# OUTRAGEOUSLY LARGE NEURAL NETWORKS: THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis,
Quoc Le, Geoffrey Hinton, Jeff Dean

2017

Google Brain

# Noam Shazeer

- 딥러닝 및 자연어 처리 분야
- Attention is All You Need의 공동저자
- Mixture-of-Experts(MoE) 개발
- T5 (Text-to-Text Transfer Transformer) 개발
- Character.AI 공동 창업

# 연구 배경 및 문제점

- 딥러닝 성능은 모델 크기와 데이터 양에 의해 결정

- 기존 모델은 모든 샘플에 대해 모든 파라미터를 사용 → **연산량 증가**

- 문제점:
  1) 연산량이 **모델 크기에 비례**하여 증가
  2) GPU 연산에서 **branching**이 비효율적
  3) 데이터 배치 크기가 작아지면 계산 효율성 저하
  4) 네트워크 대역폭이 병목이 될 가능성
  5) 전문가(Experts) 간 부하 균형 유지가 어렵다

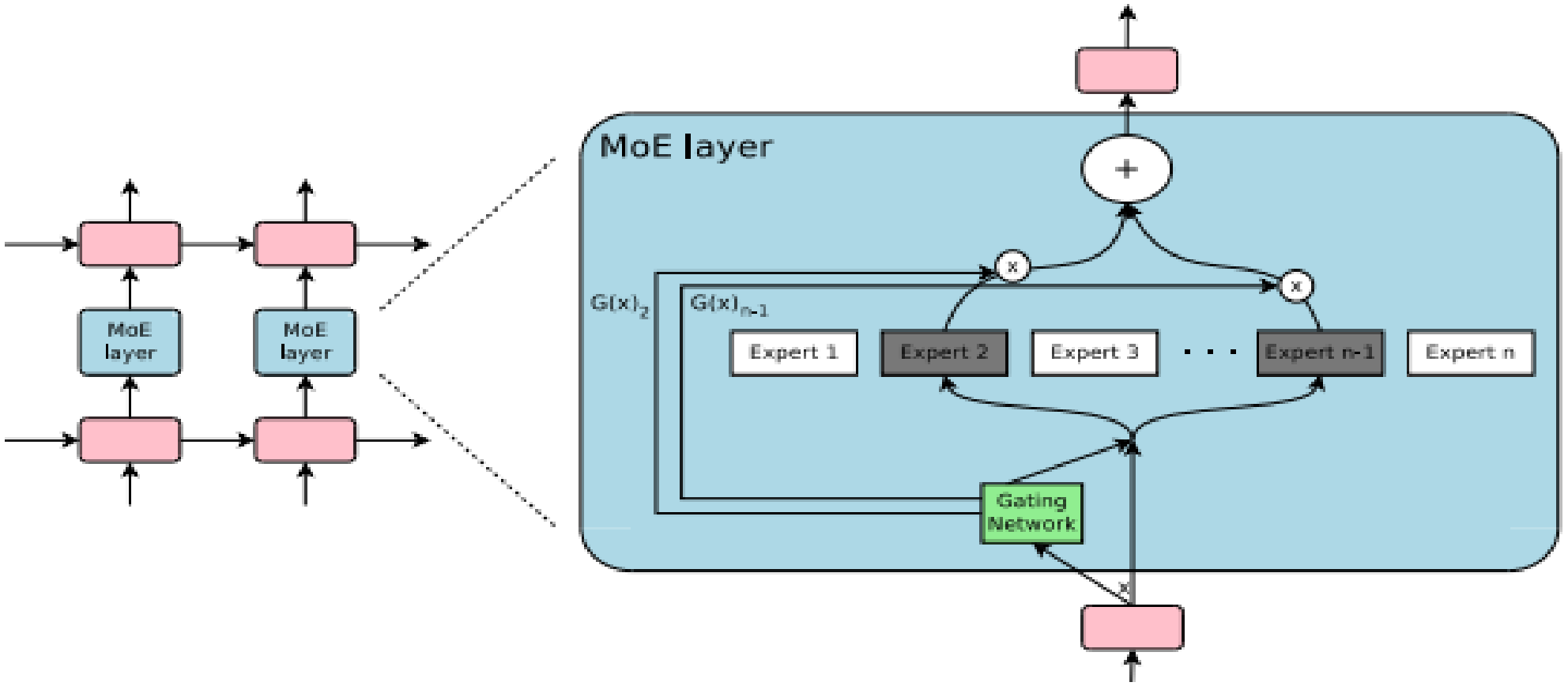
# 1-1. Conditional Computation

- 딥러닝에서 데이터와 모델의 크기가 중요한 요소이다.
- 데이터가 충분히 크면 파라미터 수를 늘리면 예측 정확도가 향상된다.
- 기존의 모델(2017년 이전)은 모든 입력 데이터에 대해 전체 모델이 활성화 되었다.
- 기존의 연구들은 **계산 비용 증가 없이** 모델 용량을 늘리기 위한 **Conditional Computation** 개념이 제안되었다.
- Conditional Computation은 네트워크의 일부만 활성화되며 case별로 다르게 결정됨을 의미한다.

# 1-2. Sparsely-Gated Mixture-of-Experts Layer

- 해당 논문에서는 **Sparsely-Gated Mixture-of-Experts (MoE) Layer**를 도입하여 조건부 연산이 어떻게 효율적인가를 설명한다.
- MoE는 다수의 피드포워드 네트워크(전문가, experts)로 구성되며, 학습 가능한 **게이팅 네트워크**가 각 입력에 대해 사용할 전문가 조합을 결정한다.
- MoE를 **자연어 처리(NLP) 문제**에 적용하여 **언어 모델링(Language Modeling) 및 기계 번역(Machine Translation)에서 성능을 크게 향상**시킨다.

# 1-2. Sparsely-Gated Mixture-of-Experts Layer



# 1-3. Related Work on Mixtures of Experts

- MoE는 1991년 Jacobs 등에 의해 처음 소개 되었으며, 다양한 형태로 발전되어 옴
- SVM, Gaussain Process, Dirichlet Process 등을 기반으로 한 MoE 모델들이 제안됨
- 기존엔 MoE가 모델 전체의 최상위 레이어였지만,
  해당 논문에서는 딥러닝 모델 내에 계층적으로 MoE를 적용하는 방식을 소개

# 2-1. Structure

- 여러 개의 Expert Networks와 Gating Network로 구성
- 입력 x가 주어지면, 게이팅 네트워크 G(x)가 $E_i(x)$에 대한 가중치를 계산

$$y = \sum_{i=1}^{n} G(x)_i E_i(x)$$

# 2-2. Gating Network

- Simple Way

$$G_\sigma(x) = Softmax(x \cdot W_g)$$

- Noisy Top-K Gating

$$G(x) = Softmax(KeepTopK(H(x), k))$$

$$H(x)_i = (x \cdot W_g)_i + StandardNormal() \cdot Softplus((x \cdot W_{noise})_i)$$

$$KeepTopK(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v. \\ -\infty & \text{otherwise.} \end{cases}$$

### 3. Addressing Perfomance Challenges

# 3-1. Shrinking Batch Problem

- MoE 모델에서는 입력 데이터가 게이팅 네트워크를 통해 소수의 Experts에게만 분배됨
  즉, 각 미니배치 내의 샘플이 **일부 전문가에게 집중**된다.
- 이로 인해 **활성화된 전문가의 미니배치 크기가 줄어드는 문제**가 발생합니다.
- 이는 **GPU 및 병렬 처리**에서 비효율적인 연산을 초래한다.

$$\frac{kB}{n} < b, \text{(B: 배치크기, n: 전문가 개수, k: 활성된 전문가)}$$

### 3. Addressing Perfomance Challenges

# 3-1. Shrinking Batch Problem (solution.)

1. 데이터 병렬성과 모델 병렬성을 혼합하여 전문가당 배치 크기를 증가
    - 동일한 모델을 여러 GPU에서 동시에 실행하는 데이터 병렬학습
    - 전문가 네트워크를 특정 GPU에 할당하는 모델 병렬학습 **(전문가 당 배치 크기의 증가)**
2. Convolutionality
    - MoE 레이어를 계층 사이에 삽입하여 **한 번 호출될 때 time steps의 데이터를 한번에 처리**한다.
3. RNN 기반 모델에서는 배치 크기를 늘리는 추가 최적화 기법을 적용
    - Gradient Checkpointing
    - Recomputing Forward Activations

# 3-2. Network Bandwidth

- 분산 환경에서 전문가 네트워크 간 데이터 이동량이 많으면 네트워크 병목 현상이 발생
- 해결책
  - Stationary Experts
  - 전문가의 Hidden layer 크기 증가
  - 계층적 MoE 구조 (1차 게이팅: 상위 전문가 선택, 2차게이팅: 세부 전문가 선택)
  - 배치 크기 최적화
  - 데이터 공유 최소화

# 4. Balancing Expert Utilization

- 게이팅 네트워크가 특정 전문가를 반복적으로 선택하면 문제가 발생한다.

- 해결책
  - Importance Loss
  - Load Balancing Loss

$$Importance(X) = \sum_{x \in X} G(x)$$

$$L_{importance}(X) = w_{importance} \cdot CV(Importance(X))^2$$

$$Load(X)_i = \sum_{x \in X} P(x, i)$$

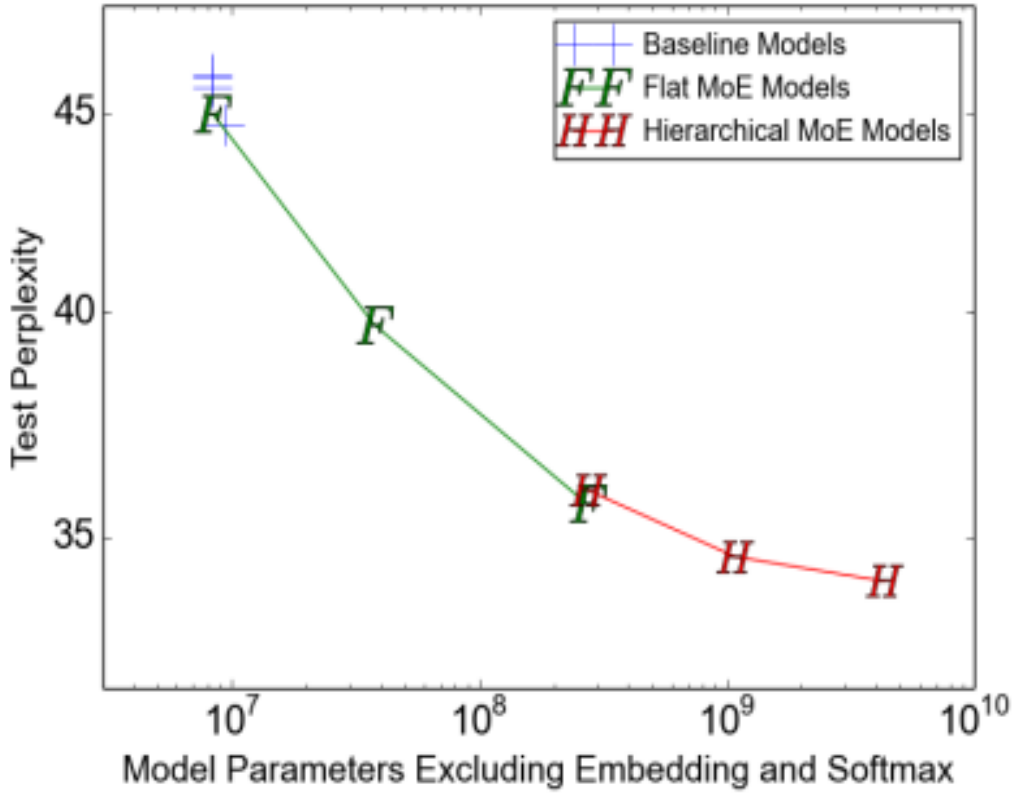
$$L_{load}(X) = w_{load} \cdot CV(Load(X))^2$$

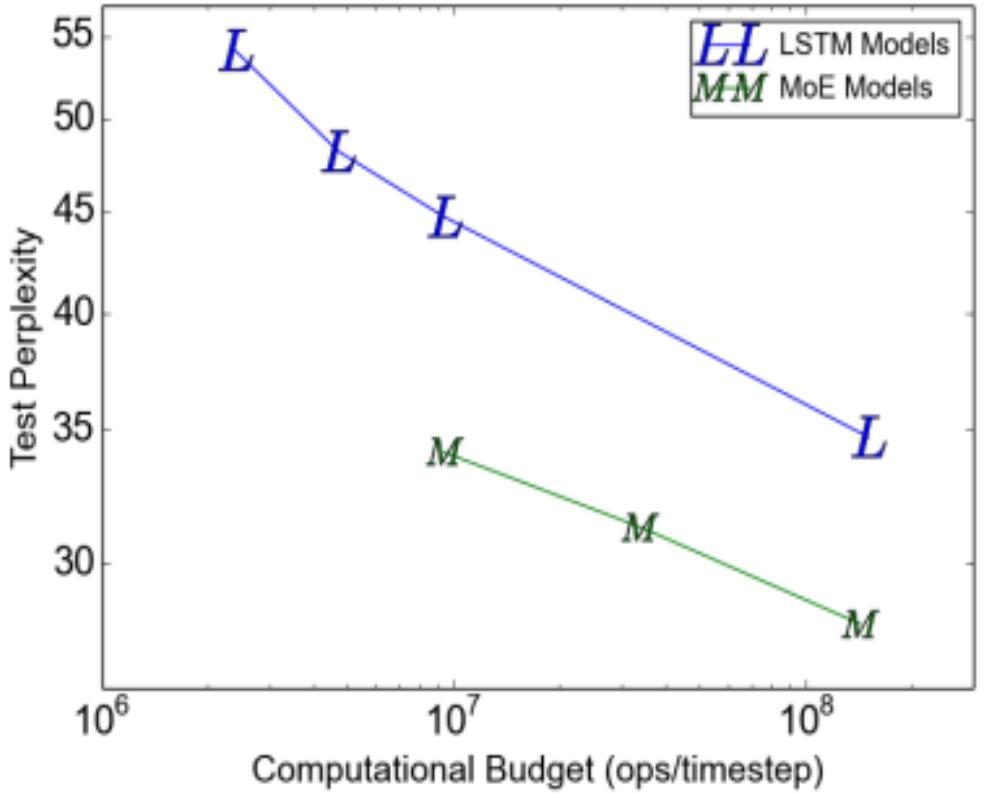
# 4. Balancing Expert Utilization

Table 6: Experiments with different combinations of losses.

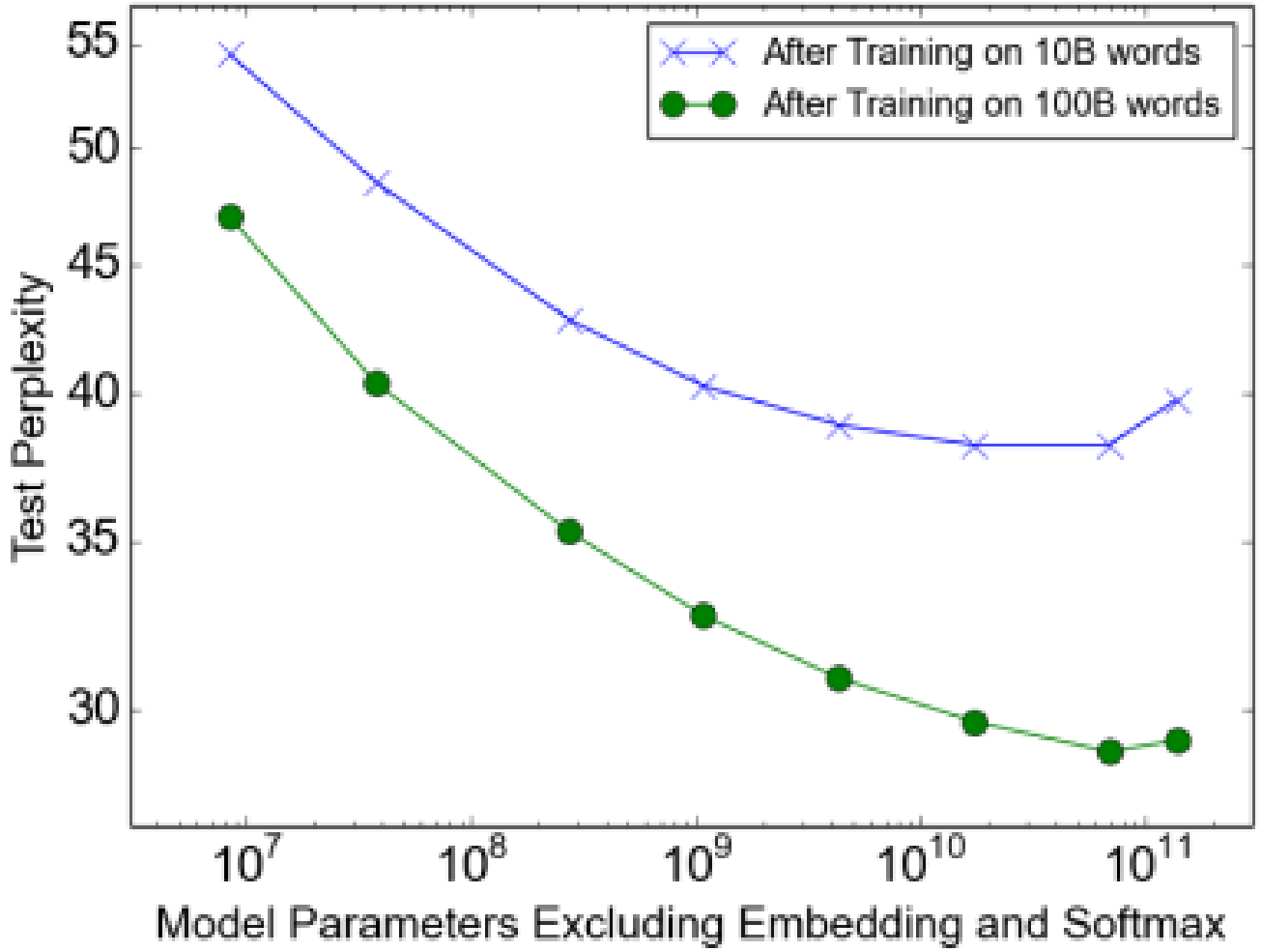
| $w_{importance}$ | $w_{load}$ | Test Perplexity | $CV(Importance(X))$ | $CV(Load(X))$ | $\frac{max(Load(X))}{mean(Load(X))}$ |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 39.8 | 3.04 | 3.01 | 17.80 |
| 0.2 | 0.0 | **35.6** | 0.06 | 0.17 | 1.47 |
| 0.0 | 0.2 | 35.7 | 0.22 | 0.04 | 1.15 |
| 0.1 | 0.1 | **35.6** | 0.06 | 0.05 | 1.14 |
| 0.01 | 0.01 | 35.7 | 0.48 | 0.11 | 1.37 |
| 1.0 | 1.0 | 35.7 | 0.03 | 0.02 | **1.07** |

# 5-1. 1B Word Language Modeling Benchmark



# 5-2. 100B Word Google News Corpus



# 5-3. Machine Translation (Single Language)

Table 2: Results on WMT'14 En→ Fr newstest2014 (bold values represent best results).

| Model | Test Perplexity | Test BLEU | ops/timenstep | Total #Parameters | Training Time |
|---|---|---|---|---|---|
| MoE with 2048 Experts | 2.69 | 40.35 | 85M | 8.7B | 3 days/64 k40s |
| MoE with 2048 Experts (longer training) | **2.63** | **40.56** | 85M | 8.7B | 6 days/64 k40s |
| GNMT (Wu et al., 2016) | 2.79 | 39.22 | 214M | 278M | 6 days/96 k80s |
| GNMT+RL (Wu et al., 2016) | 2.96 | 39.92 | 214M | 278M | 6 days/96 k80s |
| PBMT (Durrani et al., 2014) | | 37.0 | | | |
| LSTM (6-layer) (Luong et al., 2015b) | | 31.5 | | | |
| LSTM (6-layer+PosUnk) (Luong et al., 2015b) | | 33.1 | | | |
| DeepAtt (Zhou et al., 2016) | | 37.7 | | | |
| DeepAtt+PosUnk (Zhou et al., 2016) | | 39.2 | | | |

Table 3: Results on WMT'14 En → De newstest2014 (bold values represent best results).

| Model | Test Perplexity | Test BLEU | ops/timestep | Total #Parameters | Training Time |
|---|---|---|---|---|---|
| MoE with 2048 Experts | **4.64** | **26.03** | 85M | 8.7B | 1 day/64 k40s |
| GNMT (Wu et al., 2016) | 5.25 | 24.91 | 214M | 278M | 1 day/96 k80s |
| GNMT +RL (Wu et al., 2016) | 8.08 | 24.66 | 214M | 278M | 1 day/96 k80s |
| PBMT (Durrani et al., 2014) | | 20.7 | | | |
| DeepAtt (Zhou et al., 2016) | | 20.6 | | | |

Table 4: Results on the Google Production En→ Fr dataset (bold values represent best results).

| Model | Eval Perplexity | Eval BLEU | Test Perplexity | Test BLEU | ops/timestep | Total #Parameters | Training Time |
|---|---|---|---|---|---|---|---|
| MoE with 2048 Experts | **2.60** | **37.27** | **2.69** | **36.57** | 85M | 8.7B | 1 day/64 k40s |
| GNMT (Wu et al., 2016) | 2.78 | 35.80 | 2.87 | 35.56 | 214M | 278M | 6 days/96 k80s |

# 5-3. Machine Translation (Multilingual)

Table 5: Multilingual Machine Translation (bold values represent best results).

| | GNMT-Mono | GNMT-Multi | MoE-Multi | MoE-Multi vs. GNMT-Multi |
|---|---|---|---|---|
| Parameters | 278M / model | 278M | 8.7B | |
| ops/timestep | 212M | 212M | 102M | |
| training time, hardware | various | 21 days, 96 k20s | **12 days, 64 k40s** | |
| Perplexity (dev) | | 4.14 | **3.35** | -19% |
| French → English Test BLEU | 36.47 | 34.40 | **37.46** | +3.06 |
| German → English Test BLEU | 31.77 | 31.17 | **34.80** | +3.63 |
| Japanese → English Test BLEU | 23.41 | 21.62 | **25.91** | +4.29 |
| Korean → English Test BLEU | 25.42 | 22.87 | **28.71** | +5.84 |
| Portuguese → English Test BLEU | 44.40 | 42.53 | **46.13** | +3.60 |
| Spanish → English Test BLEU | 38.00 | 36.04 | **39.39** | +3.35 |
| English → French Test BLEU | 35.37 | 34.00 | **36.59** | +2.59 |
| English → German Test BLEU | **26.43** | 23.15 | 24.53 | +1.38 |
| English → Japanese Test BLEU | **23.66** | 21.10 | 22.78 | +1.68 |
| English → Korean Test BLEU | **19.75** | 18.41 | 16.62 | -1.79 |
| English → Portuguese Test BLEU | **38.40** | 37.35 | 37.90 | +0.55 |
| English → Spanish Test BLEU | 34.50 | 34.25 | **36.21** | +1.96 |

# 6. Conclusion

- Conditional Computation을 활용하여 기존 DL 모델보다 높은 모델 용량을 효율적으로 학습

- Sparsely-Gated Mixtur-of-Experts 레이어를 적용하여 대규모 데이터에서 성능을 극적으로 향상시킴

- 언어모델 뿐만 아니라 이미지처리, 음성 인식에서도 사용 가능