ADAPTING PRETRAINED MODELS FOR MACHINE TRANSLATION

Aditya Kurniawan

MOTIVATIONS

Pre-trained Models

1. We want to benefit from the knowledge that was gathered in the pre-trained models
2. We want to benefit from the potential performance boost from pre-trained models

Fine-tuning with Adapters

1. More stable and robust than naïve fine-tuning
2. Improves efficiency of fine-tuning large pre-trained models

GOALS

QUALITY OF PRE-TRAINED MODELS

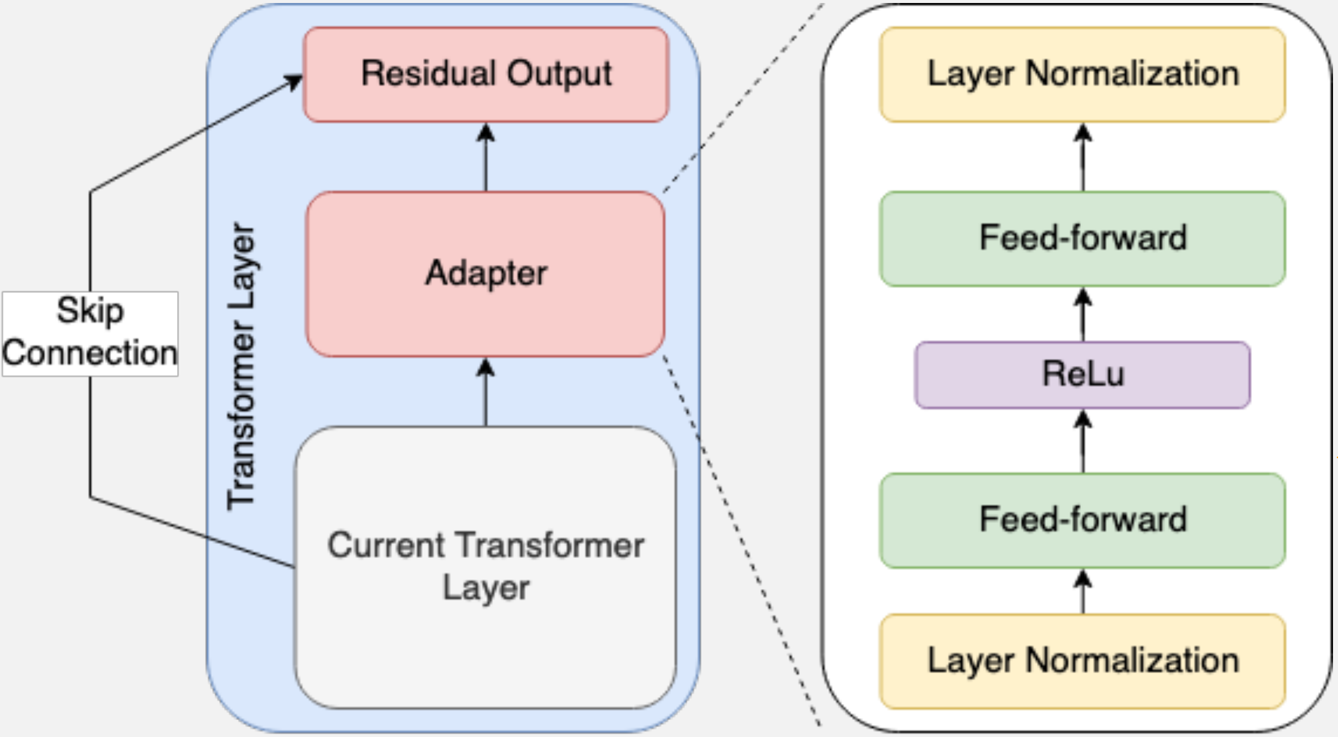
1. Investigate the quality of pre-trained models when fine-tuned with adapters

EFFECTIVENESS OF ADAPTERS

2. Investigate the importance of adapters in encoder or decoder
3. Investigate the importance of the actual weights in the pre-trained models when fine-tuned with adapters
4. Investigate techniques to reduce the original pre-trained BERT model size when fine-tuned with adapters

METHODOLOGY

ADAPTER MODULE
(PFEIFFER ET AL. 2020)

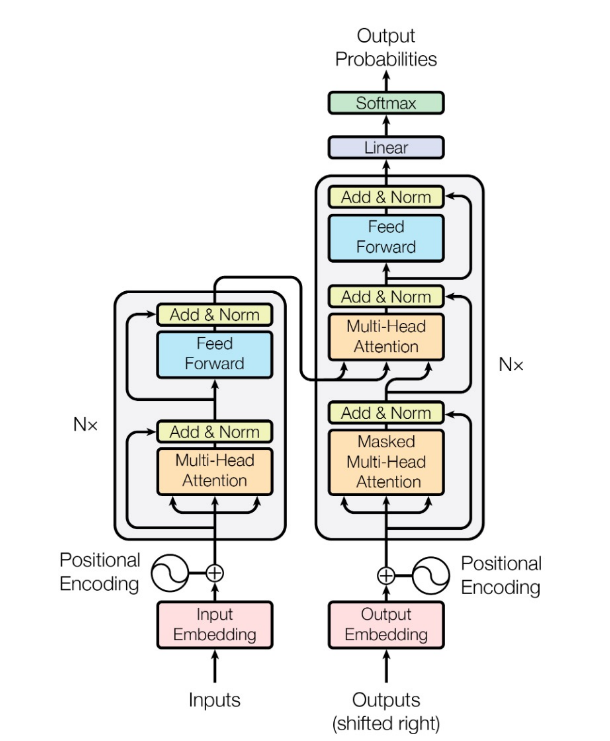


Fine-tuning Process

1. Seed model is firstly pre-trained on source domain data / source task.
2. During fine-tuning stage only adapter parameters are trained.

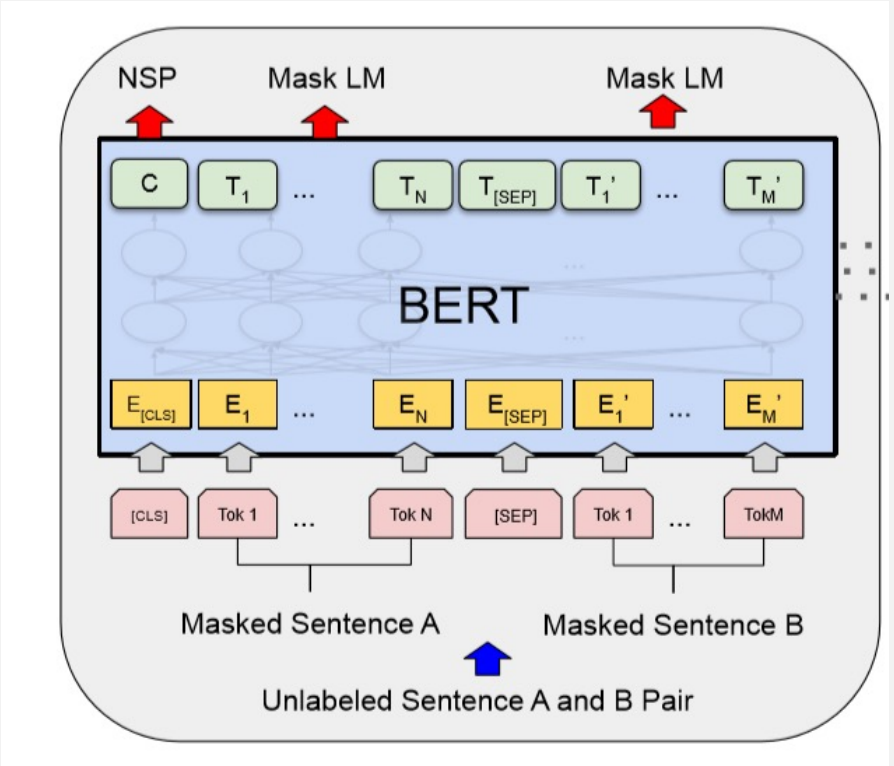
Bottleneck layer with reduction ratio = R

TRANSFORMER
(VASWANI ET AL. 2017)



Transformer architecture diagram from Vaswani et al. 2017

BERT
(DEVLIN ET AL 2018)



BERT diagram from Devlin et al. 2018

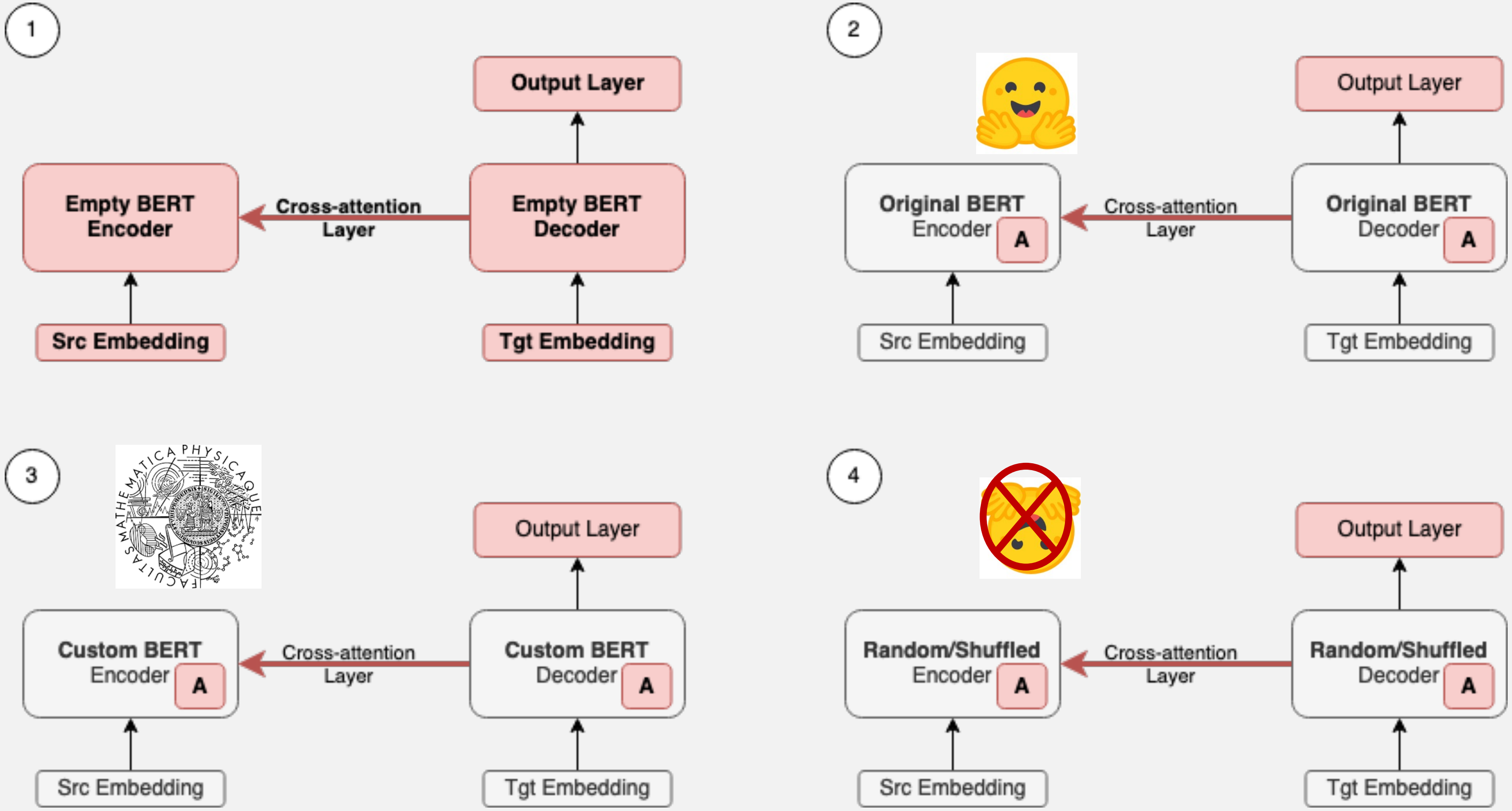
What we used from BERT?:
- Pre-trained weights
- Hyperparameters
    - Number of layers
    - Attention head numbers
    - etc

TASKS AND DATASET

	LANGUAGE MODEL (PRE-TRAINING)			MACHINE TRANSLATION (BASELINE EXP 1)			MACHINE TRANSLATION (FINE-TUNING)		
	TRAIN	DEV	TEST	TRAIN	DEV	TEST	TRAIN	DEV	TEST
IWSLT 2014	V	V	V	V	V	V	V	V	V
IWSLT 2014 + WMT 2019 (500K)	V	X	X	V	X	X	X	X	X
IWSLT 2014 + WMT 2019 (2M)	V	X	X	V	X	X	X	X	X

ADAPTERS IN MACHINE TRANSLATION

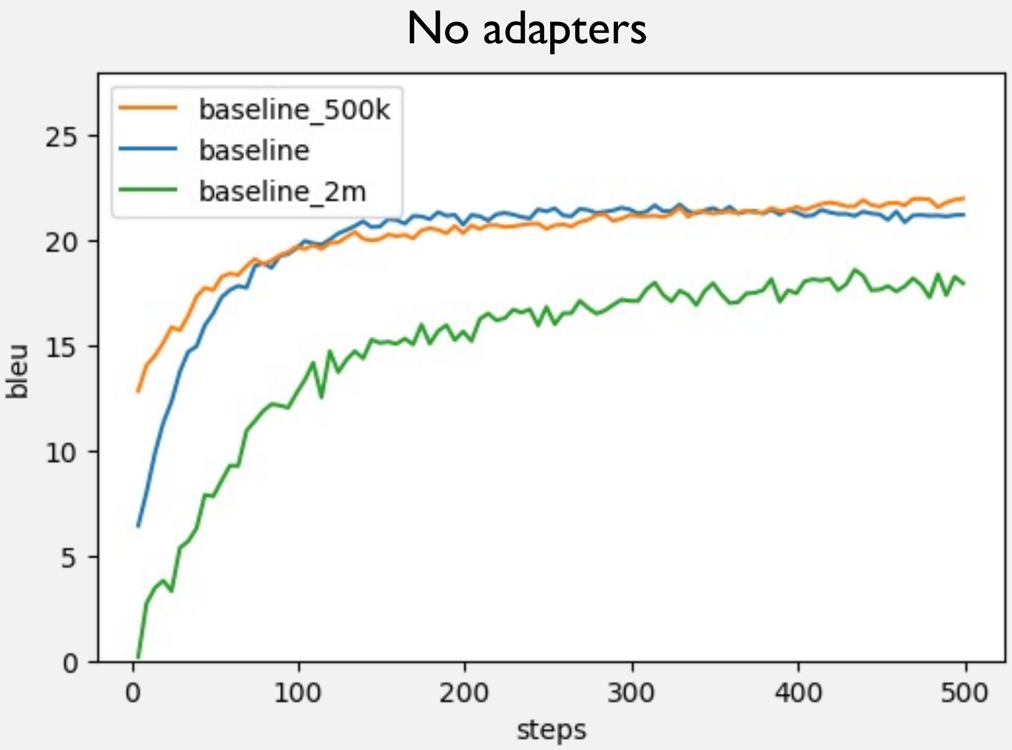
EXPERIMENTS FOR GOAL NO. 1

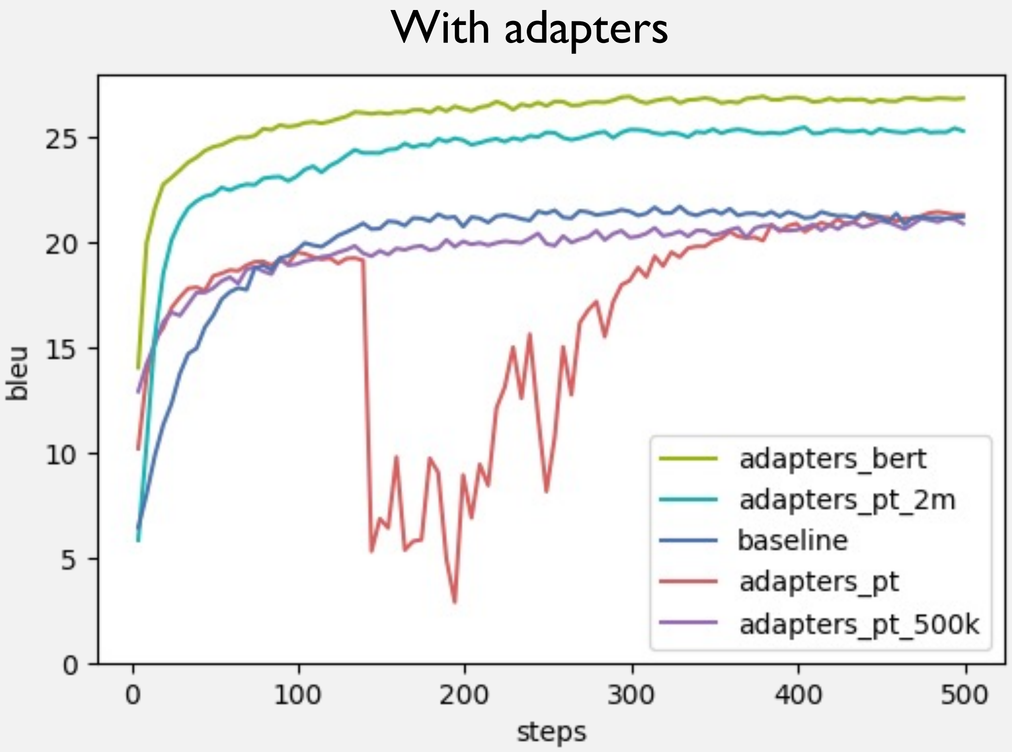


A = Adapters

1. Baseline
2. BERT + Adapters
3. Custom BERT with different volumes of pre-training data
4. Random/Shuffled pre-trained

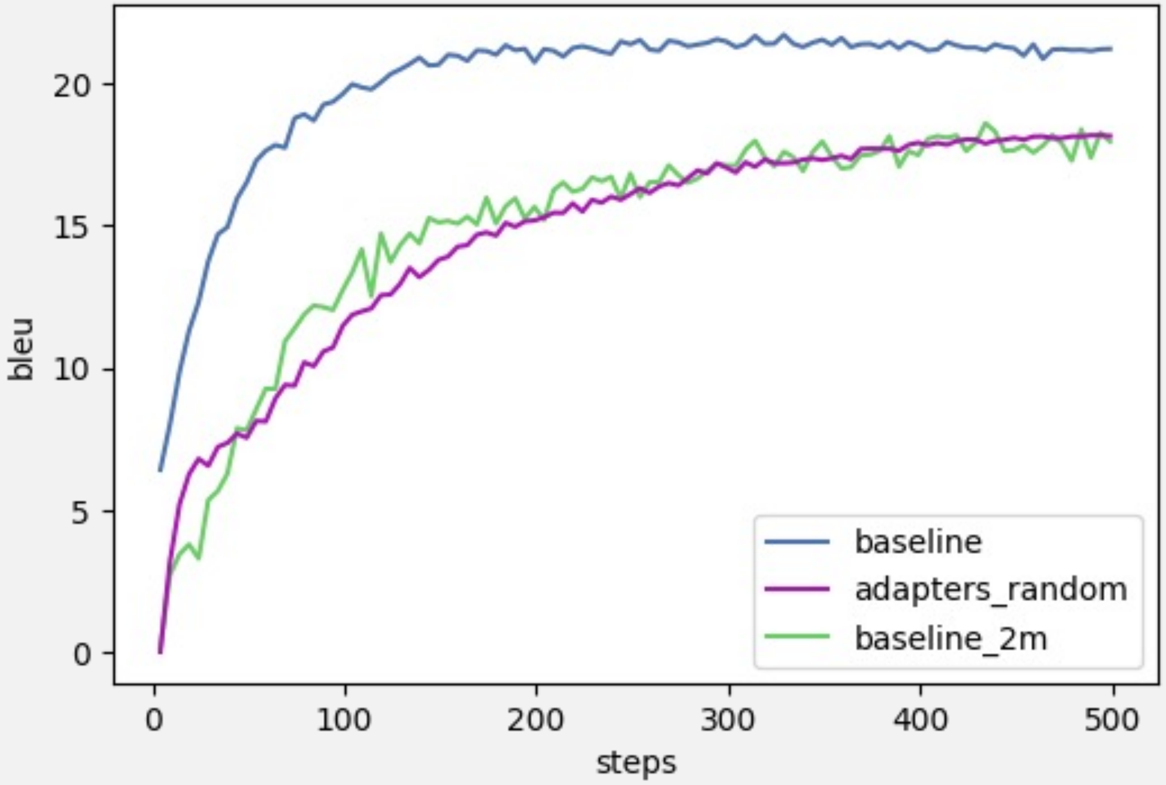
ADDING MORE DATA FOR TRAINING





- Adding more data when training from scratch without adapters doesn't always help
- In contrast with the baseline, when adding more data to the pre-training we can see benefit where the performance of 2m exceeds the 500k

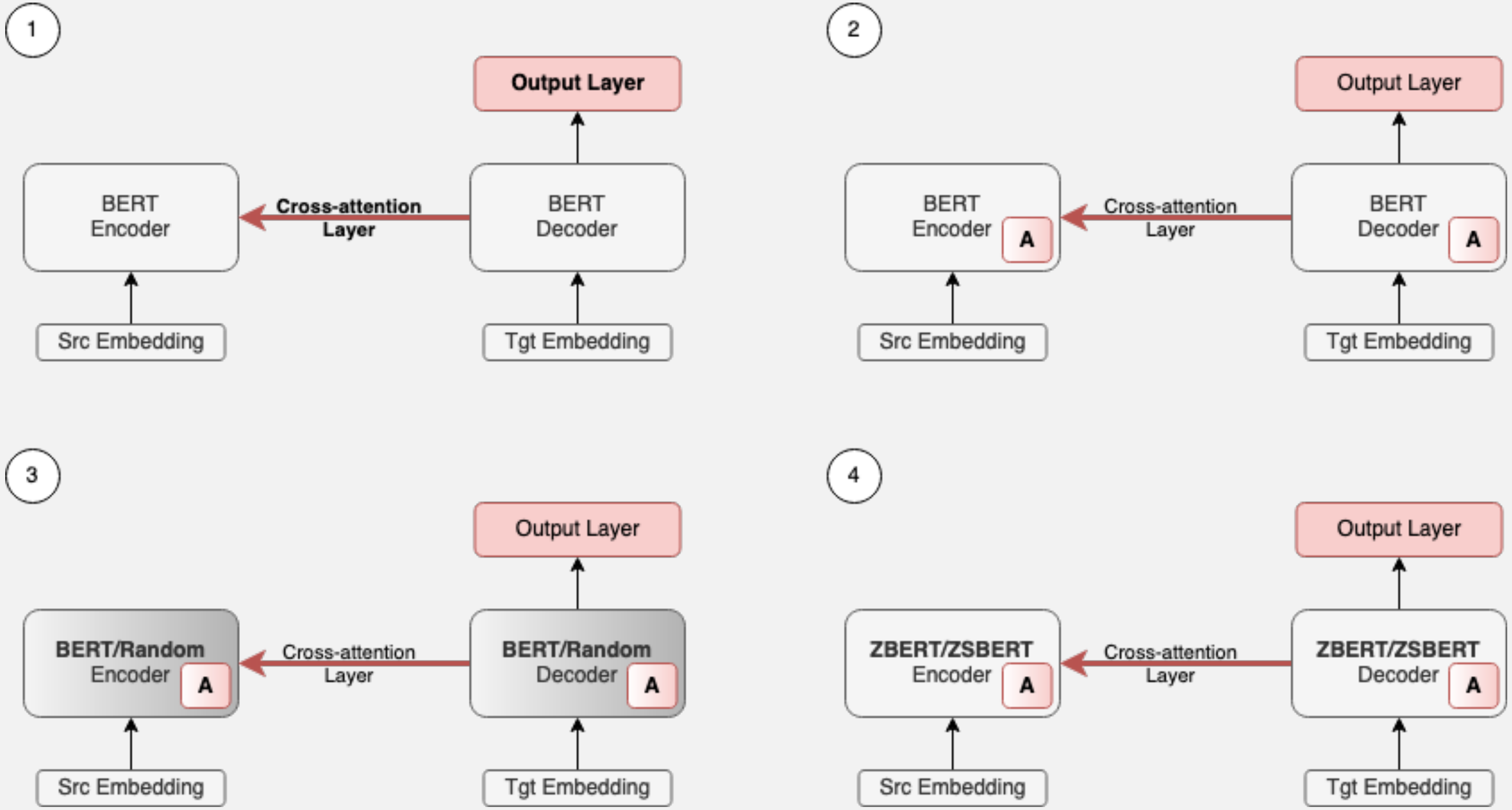
RANDOM PRE-TRAINED VS BASELINE



The performance of the random pre-trained is actually not that bad if compared to the baseline that trained with 2 millions sentence pairs

ADAPTERS EFFECTIVENESS IN MACHINE TRANSLATION

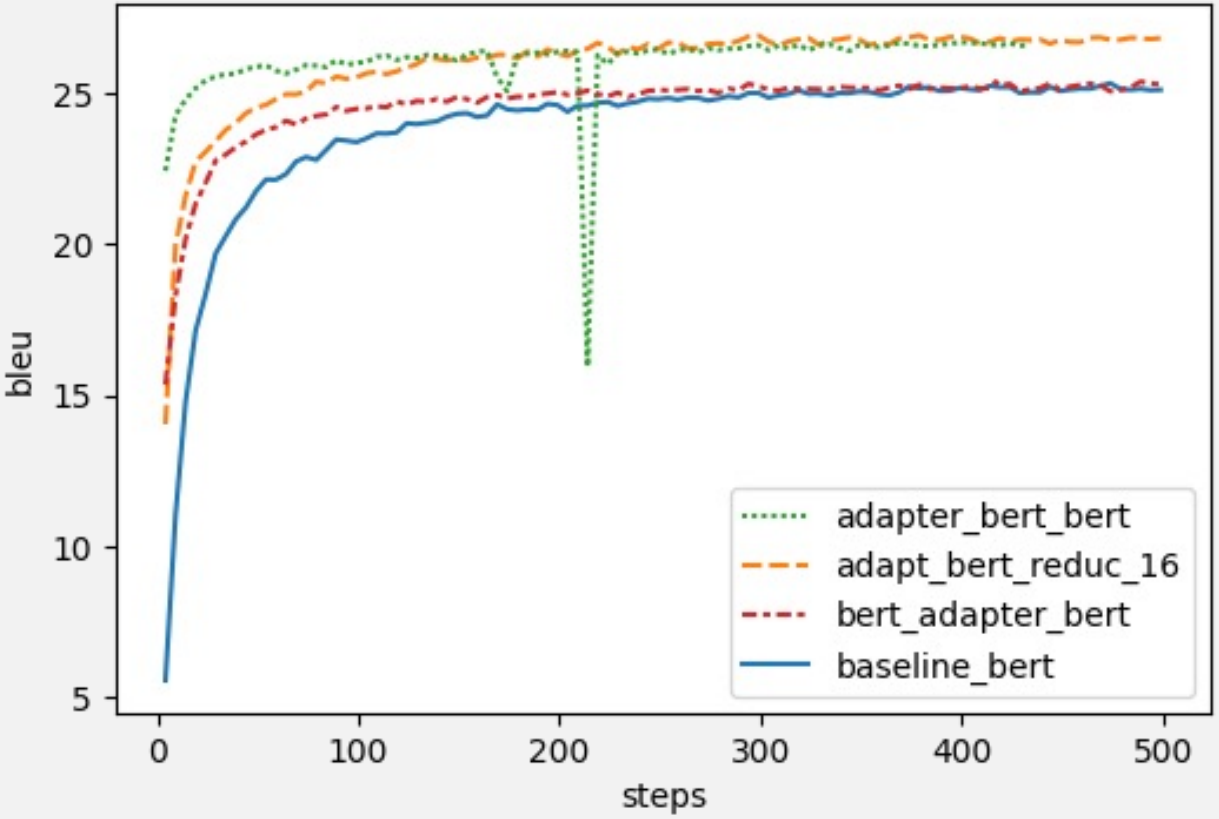
EXPERIMENTS FOR GOALS NO. 2, 3, AND 4



A = Adapters ablated on encoder or decoder

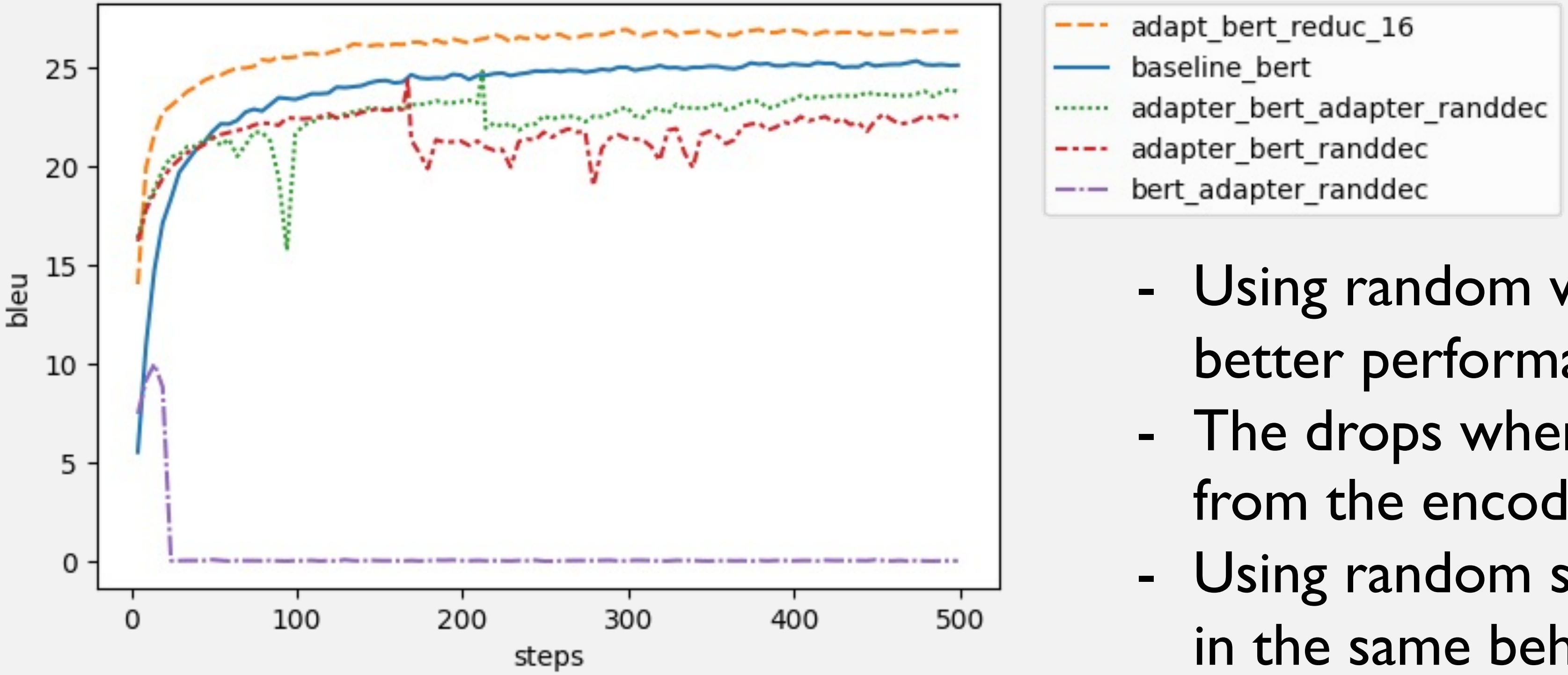
1. Baseline
2. Adapters Position
3. Random pre-trained
4. Pre-trained model size reduction

ADAPTERS POSITION (ENCODER VS DECODER)



- **Green vs Orange line**: removing adapters on the decoder learns faster in the beginning but has no impact on the final performance
- **Red**: removing adapters on the encoder reduces the performance to the baseline level (blue line)

RANDOMLY SET WEIGHTS ON DECODER



- Using random weights on the decoder has better performance than in the encoder
- The drops when the adapter is removed from the encoder
- Using random set weights on encoder results in the same behaviour but lower performance

USING FEWER WEIGHTS: ZBERT AND ZSBERT

Original BERT

W =

2	1	3
4	5	10
7	8	9

zbert

W' =

2	**0**	3
4	**0**	10
7	**0**	9

zsbert

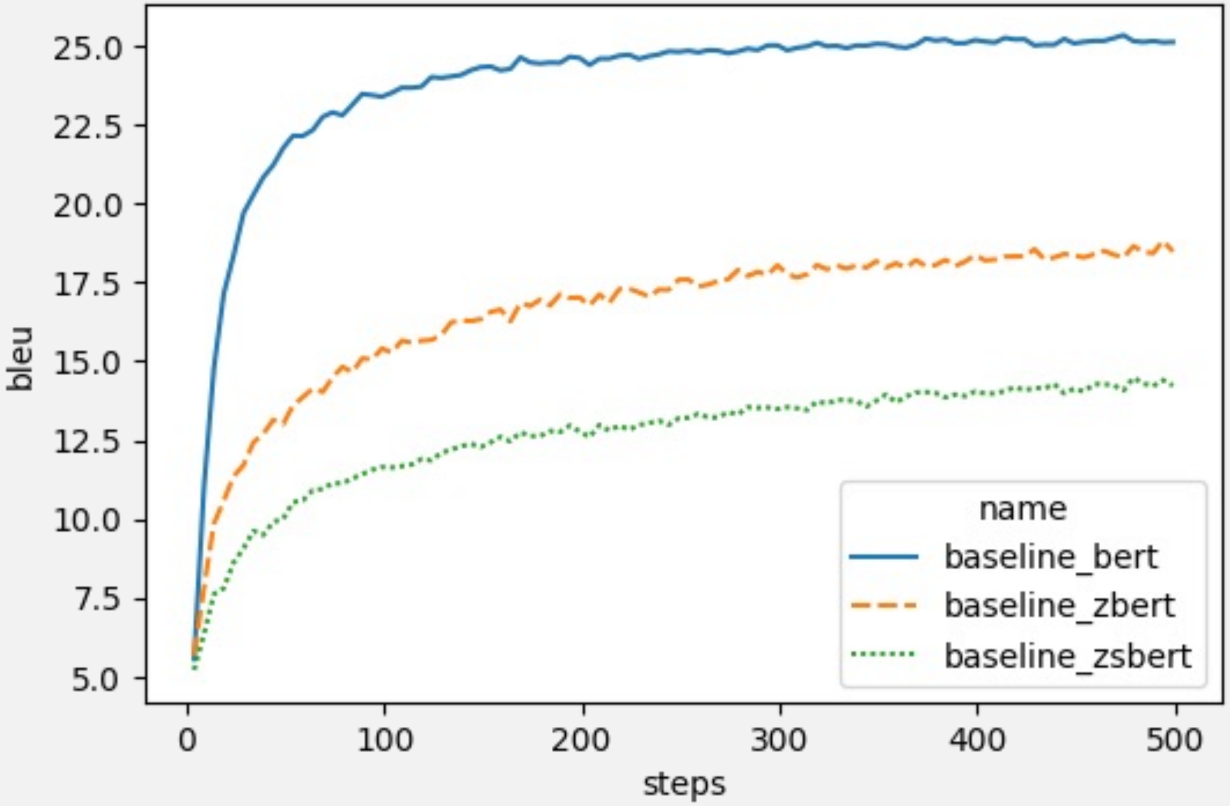
W' =

2	3
4	10
7	9

Rows = Neurons

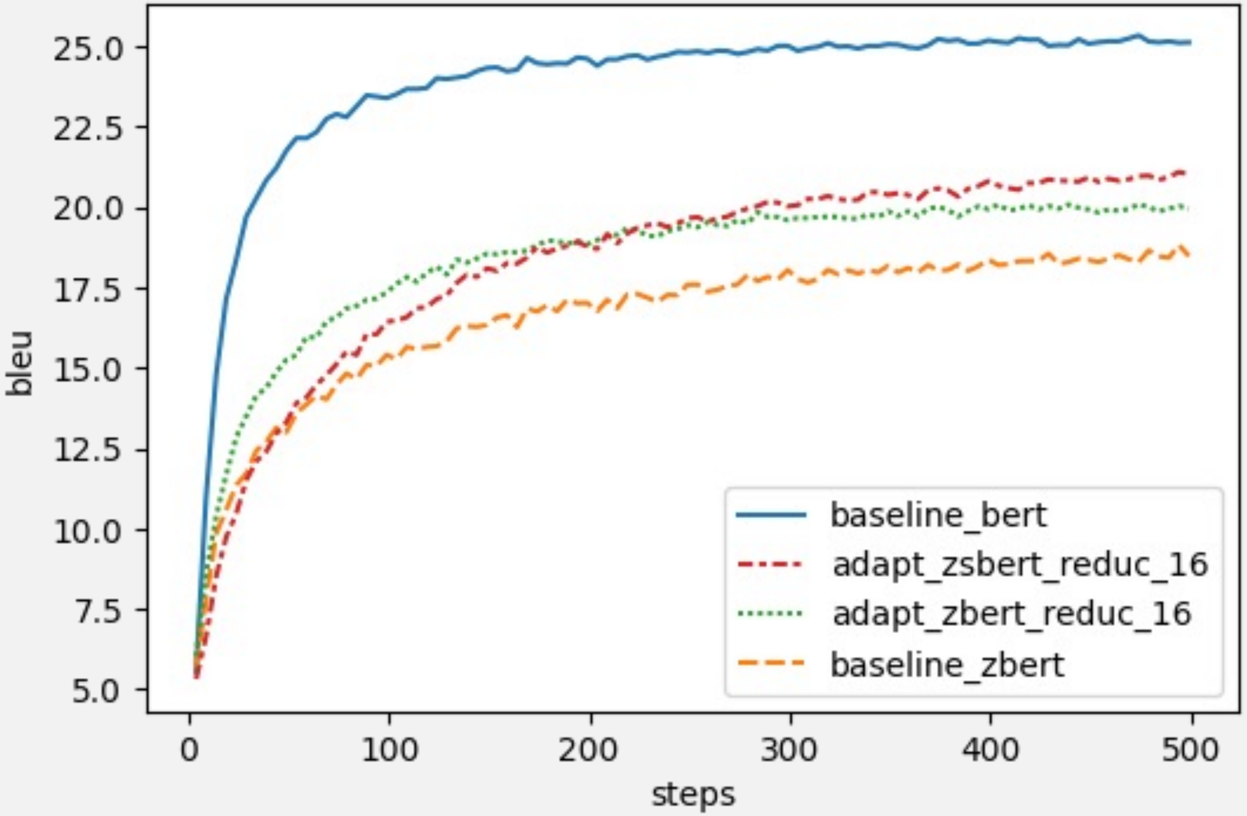
Columns = Features

BASELINE



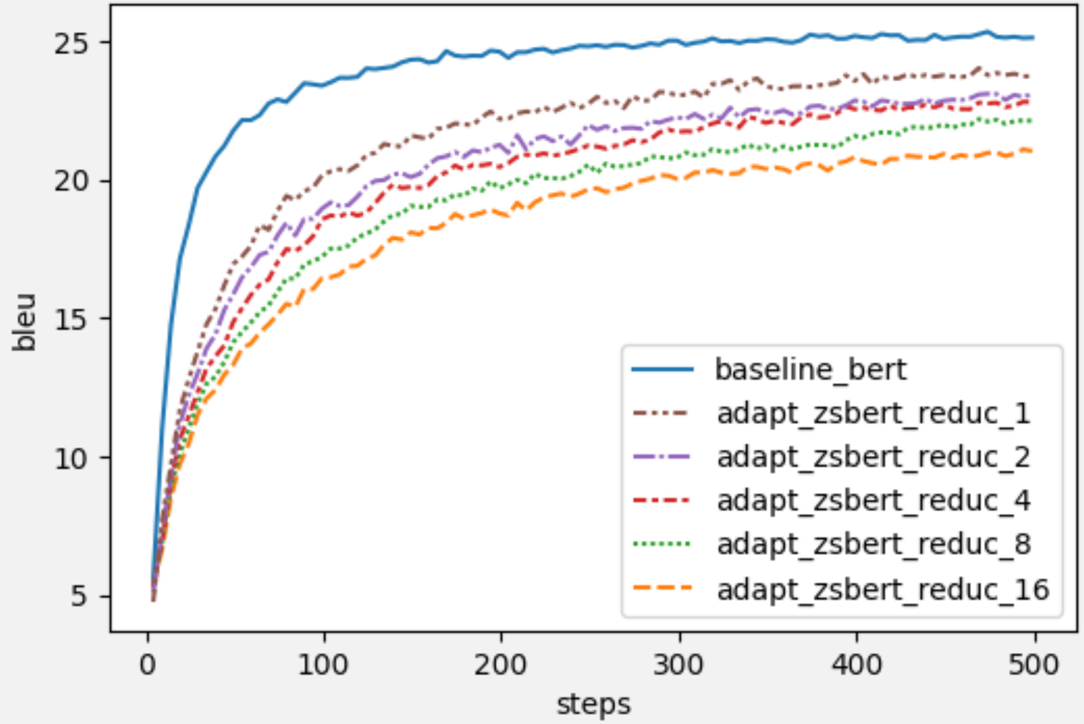
Removing BERT weights arbitrarily clearly has a detrimental impact to the model's performance

BERT SIZE REDUCTION



- Adapters help but not much to recover the performance back to baseline
- Eventually the adapters in ZSBERT manage to outperform ZBERT

BERT SIZE REDUCTION WITH SMALLER REDUCTION RATIO



Name	# Trained Variables	# Untrained Variables	# Total Variables
Adapters ratio 16	7.74M	95.14M	102.88M
Adapters ratio 8	8.17M	95.14M	103.32M
Adapters ratio 4	9.00M	95.14M	104.20M
Adapters ratio 2	10.83M	95.14M	105.98M
Adapters ratio 1	14.38M	95.14M	109.52M
Normal BERT	28.99M	218.81M	247.80M

- Reducing the reduction ratio helps to recover the performance
- Even though more weights are added, the total variables are still way fewer than the original BERT

CONCLUSION

- Investigate the quality of pre-trained models when fine-tuned with adapters
  - ✓ Incorporating more data in pre-training helps the final performance after fine-tuning [compared to training the model from scratch]
  - ✓ Fine-tuning adapters with random pre-trained models achieves on-par performance [compared to training the models from scratch with larger data]

CONCLUSION

- Investigate the importance of adapters in encoder or decoder
  - ✓ Adapters on the encoder side are more important than in the decoder
- Investigate the importance of pre-trained weights in the pre-trained models when fine-tuned with adapters
  - ✓ The actual pre-trained weights are more important in the encoder. Interestingly, when the adapters were injected only to the decoder, the performance dropped to zero.
- Investigate techniques to reduce the original pre-trained BERT weights size with adapters
  - ✓ ZSBERT can match the performance of the baseline when the reduction ratio is not big.

WHAT DID I LEARN?

- Manage to complete 8 different types of experiments
- Adapting code from huggingface for the experiments (both LM and MT)
- Understanding the inside implementation of transformer for debugging
- Learn to integrate the huggingface with WANDB for better training monitoring

Q&A