World of Transformer

Karol Kaczmarek

Adam Mickiewicz University
Poznań

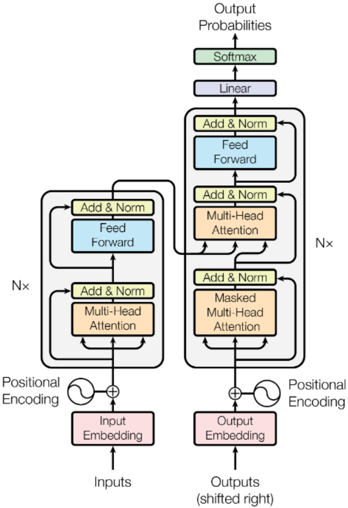
Applica.ai
Warsaw

2019

Transformer [1]

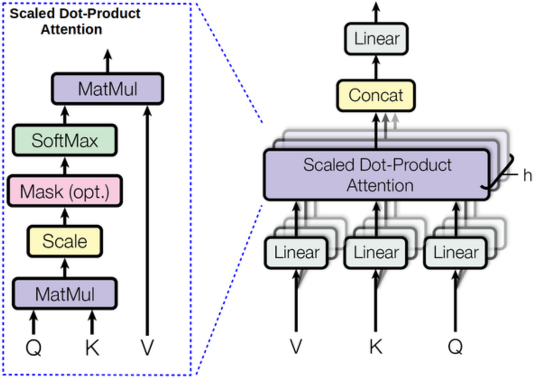
- June 2017
- encoder-decoder model
- dispensing with recurrence and convolutions entirely
- attention mechanism (MultiHeadAttention)
- positional encoding

Architecture



Multi-head attention

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$



GPT-2 [4]

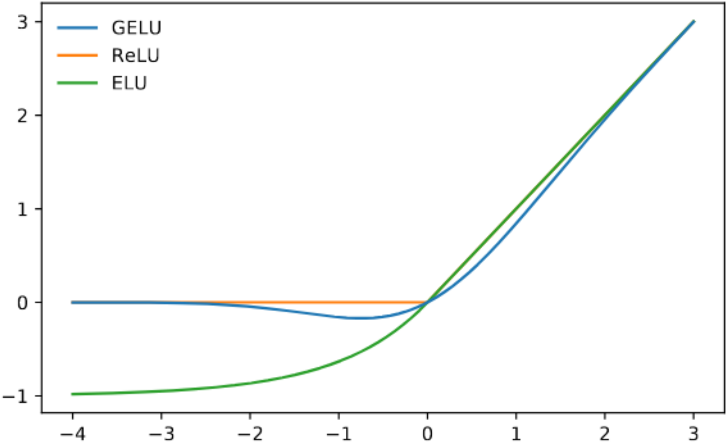
- February 2019
- based on Transformer [1] (GPT-1)
- Byte Pair Encoding (BPE) [2] on the byte level
  - $<$UNK$>$ occurs 26 times in 40 billion bytes
- use custom regex text splitter
- trained on 40GB of text collected from the Internet (WebText)
  - it's not Common Crawl (quality issues)
  - OpenWebText as an alternative
- GELU – Gaussian Error Linear Unit [3]

Models

$Name$	$Parameters$	$Layers$	d_{model}
Base	$117M$	12	768
Medium	$345M$	24	1024
Large	$762M$	36	1280
XL	$1542M$	48	1600

GELU – Gaussian Error Linear Unit [3]

GELU - $sigmoid(1.702 \cdot x) \cdot x$



Comparing different levels of BPE

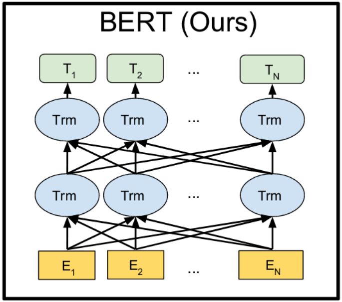
	BPE based on bytes	BPE based on characters
1	'I like cats.'	'I like cats.'
2	'I', ' like', 'cats', '.'	'I', 'like', 'cats', '.'
3	['0x49'], ['0x20', '0x6c', '0x69', '0x6b', '0x65'], ['0x20', '0x63', '0x61', '0x74', '0x73'], ['0x2e']	–
4	'I', 'Ġlike', 'Ġcats', '.'	–
5	'I', 'Ġli@@', 'ke', 'Ġca@@', 'ts', '.'	'I', 'li@@', 'ke' 'ca@@', 'ts', '.'

Comparing different levels of BPE

	BPE based on bytes	BPE based on characters
1	’Zażółć gęślą jaźń.’	’Zażółć gęślą jaźń.’
2	’Zażółć’, ’ gęślą’, ’ jaźń’, ’.’	’Zażółć’, ’gęślą’, ’jaźń’, ’.’
4	’ZaÅ¼Ã³ÅĤÄĩ’, ’ĠgÄĻÅĽlÄh’, ’ĠjaÅºÅH’, ’.’	–
5	’Z@@’ ’a@@’ ’Å@@’ ’¼@@” ’Ã³@@’ ’ÅĤ@@’ ’Äĩ’ ’Ġg@@’ ’Ä@@’ ’Ļ@@’ ’Å@@’ ’Ľ@@’ ’l@@’ ’Ä@@’ ’h’ ’Ġja@@’ ’Å@@’ ’º@@’ ’Å@@’ ’H’ ’.’	’Z@@’ ’a@@’ ’ż@@’ ’ó@@’ ’ł@@’ ’ć’ ’g@@’ ’ę@@’ ’ś@@’ ’l@@’ ’ą’ ’j@@’ ’a@@’ ’ź@@’ ’ń’ ’.’

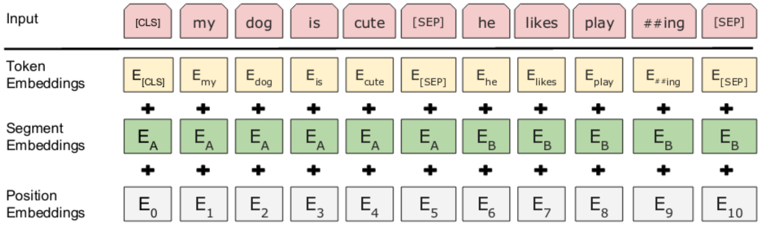
BERT [5]

- **BERT** – **B**idirectional **E**ncoder **R**epresentations from **T**ransformers = bidirectional Transformers



BERT input representation

- **[CLS]** – the special classification embedding
- **[SEP]** – the sentences separator, sentence pairs are packed together into a single sequence
- segment embeddings



Masked Language Model (MLM)

- masking some percentage of the input tokens at random
- predicting only those masked tokens (**[MASK]**)
- mask 15% of all
- masking procedure:
  - 80% of the time - replace the word with the **[MASK]** token
    - *my dog is hairy* $\rightarrow$ *my dog is [MASK]*
  - 10% of the time - replace the word with a random word
    - *my dog is hairy* $\rightarrow$ *my dog is apple*
  - 10% of the time - keep the word unchanged
    - *my dog is hairy* $\rightarrow$ *my dog is hairy*

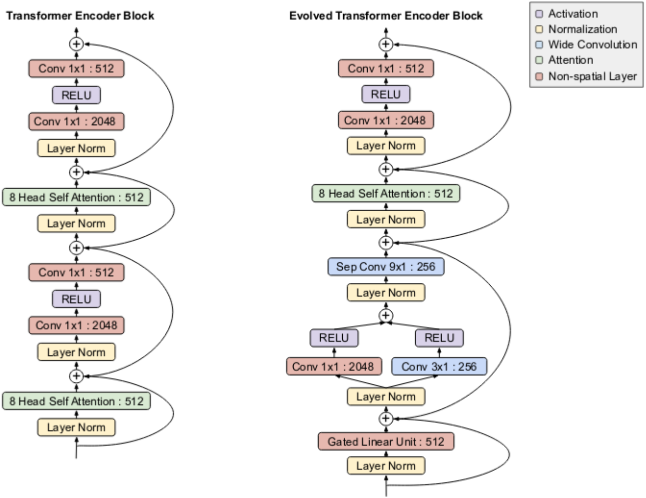
PyTorch

- PyTorch 1.2 support Transformer architecture:
  - nn.Transformer
  - nn.TransformerEncoder
  - nn.TransformerEncoderLayer
  - nn.TransformerDecoder
  - nn.TransformerDecoderLayer
  - nn.MultiheadAttention
- PyTorch 1.3 current version

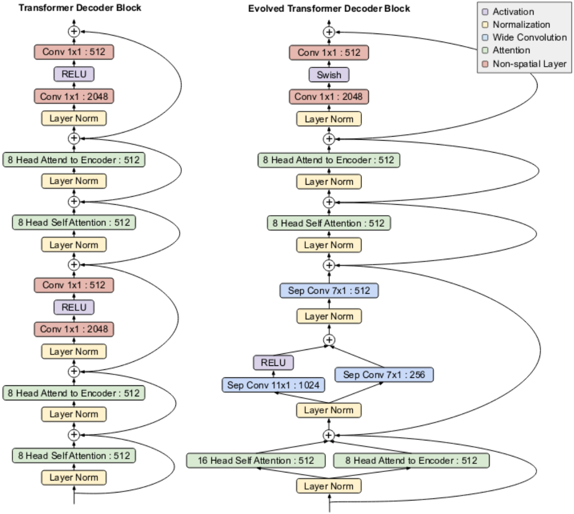
Evolved Transformer [7]

- January/February 2019
- $7,30 \cdot 10^{115}$ possible models
  - use fraction of data (WMT'14 En-De)
  - aggressive early stopping (allows models that are consistently performing well to train for more steps)
- Depth-wise separable convolutions (Xception[6])
- Gated Linear Units
- Swish activation

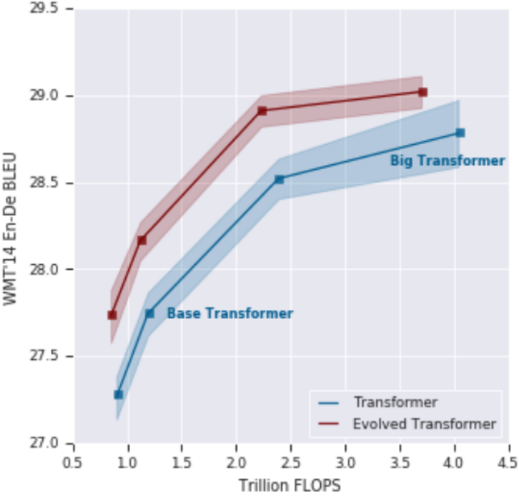
Evolved Transformer Encoder



Evolved Transformer Decoder



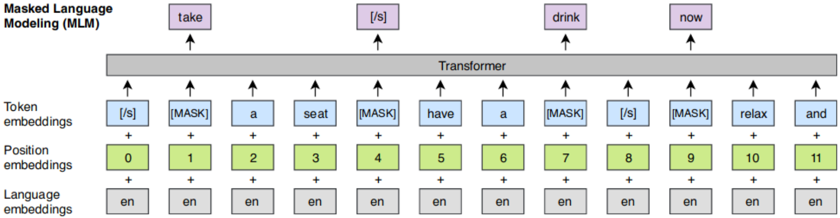
Score


29.5
29.0
28.5
28.0
27.5
27.0
WMT'14 En-De BLEU
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Trillion FLOPS
Big Transformer
Base Transformer
Transformer
Evolved Transformer

XLM [8]

- January 2019
- created by Facebook
- based on BERT
- use text streams of an arbitrary number of sentences instead of pairs of sentences
- no next sentence prediction
- 12 layers (BERT - 24 layers)

XLM


Masked Language Modeling (MLM)
take
[/s]
drink
now
Transformer
Token embeddings
[/s]
[MASK]
a
seat
[MASK]
have
a
[MASK]
[/s]
[MASK]
relax
and
Position embeddings
0
1
2
3
4
5
6
7
8
9
10
11
Language embeddings
en
en
en
en
en
en
en
en
en
en
en
en

XLM

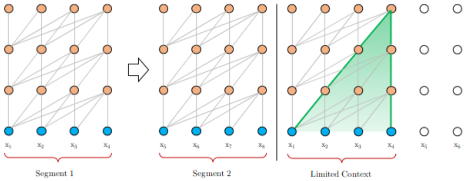
Model	Score	CoLA	SST2	MRPC	STS-B	QQP	MNLI_m	MNLI_mm	QNLI	RTE	WNLI	AX
BERT	80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
XLM	82.8	62.9	95.6	90.7/87.1	88.8/88.2	73.2/89.8	89.1	88.5	94.0	76.0	71.9	44.7

TransformerXL [9]

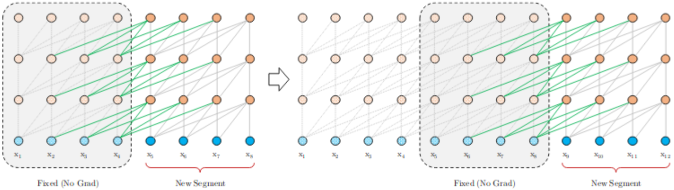
- June 2019
- Vanilla Transformer Language Models
  - limited by a fixed-length context
  - ignore all contextual information from previous segments (information never flow over segments)
- TransformerXL
  - "Recurrence" mechanism
  - Relative Positional Encoding

Score

Vanilla Transformer:



TransformerXL:



TransformerXL

- "Recurrence" mechanism
  - use fixed, cached segment (for each layer) from the previous segment
  - use stop-gradient for caching
  - different segments have the same positional encoding (an old segment is represented as [0, 1, 2, 3] and a new segment is processed as [0, 1, 2, 3, 0, 1, 2, 3] – for the two segments)
- Relative Positional Encoding
  - encode relative positional information in the cached segment
  - add content-dependent positional bias

Score

Method	enwiki8	text8	One Billion Word
Previous best	1.06	1.13	23.7
TransformerXL	0.99	1.08	21.8

Method	WikiText-103	PTB
Previous best	20.5	55.5
TransformerXL	18.4	54.5

Megatron-LM [10]

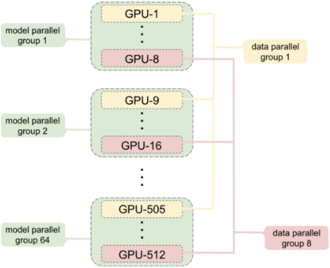
- September 2019
- created by Nvidia
- support BERT and GPT-2 models training with the memory optimization
- use Wikipedia (without Wikitext-103 articles), CC-stories, RealNews, OpenWebtext – 174 GB of text
- used 512 GPUs (Nvidia V100 32GB, trained over 9,2 days with 12 ZettaFLOPs)
- 480460 USD ($\sim$34 USD per hour – one DGX-2) to train the GPT-2 model

Parallelism

Speedup obtained for the 1.2 billion parameters model:

# of GPUs	1	2	4	8
Speedup	1.0	1.64	2.34	2.98

Hybrid Model and Data Parallelism:



Score

Model	Wikitext-103 Perplexity ↓	LAMBADA Accuracy ↑
355M	19,31	45,16
2,5B	12,76	61,73
8,3B	**10,81**	**66,51**
SOTA	16,43	63,24

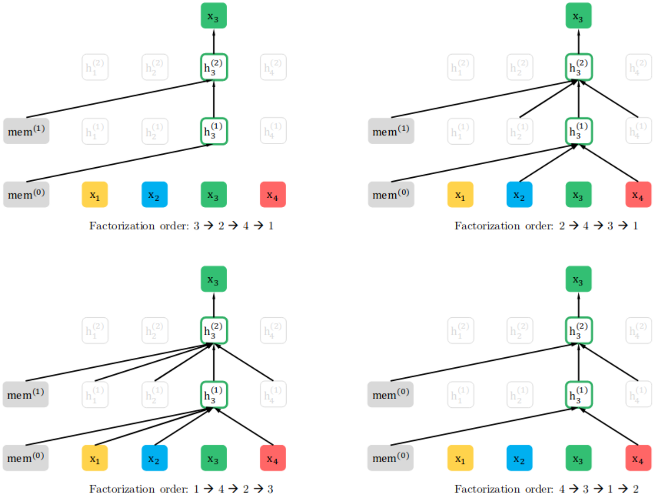
XLNet [11]

- June 2019
- BERT + TransformerXL
- Permutation Language Modelling
- 245000 USD to train the XLNet model (to beat BERT on NLP tasks)

autoregressive and autoencoding language modelling

- autoregressive (AR) language modelling (classical)
  - estimate the probability distribution of a text corpus
  - trained to encode a uni-directional context (either forward or backward)
  - is not effective at modelling deep bidirectional contexts
  - downstream tasks often require bidirectional context information
- autoencoding (AE) language modelling (like BERT)
  - aims to reconstruct the original data from corrupted input ([MASK] token – masked language model)
  - density estimation is not part of the objective

Permutation Language Modelling



BERT vs XLNet

- Sentence: [New, York, is, a, city]
- Select the two tokens: [New, York]
- Maximize: $log\ p($New York $|$ is a city$)$

Model	**First prediction**	**Second prediction**
BERT	$log\ p($New $|$ is a city$)$	$log\ p($York $|$ is a city$)$
XLNet	$log\ p($New $|$ is a city$)$	$log\ p($York $|$ **New**, is a city$)$

Score

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
XLNet	**89.8/-**	**93.9**	**91.8**	**83.8**	**95.6**	**89.2**	**63.6**	**91.8**	-
Single-task single models on test									
BERT	86.7/85.9	91.1	89.3	70.1	94.9	89.3	60.5	87.6	65.1
Multi-task ensembles on test (from leaderboard as of June 19, 2019)									
ALICE*	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8
XLNet*	**90.2/89.7**†	**98.6**†	90.3†	**86.3**	**96.8**†	**93.0**	67.8	**91.6**	**90.4**

RoBERTa – **R**obustly **o**ptimized **BERT** **a**pproach [12]

- July 2019
- created by Facebook
- based on BERT
- more data (over 160GB) + more steps + larger batches (8K)
- no next sentence prediction
- dynamic masking instead of static
- larger byte level byte pair encoding vocabulary (50K units)
- used 1024 GPUs (Nvidia V100 32GB, trained over one day)
- 104448 USD ($\sim$34 USD per hour – one DGX-2) to train the RoBERTa model

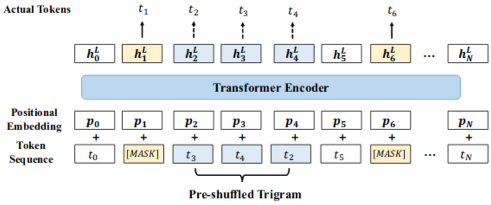
Score

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
$\text{BERT}_{\text{LARGE}}$	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
$\text{XLNet}_{\text{LARGE}}$	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	**90.2/90.2**	**94.7**	**92.2**	**86.6**	**96.4**	**90.9**	**68.0**	**92.4**	**91.3**	-
Ensembles on test (from leaderboard as of July 25, 2019)										
ALICE	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8	86.3
XLNet	90.2/89.8	98.6	90.3	86.3	**96.8**	**93.0**	67.8	91.6	**90.4**	88.4
RoBERTa	**90.8/90.2**	**98.9**	90.2	**88.2**	96.7	92.3	67.8	**92.2**	89.0	**88.5**

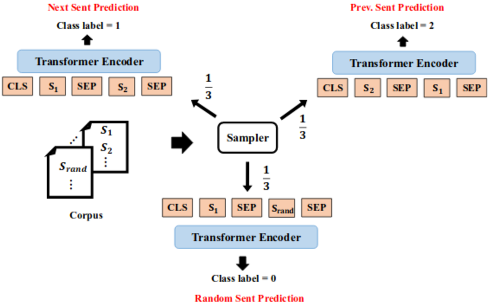
StructBERT (ALICE – Alibaba) [13]

- August 2019
- base on BERT
- Word Structural Objective – ability to reconstruct the right order of certain number of intentionally shuffled word token
- Sentence Structural Objective – extend the sentence prediction task by predicting both the next sentence and the previous sentence
- used 64 GPUs (Nvidia V100, trained over 38 hours/7 days)
- 8208-10336 USD ($\sim$27-34 USD per hour) to train base model and 36288-45696 USD to train huge model

Word Structural Objective



Sentence Structural Objective



Score (old)

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
+	2	王玮	ALICE large (Alibaba DAMO NLP)		82.9	61.6	95.2	91.1/87.7	89.6/88.6	74.0/90.4	87.9	87.4	95.4	80.9	65.1	40.7
+	3	Microsoft D365 AI & MSR AI	MT-DNNv2 (BigBird)		82.9	62.5	95.6	91.1/88.2	89.5/88.8	72.7/89.6	86.7	86.0	94.9	81.4	65.1	40.3
−	4	Jason Phang	BERT on STILTs		82.0	62.1	94.3	90.2/86.6	88.7/88.3	71.9/89.4	86.4	85.6	92.7	80.1	65.1	28.3
			GPT on STILTs		76.9	47.2	93.1	87.7/83.7	85.3/84.8	70.1/88.1	80.7	80.6	-	69.1	65.1	29.4
+	5	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hi		80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
	6	Neil Houlsby	BERT + Single-task Adapters		80.2	59.2	94.3	88.7/84.3	87.3/86.1	71.5/89.4	85.4	85.0	92.4	71.6	65.1	9.2
	7	Alec Radford	Singletask Pretrain Transformer		72.8	45.4	91.3	82.3/75.7	82.0/80.0	70.3/88.5	82.1	81.4	-	56.0	53.4	29.8
+	8	Samuel Bowman	BiLSTM+ELMo+Attn		70.5	36.0	90.4	84.9/77.9	75.1/73.3	64.8/84.7	76.4	76.1	-	56.8	65.1	26.5
	9	GLUE Baselines	BiLSTM+ELMo+Attn		70.0	33.6	90.4	84.4/78.0	74.2/72.3	63.1/84.3	74.1	74.5	79.8	58.9	65.1	21.7
			BiLSTM+ELMo		67.7	32.1	89.3	84.7/78.0	70.3/67.8	61.1/82.6	67.2	67.9	75.5	57.4	65.1	21.3

Score

System	CoLA 8.5k	SST-2 67k	MRPC 3.5k	STS-B 5.7k	QQP 363k	MNLI 392k	QNLI 108k	RTE 2.5k	WNLI 634	AX	Average
Human Baseline	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-	
BERT$_{\text{Large}}$	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7/85.9	92.7	70.1	65.1	39.6	80.5
BERT on STILTs	62.1	94.3	90.2/86.6	88.7/88.3	71.9/89.4	86.4/85.6	92.7	80.1	65.1	28.3	82.0
StructBERT$_{\text{Base}}$	57.2	94.7	89.9/86.1	88.5/87.6	72.0/89.6	85.5/84.6	92.6	76.9	65.1	39.0	80.9
StructBERT$_{\text{Large}}$	65.3	95.2	92.0/89.3	90.3/89.4	74.1/90.5	88.0/87.7	95.7	83.1	65.1	43.6	83.9
StructBERT$_{\text{Large}}$ ensemble	68.6	95.2	92.5/90.1	91.1/90.6	74.4/90.7	88.2/87.9	95.7	83.1	65.1	43.9	84.5
XLNet ensemble	67.8	96.8	93.0/90.7	91.6/91.1	74.2/90.3	90.2/89.8	98.6	86.3	90.4	47.5	88.4
RoBERTa ensemble	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8/90.2	98.9	88.2	89.0	48.7	88.5
Adv-RoBERTa ensemble	68.0	96.8	93.1/90.8	92.4/92.2	**74.8/90.3**	**91.1/90.7**	98.8	**88.7**	89.0	**50.1**	88.8
StructBERT$_{\text{RoBERTa}}$ ensemble	**69.2**	**97.1**	**93.6/91.5**	**92.8/92.4**	**74.4/90.7**	90.7/90.3	**99.2**	87.3	**89.7**	47.8	**89.0**

ALBERT – A Lite BERT [14]

- October 2019
- base on BERT
- factorized embedding parameterization (decomposing into two smaller matrices)
- use sentence-order prediction (SOP)
- cross-layer parameters sharing
  - share only attention parameters
  - share only FNN parameters
  - share attention and FNN parameters

NSP – next sentence prediction
SOP – sentence-order prediction

	Intrinsic Tasks			Downstream Tasks					
SP 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**

BERT vs ALBERT

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
	xlarge	1270M	24	2048	2048	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

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	17.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	3.8x
	xlarge	1270M	86.4/78.1	75.5/72.6	81.6	90.7	54.3	76.6	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	21.1x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	6.5x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	2.4x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	1.2x

Score

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

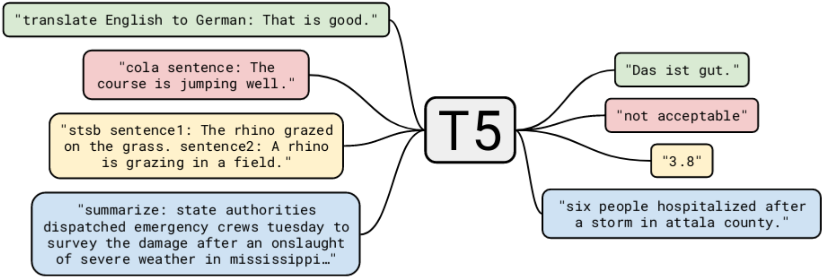
Other models

- TinyBERT [15]
  - September 2019
  - transfer the knowledge of a large teacher network to a small student network
  - 7,5 smaller, 9,4 faster, 28% parameters of BERT
- CTRL – Conditional Transformer Language [16]
  - September 2019
  - use 140GB of text from a wide variety of domains
  - large vocabulary of roughly 250k tokens
  - control codes (to generate task-specific data)
  - trained 2 weeks

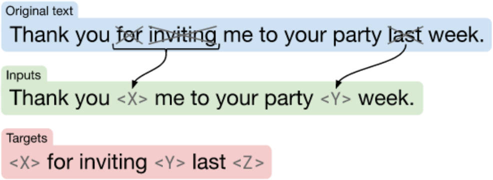
T5 – Text-to-Text Transfer Transformer [17]

- October 2019
- treat every NLP problem as a "text-to-text" problem (taking text as input and producing new text as output)
- based on Transformer (encoder and decoder)
- used "Colossal Clean Crawled Corpus" (called C4) – about 750 GB of text (this is only extracted text from April 2019)
- for fine-tuning use all of the task as a single task by concatenating all of the datasets (with the special processing into input and output form)
- trained on 1024 TPU v3 (TPU v2 costs $\sim$768 USD per hour)

Idea


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Pre-training



Score (GLUE)

	Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
	1	T5 Team - Google	T5		89.7	70.8	97.1	91.9/89.2	92.5/92.1	74.6/90.4	92.0	91.7	96.7	92.5	93.2	53.1
	2	ALBERT-Team Google Language	ALBERT (Ensemble)		89.4	69.1	97.1	93.4/91.2	92.5/92.0	74.2/90.5	91.3	91.0	99.2	89.2	91.8	50.2
+	3	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)		89.0	69.2	97.1	93.6/91.5	92.7/92.3	74.4/90.7	90.7	90.2	99.2	87.3	89.7	47.8
	4	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.8	68.0	96.8	93.1/90.8	92.4/92.2	74.8/90.3	91.1	90.7	98.8	88.7	89.0	50.1
	5	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0	48.7
	6	XLNet Team	XLNet-Large (ensemble)		88.4	67.8	96.8	93.0/90.7	91.6/91.1	74.2/90.3	90.2	89.8	98.6	86.3	90.4	47.5
+	7	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
	8	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

	ALBERT	T5-Small	T5-Base	T5-Targe	T5-3B	T5-11B
Score	89,4	77,4	82,7	86,4	88,5	89,7

Score (SuperGLUE)

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-g	AX-b
1	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	99.3/99.7	76.6
2	T5 Team - Google	T5		88.9	91.0	93.0/96.4	94.8	88.2/62.3	93.3/92.5	92.5	76.1	93.8	92.7/91.9	65.6
3	Facebook AI	RoBERTa		84.6	87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	91.0/78.1	57.9
4	IBM Research AI	BERT-mtl		73.5	84.8	89.6/94.0	73.8	73.2/30.5	74.6/74.0	84.1	66.2	61.0	97.8/57.3	29.6
5	SuperGLUE Baselines	BERT++		71.5	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	79.0	69.6	64.4	99.4/51.4	38.0
		BERT		69.0	77.4	75.7/83.6	70.6	70.0/24.1	72.0/71.3	71.7	69.6	64.4	97.8/51.7	23.0
		Most Frequent Class		47.1	62.3	21.7/48.4	50.0	61.1/0.3	33.4/32.5	50.3	50.0	65.1	100.0/50.0	0.0
		CBoW		44.5	62.2	49.0/71.2	51.6	0.0/0.5	14.0/13.6	49.7	53.1	65.1	100.0/50.0	-0.4
		Outside Best		-	80.4	-	84.4	70.4/24.5	74.8/73.0	82.7	-	-	-	-
-	Stanford Hazy Research	Snorkel [SuperGLUE v1.9]		-	-	88.6/93.2	76.2	76.4/36.3	-	78.9	72.1	72.6	-	47.6

References I

[1] A. Vaswani and et al., “Attention is all you need,” 2017.

[2] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” 2016.

[3] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” 2016.

[4] A. Radford, J. Wu, and et al, “Language models are unsupervised multitask learners,” 2019.

[5] J. Devlin and et al., “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.

[6] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2016.

[7] D. So, C. Liang, and Q. Le, “The evolved transformer,” 2019.

[8] A. C. Guillaume Lample, “Cross-lingual language model pretraining,” 2019.

[9] Z. Dai, Z. Yang, and et al., “Transformer-XL: Language modeling with longer-term dependency,” 2019.

[10] M. Shoeybi, M. Patwary, R. Puri, and et al., “Megatron-lm: Training multi-billion parameter language models using model parallelism,” 2019.

[11] Z. Yang, Z. Dai, and et al., “Xlnet: Generalized autoregressive pretraining for language understanding,” 2019.

[12] Y. Liu and et al., “Roberta: A robustly optimized bert pretraining approach,” 2019.

[13] W. Wang and et al., “Structbert: Incorporating language structures into pre-training for deep language understanding,” 2019.

[14] Z. Lan and et al., “Albert: A lite bert for self-supervised learning of language representations,” 2019.

[15] X. Jiao and et al., “Tinybert: Distilling bert for natural language understanding,” 2019.

[16] N. S. Keskar and et al., “Ctrl - a conditional transformer language model for controllable generation,” 2019.

[17] C. Raffel and et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2019.

References II