Using the Transformer

ALBERT (Lan et al., 2019)

Learning goals

- Understand the improvements over BERT
- Parameter sharing
- Disentangling *E* and *H*

SIZE OF EMBEDDING AND HIDDEN LAYER

Disentanglement of E and H

- WordPiece-Embeddings (size E)
  - first layer of the model
  - each token is initially mapped to this embedding
  - context-independent
- In Transformer/BERT:
  - $H = E$
  - down-project E to keys, queries and values of size H/A
  - concatenate resulting embeddings from all A heads
  - results in hidden layer representation of size H
- Implications?

THOUGHTS / IMPLICATIONS

- WordPiece-Embeddings (size E)
  - required representational capacity?
  - probably could be limited w/o loosing much
- Hidden-Layer-Embedding (size H)
  - required representational capacity?
  - depending on how polysemous a word/token might be
  - difficult to say "one size fits all"
  - probably might be better to rather increase this, compared to the WordPiece embeddings

$\rightarrow$ *Setting* $E = H$ *does not allow us to pursue these considerations*

DISENTANGLEMENT SOLVES THIS

- Hidden-Layer-Embeddings (size H) context-dependent
  $\rightarrow$ providing more capacity makes more sense here
- Setting $H >> E$ enlargens model capacity in the hidden layers without increasing the size of the embedding matrix
- $O(V \times H) > O(V \times E + E \times H)$ if $H >> E$

CROSS-LAYER PARAMETER SHARING

- Typically pre-trained transformer-based models are deep and thus have many parameters
- Sharing them as a way to gain parameter efficiency
- Two different places in the network, where sharing can be done
  - Attention parameters
  - FFN parameters
  - (or both)
- Ablations: both; both individually; none

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

Table 4: The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

Source: Lan et al. (2019)

OBSERVATIONS

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

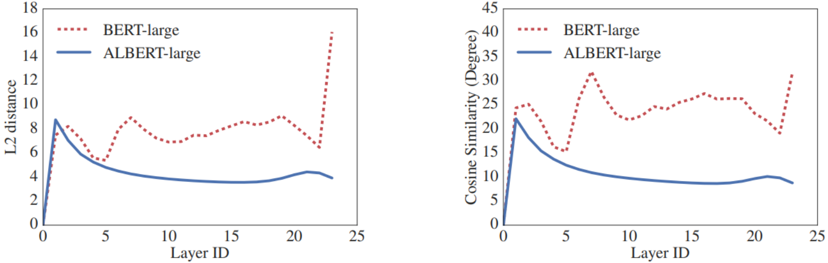
Table 4: The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

Source: Lan et al. (2019)

- (Drastic) reduction of model size (more for sharing FFN weights)
- Sharing parameters hurts performance
  - Worse for models with larger E
  - Worse for sharing FNN compared to Attention weights

$\rightarrow$ **Why?**

CROSS-LAYER PARAMETER SHARING



Source: Lan et al. (2019)

CHANGES IN PRE-TRAINING

Change/Substitution of the NSP objective

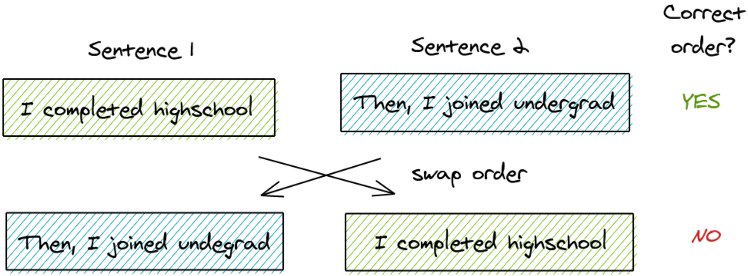
- Previous works questioned the usefulness of NSP

$\rightarrow$ Lan et al. assume that this is due to lacking difficulty

- Introduction of *Sentence-Order Prediction* (SOP) as a new pre-training task
- Positive examples created alike to those from NSP (take two consecutive sentences from the same document)
- Negative examples: Just swap the ordering of sentences

CHANGES IN PRE-TRAINING

Illustration:



Source: Amit Chaudhary

Effectiveness:

SP tasks	Intrinsic Tasks			Downstream Tasks					
	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

Table 5: The effect of sentence-prediction loss, NSP vs. SOP, on intrinsic and downstream tasks.

Source: Lan et al. (2019)

CHANGES IN PRE-TRAINING

$n - gram$ **masking for the MLM task**

- During pre-training BERT single tokens are masked
- Lan et al. mask up to three consecutive tokens
- Choice of *n*:

$$p(n) = \frac{1/n}{\sum_{k=1}^{N} 1/k}$$

ALBERT ▸ LAN ET AL., 2019

Performance differences:

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	0.3x

Source: Lan et al. (2019)

Notes:

- In General: Smaller model size (because of parameter sharing)
- Nevertheless: Scale model up to almost similar size (`xxlarge` version)
- Strong performance compared to BERT