1 - p_1$$

Setting: Interference

- We attempt to first fully understand **data-rich** scenario
- We train models for three language-pair combinations
  - English→German+Chinese, English→German+French and German+Chinese→English
  - 600M sentences of production data for each language pair (1.2B for each model)
- We train up to 8 model sizes (from 10M to 1B non-embedding parameters)
- We vary the task **weight/probability** for each language (different mixture probabilities)

$$p_1 \in [0,\ 0.01,\ 0.05,\ 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 0.95,\ 0.99,\ 1] \qquad p_2 = 1 - p_1$$

- We evaluate on **in-domain** and **out-of-domain** test sets

Results: English⟶German+Chinese

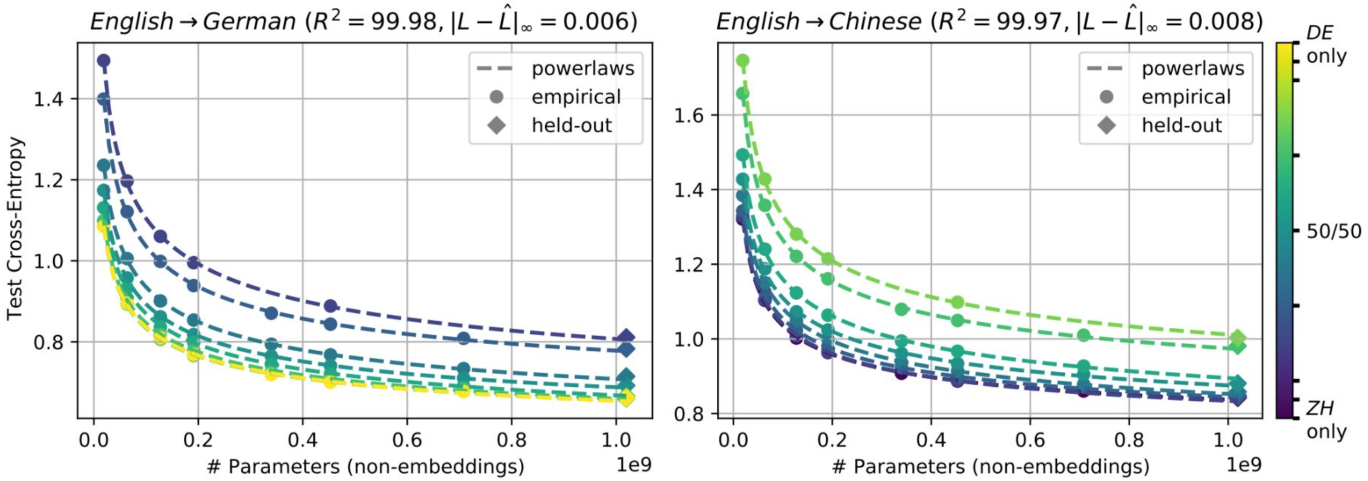
- We then fit individual scaling laws for each task weighting **for both languages**

$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_{p,i}} + L_\infty^{(p,i)}$$

Results: English→German+Chinese

- We then fit individual scaling laws for each task weighting **for both languages**

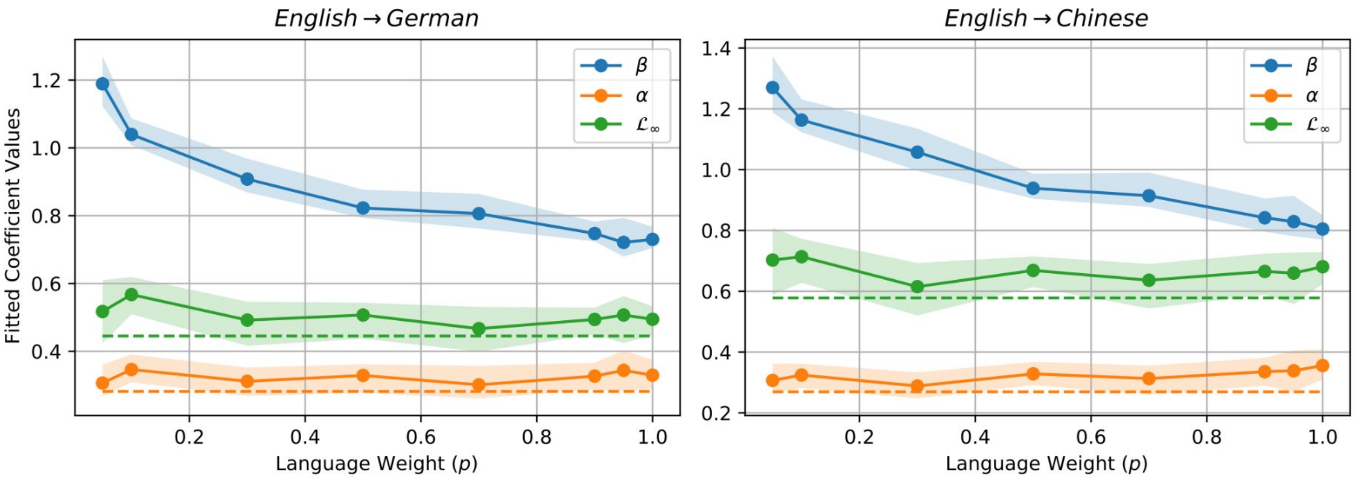
$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_{p,i}} + L_\infty^{(p,i)}$$



Results: English⟶German+Chinese

- The scalings laws seem to possess certain **invariances**

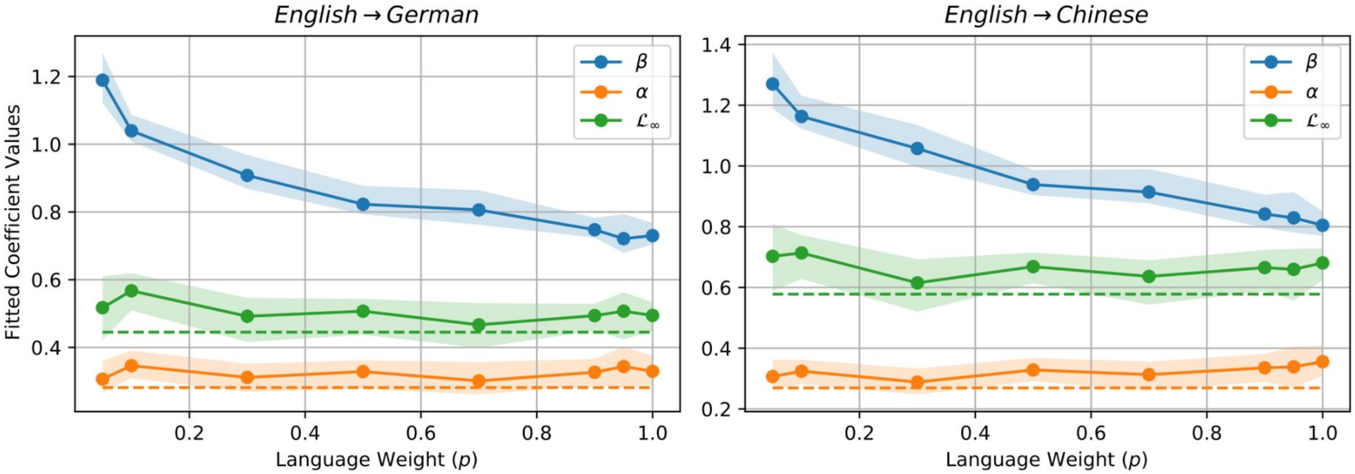
$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_{p,i}} + L_\infty^{(p,i)}$$



Results: English→German+Chinese

- The scalings laws seem to possess certain **invariances**

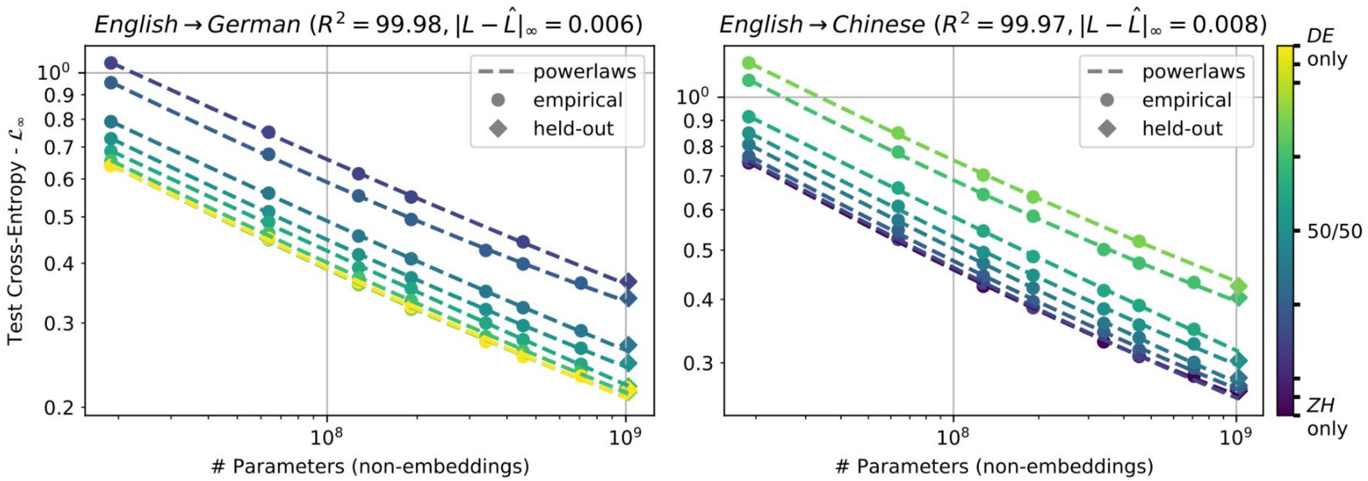
$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_{p,i}} + L_{\infty}^{(p,i)}$$



- Scaling exponent and irreducible loss seem to be (~) constant across mixture probabilities!

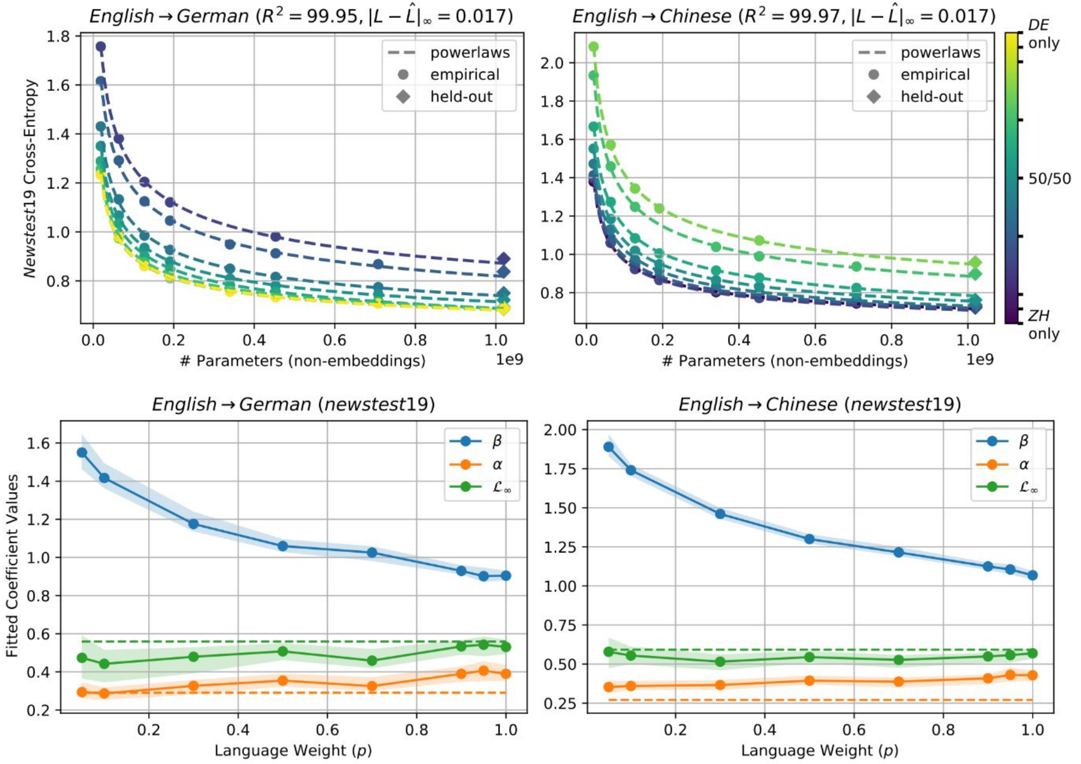
Results: English⟶German+Chinese

- When we subtract a constant for irreducible loss, we plot in log-log axes



Results: Out-of-Domain

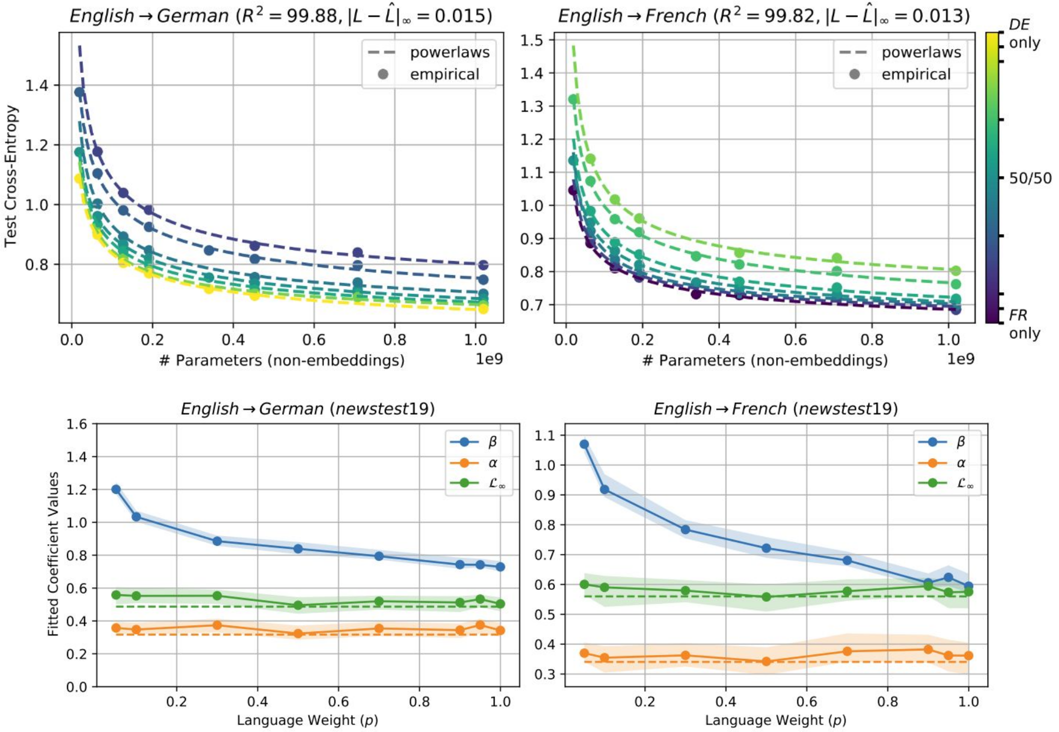
- These findings hold for different domains



Google

Results: Other Language-Pair Combinations

- These findings hold for different language-pair combinations



Google

Jointly Modeling Multitask Scaling

- Based on these findings, we make the assumption that

The scaling exponents and irreducible loss are independent of task weight

Jointly Modeling Multitask Scaling

- Based on these findings, we make the assumption that

The scaling exponents and irreducible loss are independent of task weight

- This means we can derive a joint scaling law for all task weights

$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_i} + L_{\infty}^{(i)}$$

Jointly Modeling Multitask Scaling

- Based on these findings, we make the assumption that

The scaling exponents and irreducible loss are independent of task weight

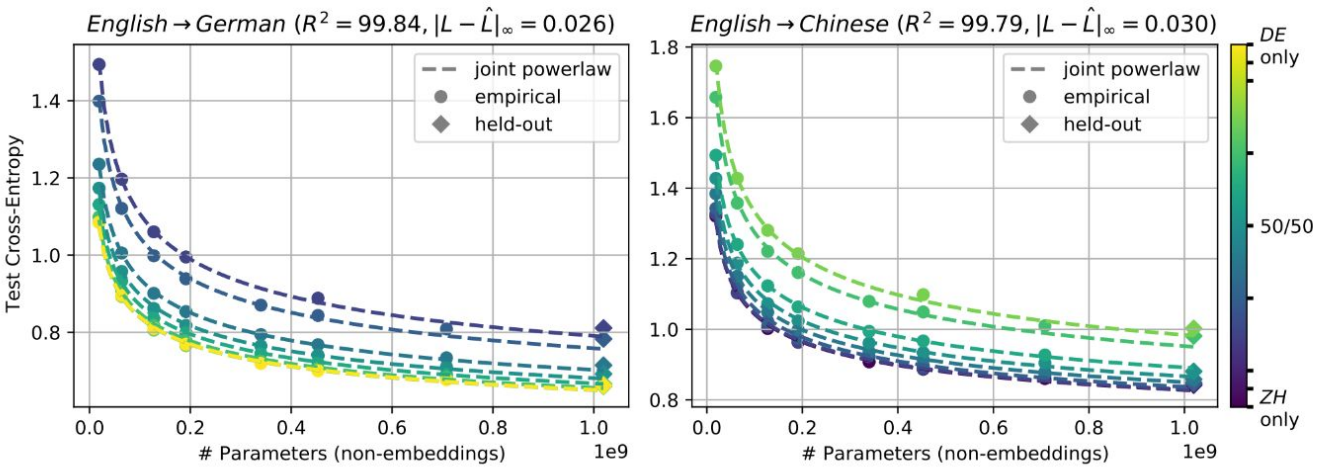
- This means we can derive a <u>joint</u> scaling law for all task weights

$$\mathcal{L}_i(N;p) = \beta_{p,i} N^{-\alpha_i} + L_\infty^{(i)}$$

  - Same exponent and irreducible loss for all task weights, different multipliers for each
  - ~1 coefficient per mixture weighting!

Jointly Modeling Multitask Scaling: En⟶De+Zh

- A joint scaling law provides a good fit for most task weightings!



Effective Network Capacity for Multitask Models

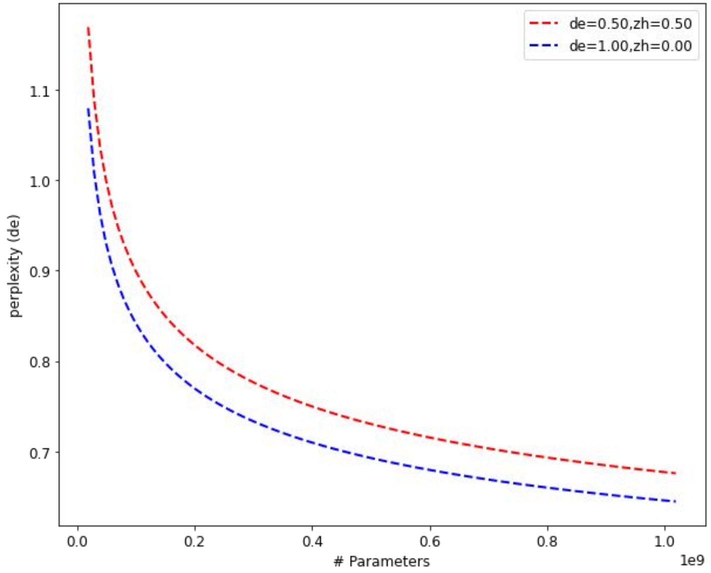
- We joint scaling exponents and loss, we can ask:

“How large should a model trained on both German+French be to match one trained only on German?”

Effective Network Capacity for Multitask Models

- We joint scaling exponents and loss, we can ask:

“How large should a model trained on both German+French be to match one trained only on German?”

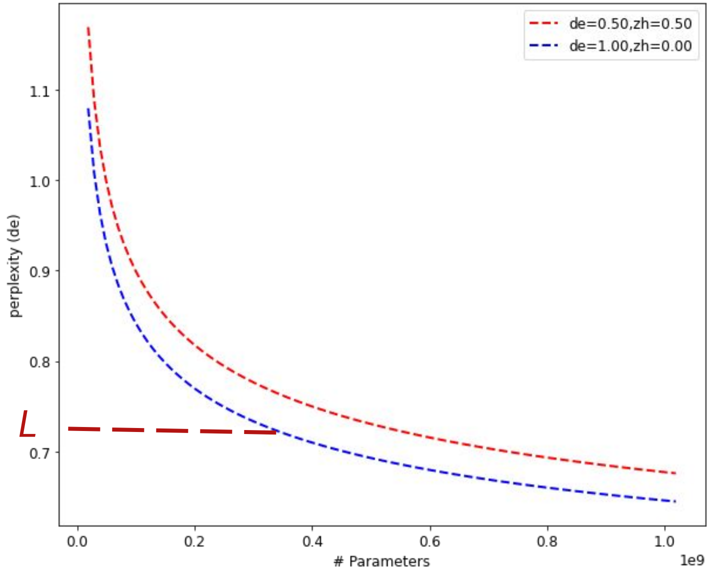


Google

Effective Network Capacity for Multitask Models

- We joint scaling exponents and loss, we can ask:

"How large should a model trained on both German+French be to match one trained only on German?"

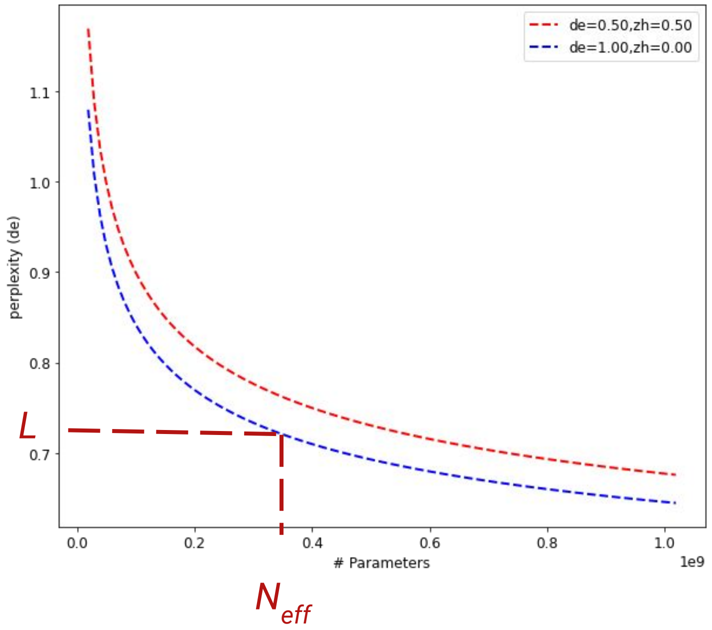


Effective Network Capacity for Multitask Models

- We joint scaling exponents and loss, we can ask:

"How large should a model trained on both German+French be to match one trained only on German?"

$$\mathcal{L}_i(N_{eff}^{(i,p)}; 1)$$

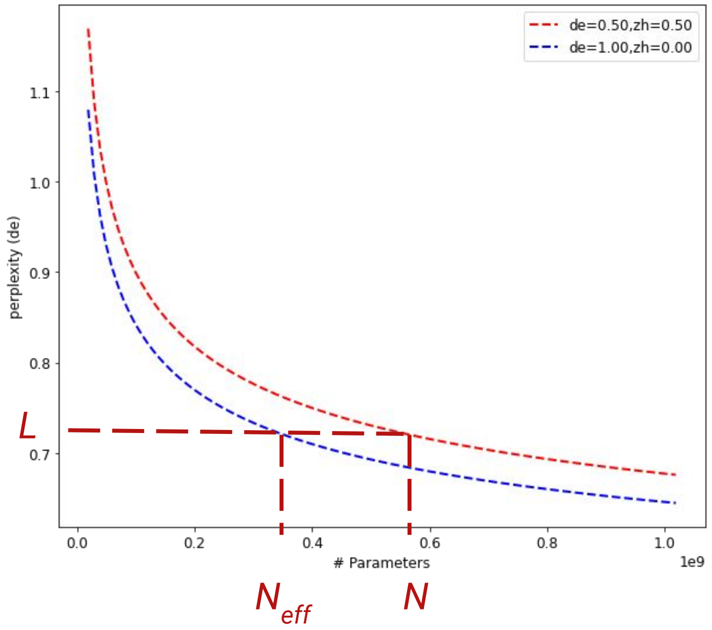


Effective Network Capacity for Multitask Models

- We joint scaling exponents and loss, we can ask:

"How large should a model trained on both German+French be to match one trained only on German?"

$$\mathcal{L}_i(N;p) = \mathcal{L}_i(N_{eff}^{(i,p)};1)$$



Effective Network Capacity for Multitask Models

- We joint scaling exponents and loss, we can ask:

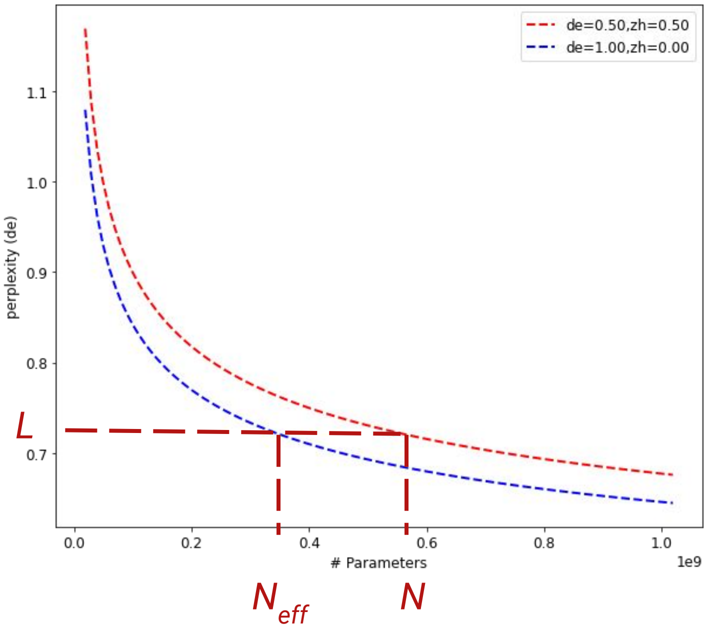
“How large should a model trained on both German+French be to match one trained only on German?”

$$\mathcal{L}_i(N;p) = \mathcal{L}_i(N_{eff}^{(i,p)};1)$$

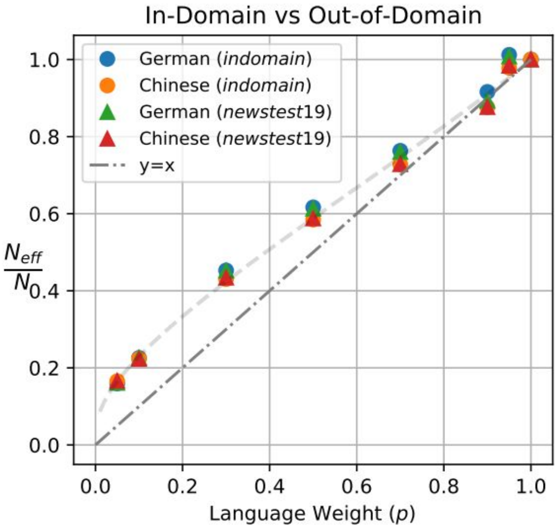
- We can empirically compute this

  effective parameters number

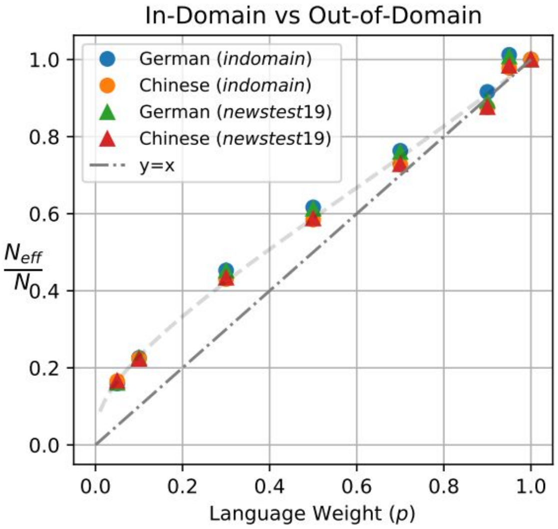
$$N_{eff}^{(i,p)} = \left(\frac{\beta_{1,i}}{\beta_{p,i}}\right)^{\frac{1}{\alpha_i}} N$$



Effective Network Capacity for Multitask Models

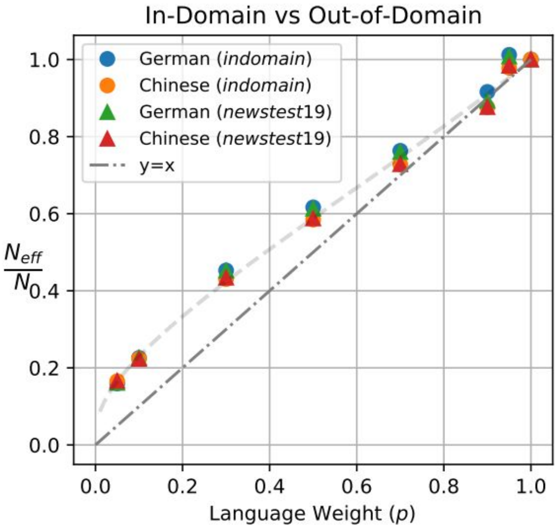


Effective Network Capacity for Multitask Models



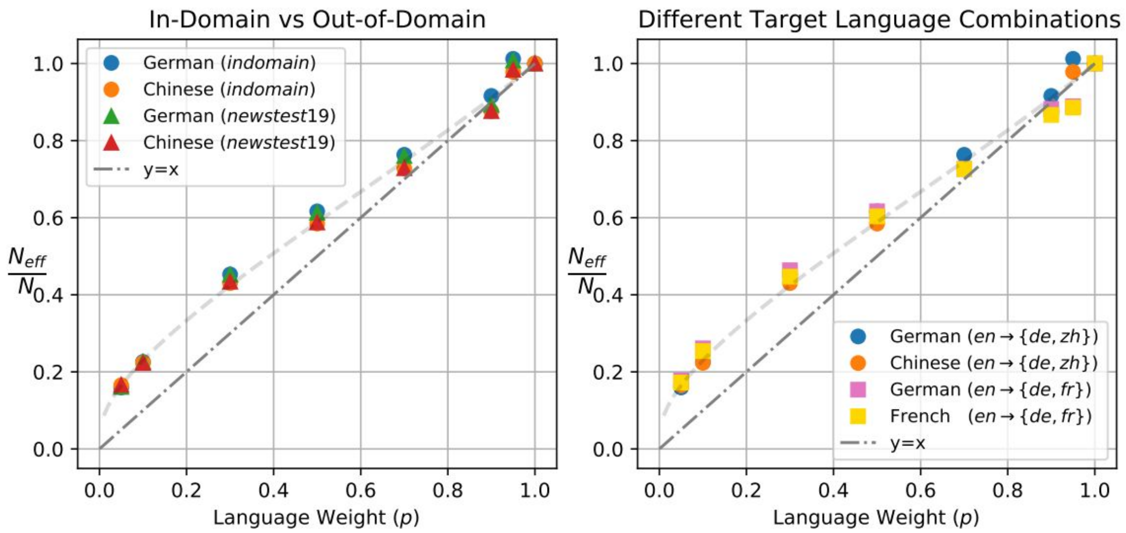
- Effective parameter reduction is almost linear on task probability!
  - Model with 50% german is close to a model 50% parameters on only german!

Effective Network Capacity for Multitask Models



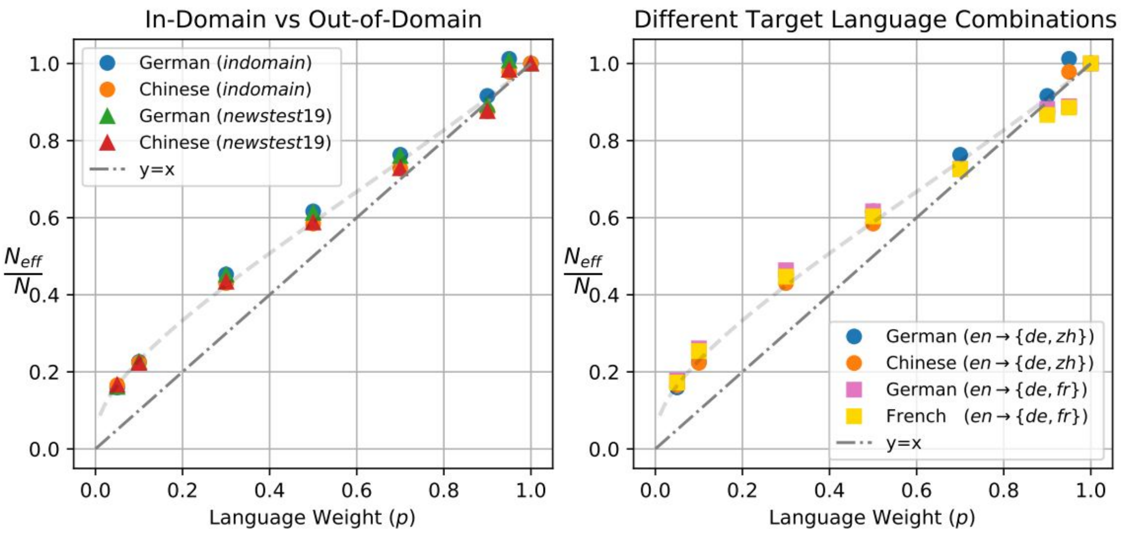
- Effective parameter reduction is almost linear on task probability!
    - Model with 50% german is close to a model 50% parameters on only german!
- Capacity splitting is similar for in-domain and out-of-domain

Effective Network Capacity for Multitask Models



Google

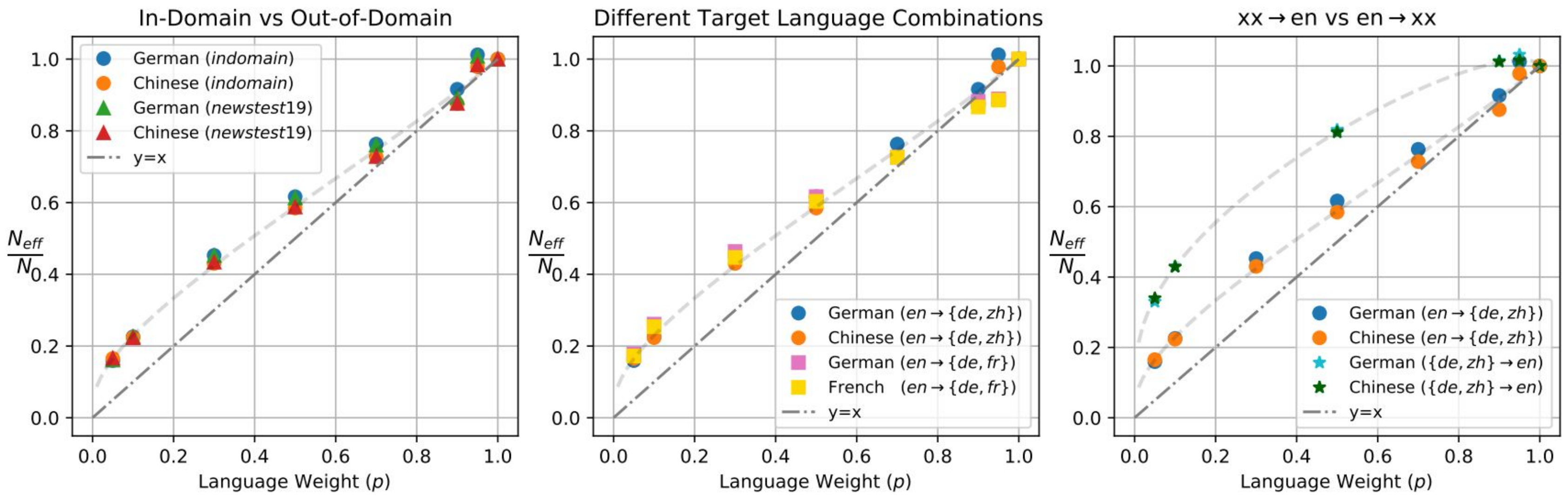
Effective Network Capacity for Multitask Models



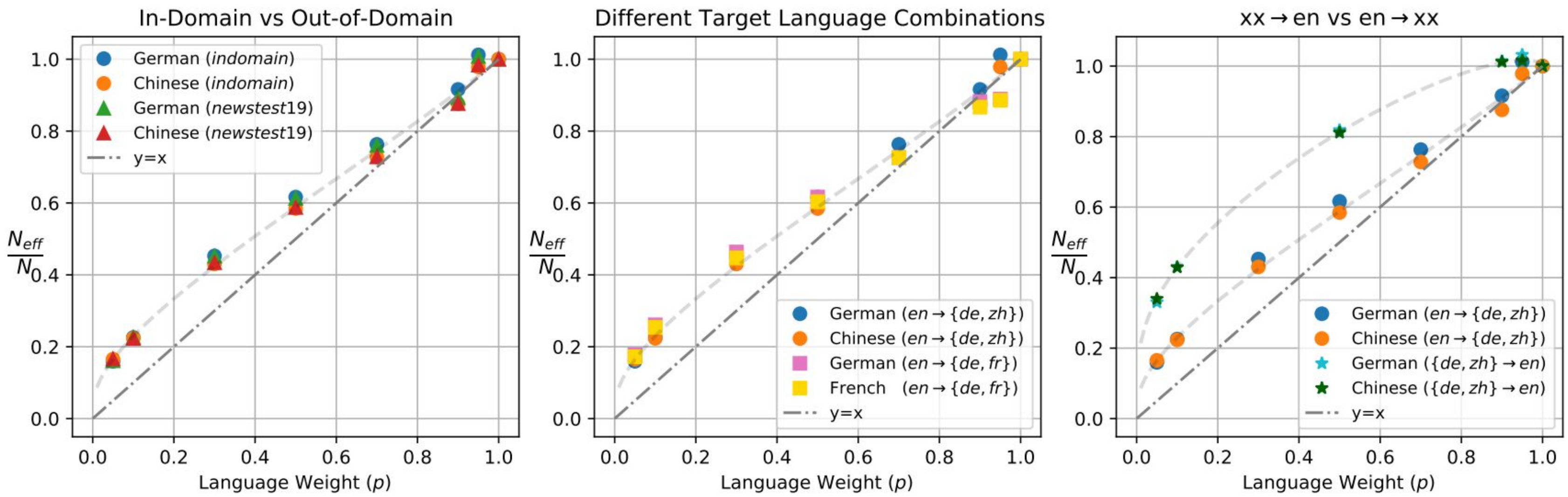
- Despite being more similar, capacity for English→German+French are very similar
  - Very little “sharing” of parameters between languages

Google

Effective Network Capacity for Multitask Models



Effective Network Capacity for Multitask Models



- In contrast, *"direction"* plays an important role in effective capacity
  - Positive synergy between tasks and higher "parameter sharing"

Guiding Task Balancing

- Can we leverage our multilingual/multitask scaling law to guide task balancing in large models?

Guiding Task Balancing

- Can we leverage our multilingual/multitask scaling law to guide task balancing in large models?

$$\mathcal{L}_i(N;p) = \mathcal{L}_i(f_i(p) \times N; 1) \qquad f_i(p) \equiv \frac{N_{\text{eff}}^{(i,p)}}{N} = \left(\frac{\beta_{1,i}}{\beta_{p,i}}\right)^{\frac{1}{\alpha_i}}$$

Guiding Task Balancing

- Can we leverage our multilingual/multitask scaling law to guide task balancing in large models?

$$\mathcal{L}_i(N;p) = \mathcal{L}_i(f_i(p) \times N; 1) \qquad f_i(p) \equiv \frac{N_{\text{eff}}^{(i,p)}}{N} = \boxed{\left(\frac{\beta_{1,i}}{\beta_{p,i}}\right)}^{\frac{1}{\alpha_i}}$$

- In its current form can only predict performance for weighting used to fit the law

Guiding Task Balancing

- Can we leverage our multilingual/multitask scaling law to guide task balancing in large models?

$$\mathcal{L}_i(N; p) = \mathcal{L}_i(f_i(p) \times N; 1) \qquad f_i(p) \equiv \frac{N_{\text{eff}}^{(i,p)}}{N} = \left(\frac{\beta_{1,i}}{\beta_{p,i}}\right)^{\frac{1}{\alpha_i}}$$

- In its current form can only predict performance for weighting used to fit the law

- To extend to unseen task weightings, we instead focus on estimating $f(p)$

$$\hat{f}_i(p) = p + c_1 p^{c_2} (1-p)^{c_3}$$

  - c^1, c^2 and c^3 are coefficient fitted with joint scaling law

Guiding Task Balancing

- Can we leverage our multilingual/multitask scaling law to guide task balancing in large models?

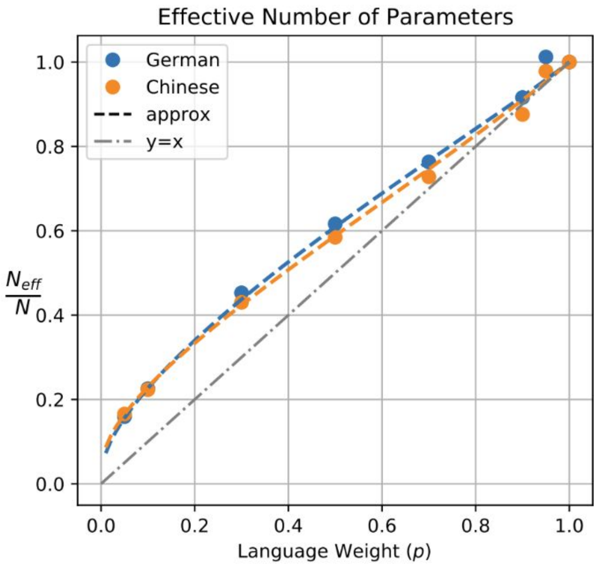
$$\mathcal{L}_i(N;p) = \mathcal{L}_i(f_i(p) \times N; 1) \qquad f_i(p) \equiv \frac{N_{\text{eff}}^{(i,p)}}{N} = \left(\frac{\beta_{1,i}}{\beta_{p,i}}\right)^{\frac{1}{\alpha_i}}$$

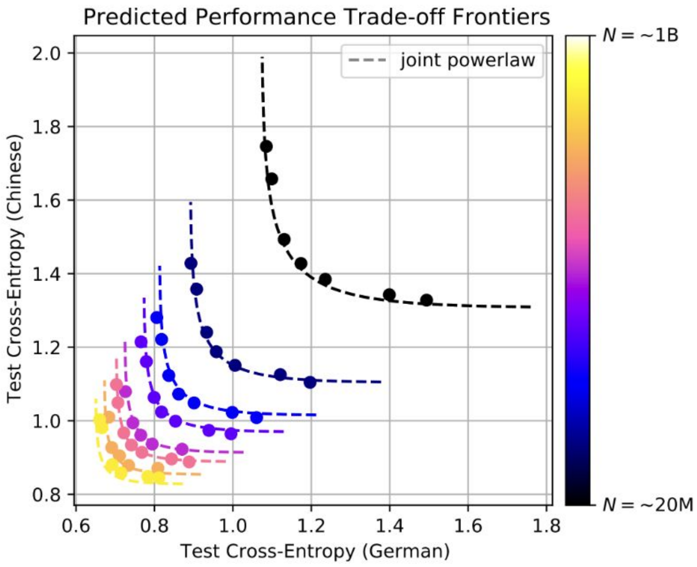
- In its current form can only predict performance for weighting used to fit the law
- To extend to unseen task weightings, we instead focus on estimating $f(p)$

$$\hat{f}_i(p) = p + c_1 p^{c_2} (1-p)^{c_3}$$

  - c^1, c^2 and c^3 are coefficient fitted with joint scaling law
- With this parameterization, we can predict performance **for any task weighting**

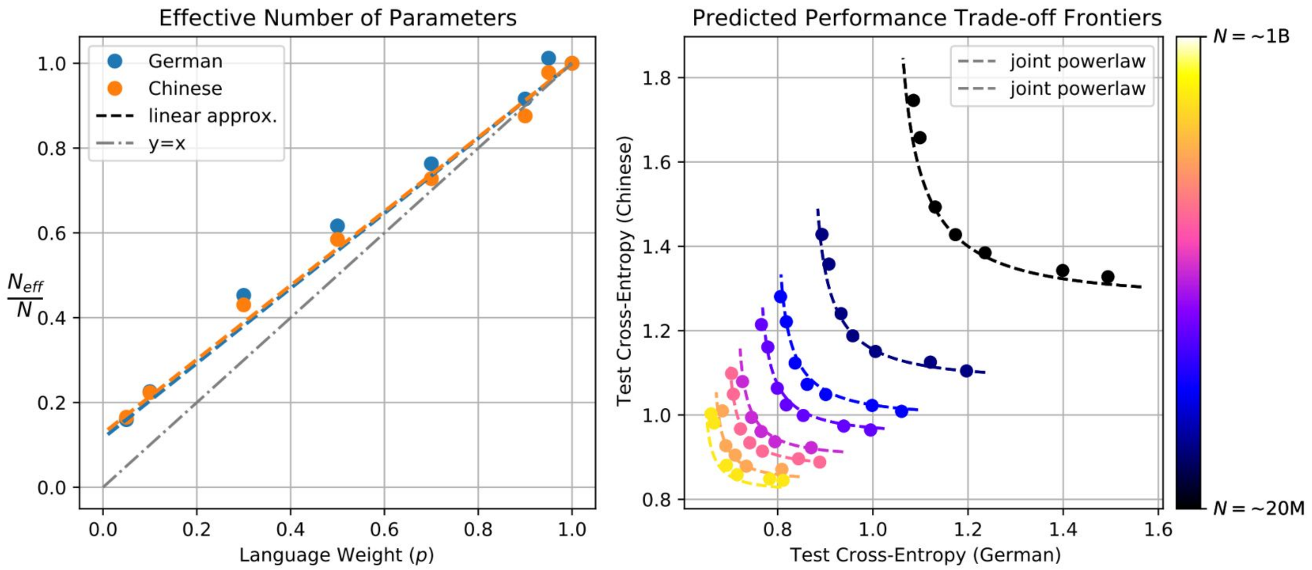
Guiding Task Balancing: En⟶De+Zh





- Almost perfectly captures the full task performance frontier across a variety of model scales.

Guiding Task Balancing: Simpler Models



- A simpler linear model is still able to perform relatively well
  - Requires training less models/task weightings

$$\hat{f}_i(p) = c_1(p-1) + 1.$$

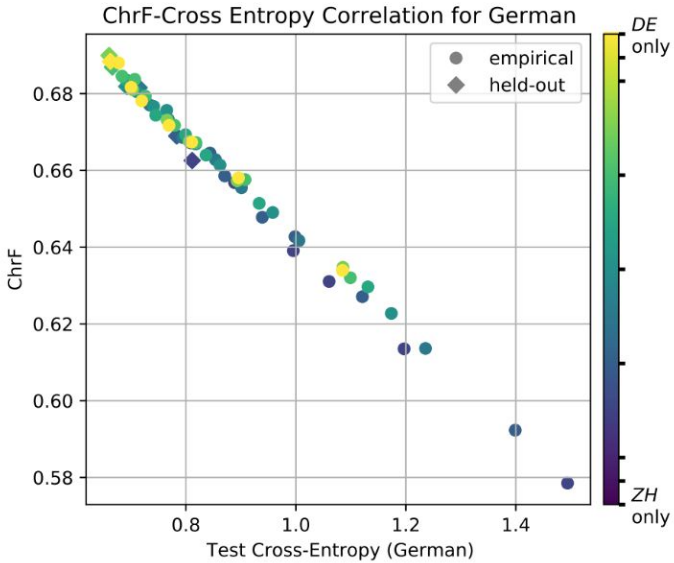
Translation Quality

- In MT research, quality is often measured via *automatic metrics* opposed to cross-entropy
  - BLEU, ChrF, BLEURT, COMET, …
  - These metrics take into account the *decoding* problem
  - Likelihood might not correlate human preference in certain situations

Translation Quality

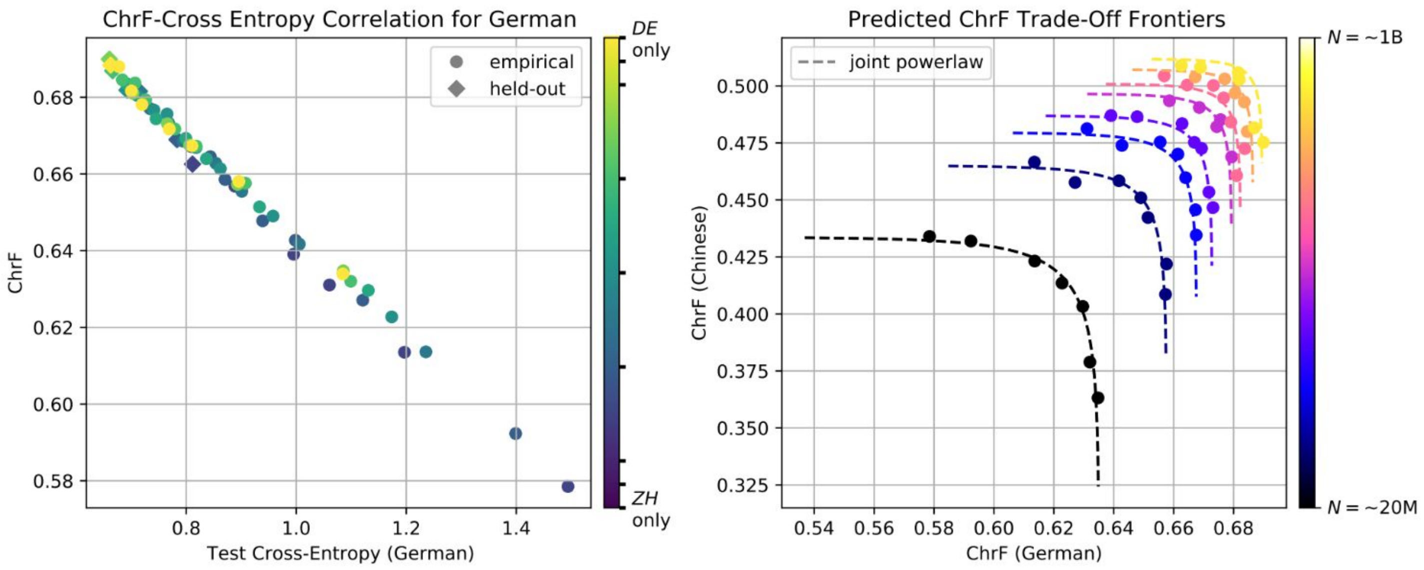
- In MT research, quality is often measured via *automatic metrics* opposed to cross-entropy
  - BLEU, ChrF, BLEURT, COMET, …
  - These metrics take into account the *decoding* problem
  - Likelihood might not correlate human preference in certain situations
- To ensure the practical applicability of results, we repeat our analysis for ChrF and BLEURT
  - Obtain translations from model by decoding with *beam search*

Translation Quality: ChrF



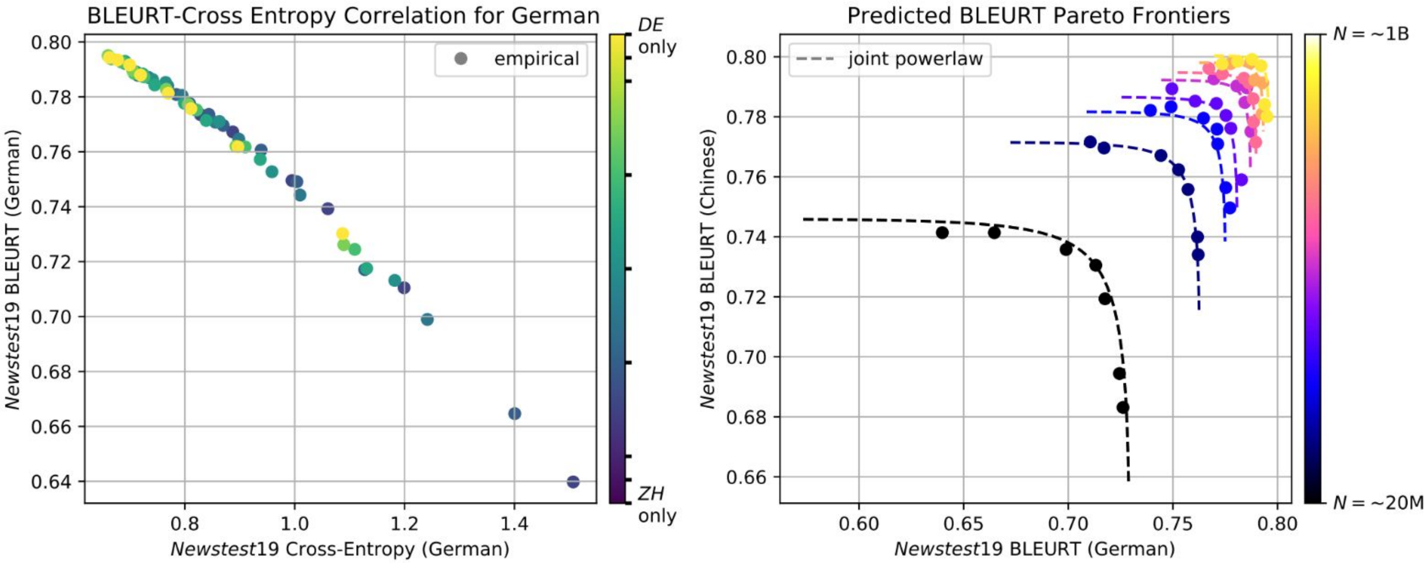
- ChrF has an almost linear relationship with perplexity

Translation Quality: ChrF



- ChrF has an almost linear relationship with perplexity
- We are able to capture the trade-off frontier well

Translation Quality: BLEURT



- Similar findings for BLEURT in *out-of-domain* test sets

Google

Conclusion & Future Work

- The scaling behaviour of multilingual models in an <u>interference</u> scenario is surprisingly simple
  - Almost constant scaling independent of task weight
  - Language/task similarity plays a limited role
  - “Direction” matters for parameter sharing

Conclusion & Future Work

- The scaling behaviour of multilingual models in an interference scenario is surprisingly simple
  - Almost constant scaling independent of task weight
  - Language/task similarity plays a limited role
  - “Direction” matters for parameter sharing

- As future work, we plan investigate:
  - Multi-task optimization for more diverse tasks (LMs + Code Modelling for example)
  - The scaling properties in the transfer scenario

Generalizing to More Tasks

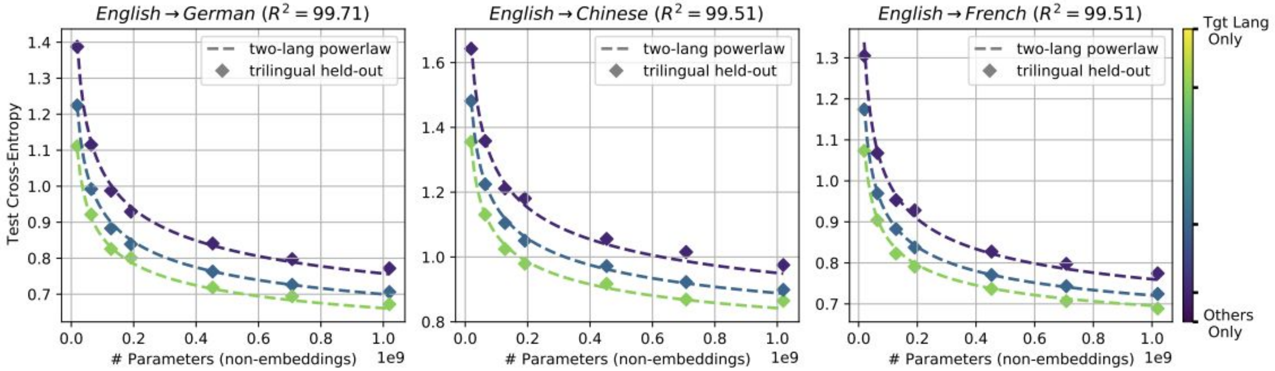


Figure 20. The evolution of the (in-domain) test cross-entropy loss with model size for En→{De, Fr, Zh} models, as well as the fitted scaling laws fitted for En→{De, Zh} (left and middle) and En→{De, Fr} (right). The color represents the weighting of the languages. Note that we don't show the *zero-shot* behavior.

Extension to Low-Resource Languages

On the Pareto Front of Multilingual Neural Machine Translation

Liang Chen[1*] Shuming Ma[2*] Dongdong Zhang[2] Furu Wei[2] Baobao Chang[1†]
Peking University[1]
Microsoft Research[2]
leo.liang.chen@outlook.com chbb@pku.edu.cn
{shumma,dozhang,fuwei}@microsoft.com

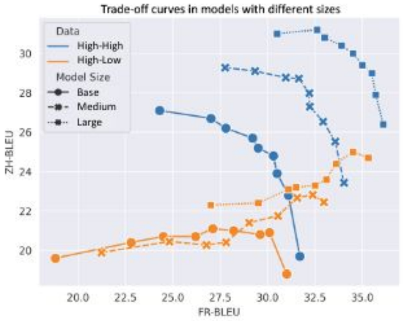


Figure 4: Generalization Performance (BLEU) trade-off curves for English→{French, Chinese} under different model sizes and data distributions. The collapse of Pareto front exists in different model sizes when the training data is imbalanced.