DistilBERT – Paper Presentation

George Tzannetos

Introduction

Huge progress in NLP → main key is applying general pre-trained language representation model in the downstream tasks

Two approaches

- *Feature-based* → like ELMo, task specific architectures, include pre-trained representations as additional features
- *Fine tuning* → GPT, **BERT**, minimal task specific parameters, fine tune all pre-trained parameters on each task

BERT

BERT (**Bidirectional** Encoder Representations from **Transformer**) → was introduced after ELMo and GPT and outperforms them

- use of *transformer* network and *bidirectionality*

Transformer based architecture

Recurrence is not used

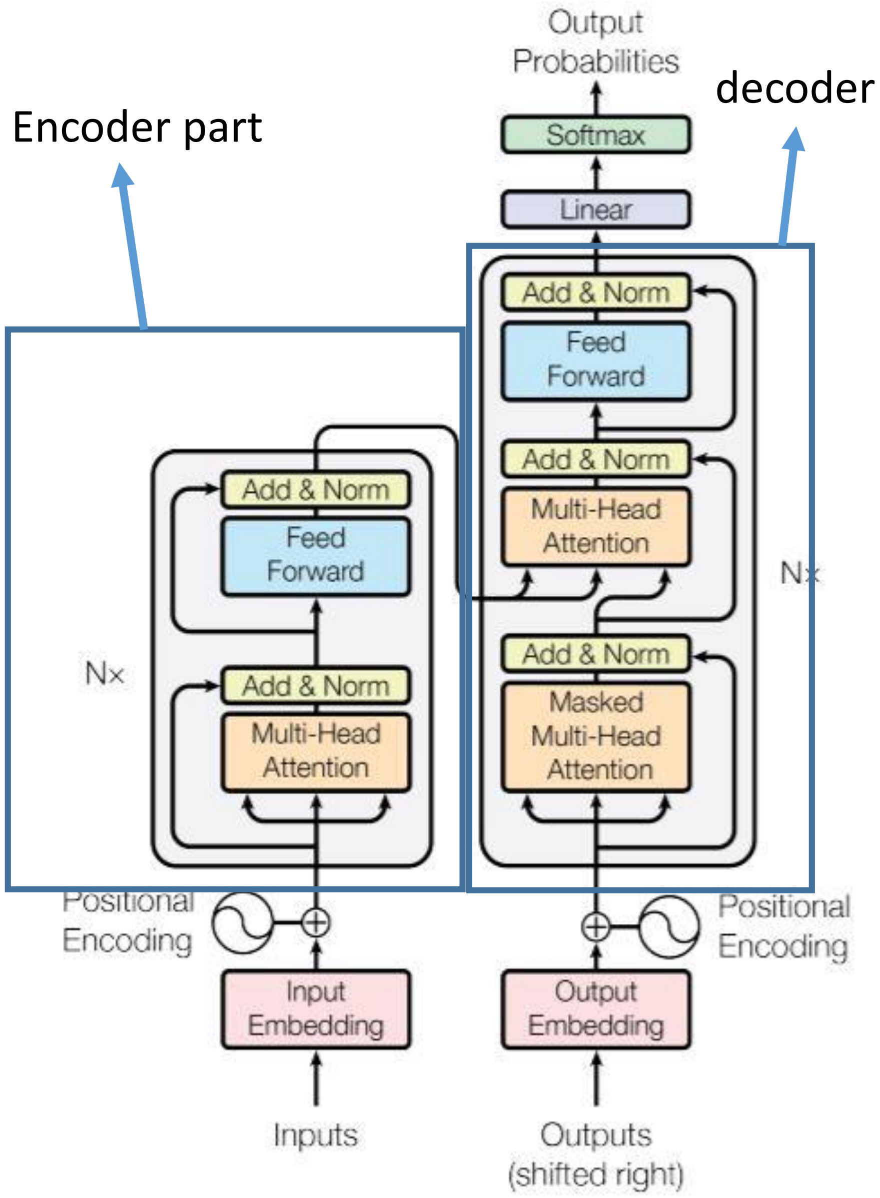
Global dependencies are drawn via self-attention mechanisms

What is a transformer?

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin, Chang, Lee, Toutanova

Transformer – Self Attention

- *Stacked encoder* → multi-head **attention** + feed forward
- *Stacked decoder* → masked multi-head **attention** + multi-head attention + feed forward
- Residual connections



What is self-attention?

<u>Idea</u>: allow the inputs to interact with itself and find who they should pay more attention to

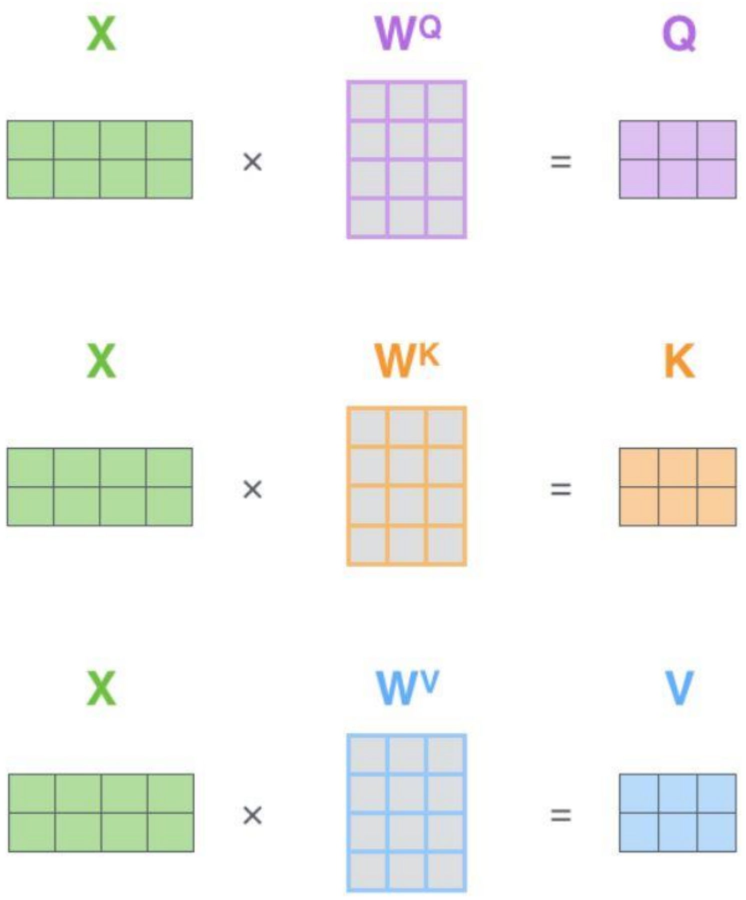
query Q, key K, value V → abstractions introduced to calculate attention

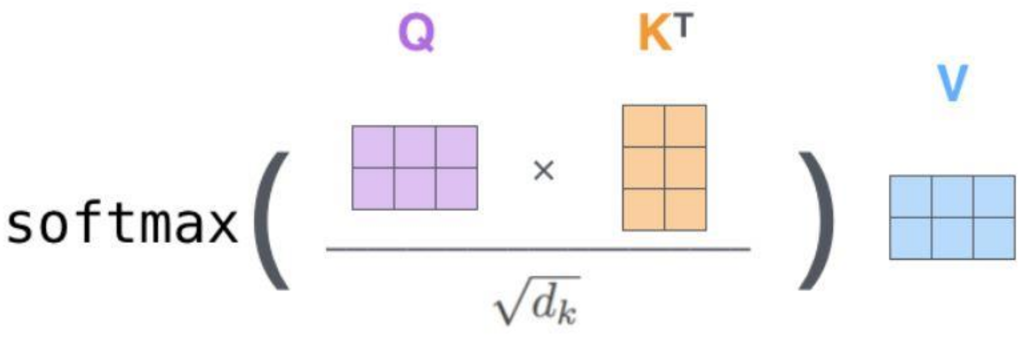
[2] Attention Is All You Need, Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin

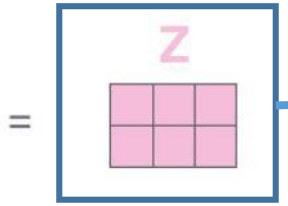
Self Attention

X → embedding of inputs packed

W^Q, W^K, W^V → Weight matrices that are trained







Attention output

Shows how much focus we should place on other parts of the sentence as we encode a word at a certain position

Transformer(2)

Helps determine position of each word, or distance between words in a seq

Positional encoding → vector added in the embedding

Multi-head attention → expands model's ability to focus on different positions

8 attentions heads → procedure similar as explained, different Q,K,V for each head

Bidirectional encoder → each word is encoded using previous and next context

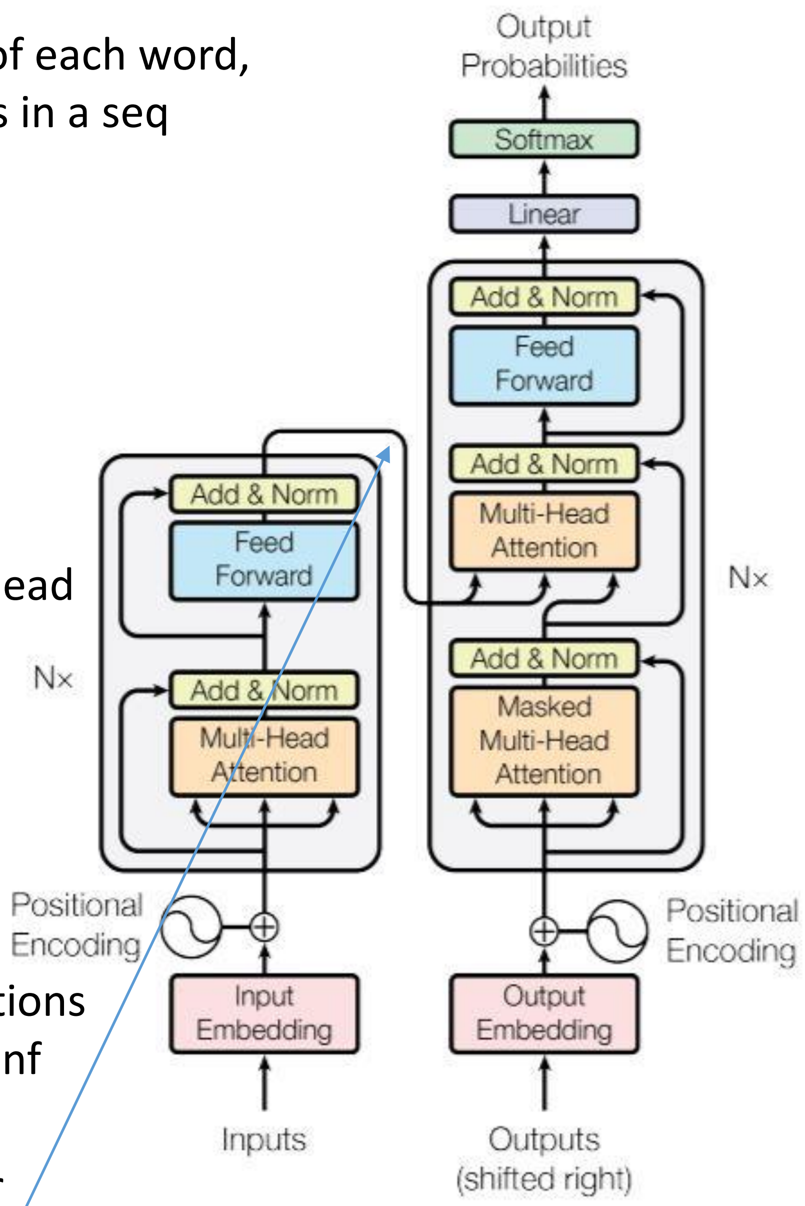
Decoder Side

- Similar components and structure

- *Differences*

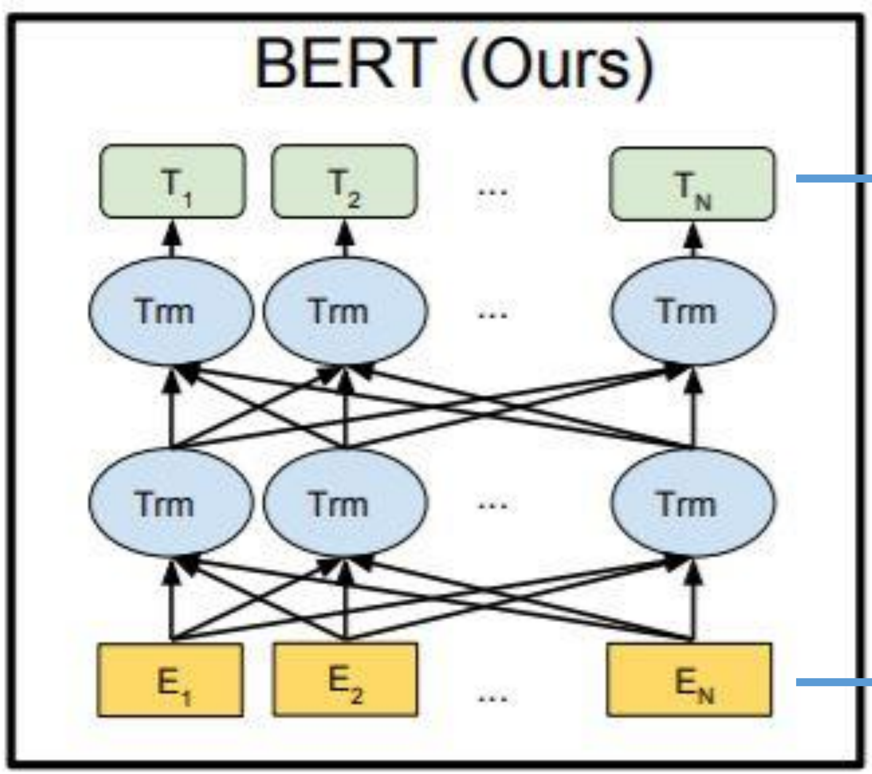
Self-attention allowed to "see" only earlier positions of the output → *masked* future positions with -inf

Keys and *Values* from the output of the encoder stack forwarded in the decoder's attention



BERT(2)

→ a Transformer Encoder stack



output

Intermediate representations

Embedding representation

Problem

Since bidirectional → would be possible for the words to "see itself" in a multilayer context

Trick → introduced the **masked language model**

Pre-Training → 2 novel unsupervised prediction tasks used

1. *Masked LM prediction*
15 % of words are masked

Fuses left and right context

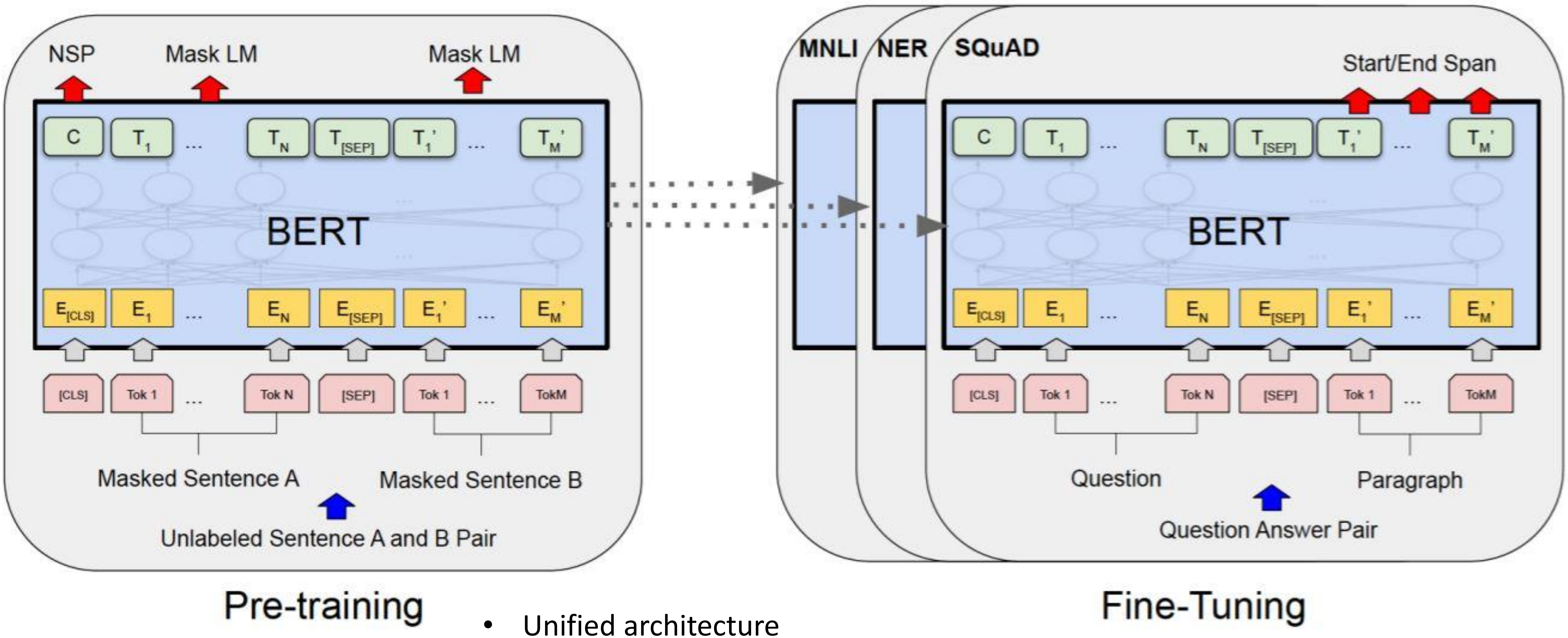
2. *Next sentence prediction*

Learns to handle relations between multiple sentences

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Devlin, Chang, Lee, Toutanova

BERT(3) – Fine Tuning

Pretrained on these 2 tasks → *all parameters* are fine-tuned using labeled data for the downstream tasks



- Unified architecture
- Same parameters for initialization
- Only output layer is task specific

DistilBERT - Paper

MegatronLM

BERT

Great progress in NLP

roBERTa

ELMO

Bigger Models → billions of parameters

Larger Datasets → GBs of text

Computational and memory requirements

Hard to adopt to production + deploy solutions on device

No energy efficient → GPU servers necessary → environmental cost

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, Sanh, Debut, Chaumond, Wolf

Goal

Reduce size of models, with retaining the performance

- quantization
- Weight pruning
- **Distillation**

<u>Knowledge distillation</u> → compression technique, where a small model(*student*) is trained to reproduce the behavior of a larger(*teacher*)one, or of an ensemble)

Knowledge Distillation

Exploit network's "dark knowledge" → We consider the teacher's full **output probability distribution**

Instead of training over **hard targets**(one hot encoding ground truth)

Train over the **soft targets**

$$p_i = \frac{\exp\left({z_i}/{T}\right)}{\sum j \exp\left({z_i}/{T}\right)}$$

Temperature-softmax introduced from Hinton → softens probabilities more

Distillation → similar to label smoothing, making model less overconfident

[4] Distilling the Knowledge in a Neural Network, Hinton, Vinyals and Dean

Training

Loss → distillation loss + supervised training loss

Masked language modelling loss + cosine embedding loss

Architecture of student

Identical with BERT

# of layers reduced by a factor of 2, token-type embeddings + pooler are removed → parameters are halved

Both have common hidden size

Initialization of weights of DistilBERT → from the teacher, taking one layer out of two

Improvements over BERT: Dynamic masking, gradient accumulation, w/o next sentence prediction

[5] RoBERTa: A Robustly Optimized BERT Pretraining Approach, Luke Zettlemoyer et al.

Experiments

General Language Understanding Evaluation
→ contains 9 tasks to eval NLU

Model's performance is compared at the **GLUE** benchmark

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Compared with ELMO and BERT-base(teacher) → better than ELMO, 97% of BERT performance - 40% fewer parameters

Size of model weights → ~200MB

Speed → 60% faster than BERT

Experiments(2)

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Comparable performance on 2 downstream tasks

IMDB sentiment classification, question answering

Ablation Study → wrt triple loss and weight initialization

Ablation	**Variation on GLUE macro-score**
$\emptyset$ - L_{cos} - L_{mlm}	-2.96
L_{ce} - $\emptyset$ - L_{mlm}	-1.46
L_{ce} - L_{cos} - $\emptyset$	-0.31
Triple loss + random weights initialization	-3.69

Masked Language loss has the smaller impact in performance

Conclusion

- A general purpose pre-training distillation rather than a task-specific one
- 40% smaller and 60% faster than BERT
- Retains the 97% of BERT’s performance on GLUE benchmark
- Outperforms ELMO on GLUE
- Tricks from roBERTa where used
- Comparable performance with BERT on downstream tasks
- Plausible for edge applications