NLP AT SCALE

ANDREW UTKIN

Plan for today

- Old models
  - GPT-1
  - GPT-2
  - Bert
- Main models
  - GPT-3
  - T5
- Learning approaches
  - Zero-shot, one-shot, few-shot learnings
  - Prompt-engineering
  - Prompt-tuning

GPT-1(OpenAI GPT Model)

- Presented in 2018
- Based on transformer-decoder architecture
- 110M trainable parameters

Transformer Decoder

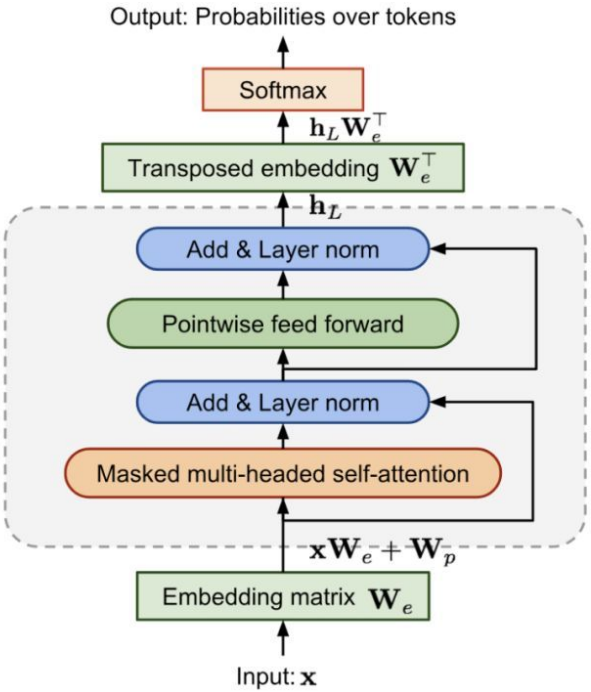
- Embeddings for input
- Positional Embeddings

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

- Layer Normalization
- Feed Forward Layer

$$\mathrm{FFN}(x) = \mathrm{act}\,(xW_1 + b_1)W_2 + b_2,$$



Training stages

1. Unsupervised pre-training

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$ - maximize

2. Supervised fine-tuning

$$P(y|x^1, \ldots, x^m) = \texttt{softmax}(h_l^m W_y).$$

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m).$$ - maximize

GPT-1 specification

- Training Dataset - BooksCorpus (7000 unpublished books)
- embedding size = 768
- 12 decoders blocks
- 12 heads of attention
- 3072 dimensional inner states
- Optimizer: Adam
- Trained for 100 epochs with minibatches of 64 randomly sampled sequences of 512 tokens

Abilities

1. Question Answering
2. Semantic Similarity
3. Textual entailment

GPT-2

- Presented in 2019
- Successor of GPT-1
- It was a response for Bert
- 1.5B parameters

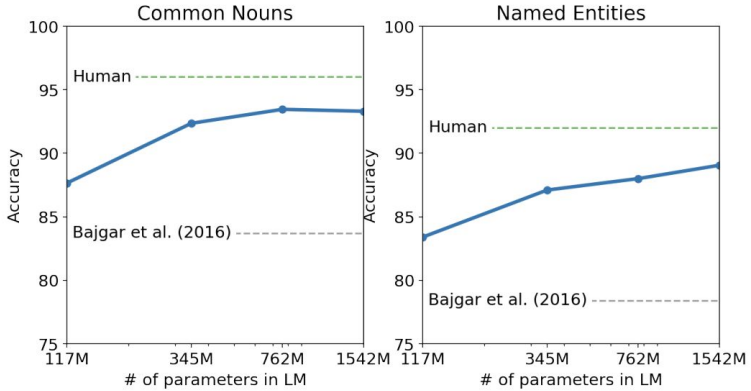
What's new in GPT2?

- byte-level BPE
- vocabulary size increased to 50k
- add one more layer normalization in the beginning
- batch size increased to 512
- context size increased to 1024
- to training dataset added text from 8 millions of sites (≈ 40 GB)

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Zero-shot learning

This is an approach in machine learning that allows the model to correctly perform tasks for which it has not been trained, without giving any examples



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:      ← task description
2  cheese =>                         ← prompt
```

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

GPT-3

- Presented in 2020 with the words “A model, which will cope with any task in English language”
- Successor of GPT-2
- 175B params

What's new in GPT-3

- change attention similar to Sparse Transformer
- increased to 175B params
- context increased to 2048 tokens
- New Dataset was collected (≈ 570 GB)

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

New two approaches

- One-shot learning

  This is an approach in machine learning that allows the model to correctly perform tasks for which it has not been trained, but giving one example

- Few-shot learning

  This is an approach in machine learning that allows the model to correctly perform tasks for which it has not been trained, but giving K examples

Example of approaches

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

Prompt-engineering

Prompt engineering involves crafting and inputting a carefully designed text “prompt” into a Large Language Model (LLM). This prompt essentially guides the model's response, steering it toward the desired output style, tone, or content.

Example 1: Content Creation

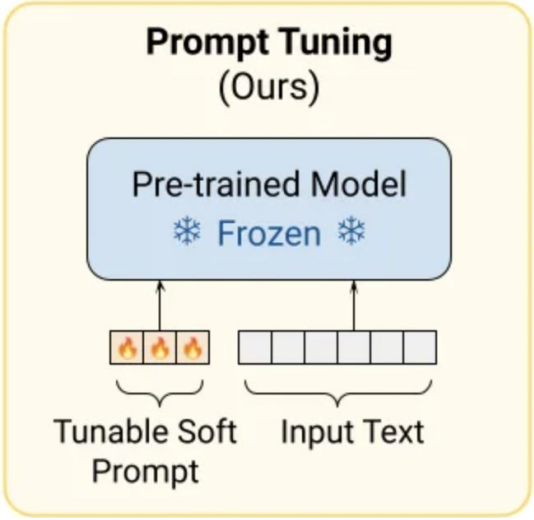
- **Original Prompt:** “Write a story about a dragon.”
- **Tuned Prompt:** “Write a humorous story about a friendly dragon who loves baking cookies and lives in a magical forest.”

Example 5: Recipe Generation

- **Original Prompt:** “Give me a chicken recipe.”
- **Tuned Prompt:** “Provide a healthy grilled chicken recipe suitable for a ketogenic diet, including ingredients like fresh herbs and olive oil, and avoiding sugars and carbs.”

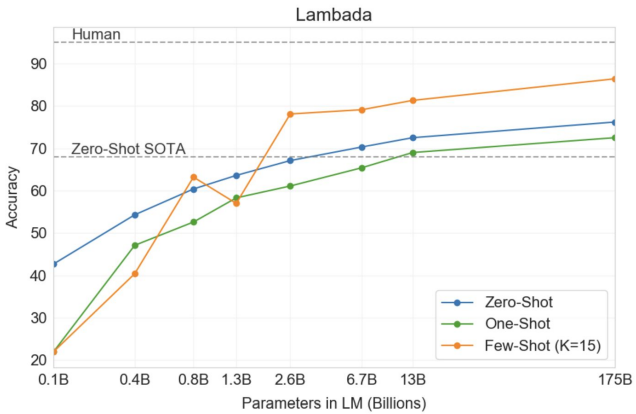
Prompt-tuning

The technique of configuring large language models, such as GPT-3 or other transformer-based models, without changing their basic weights. Soft prompts(embeddings) are added to input. During the learning process, these embeddings are adjusted (optimized) in conjunction with the task that the model is aimed at

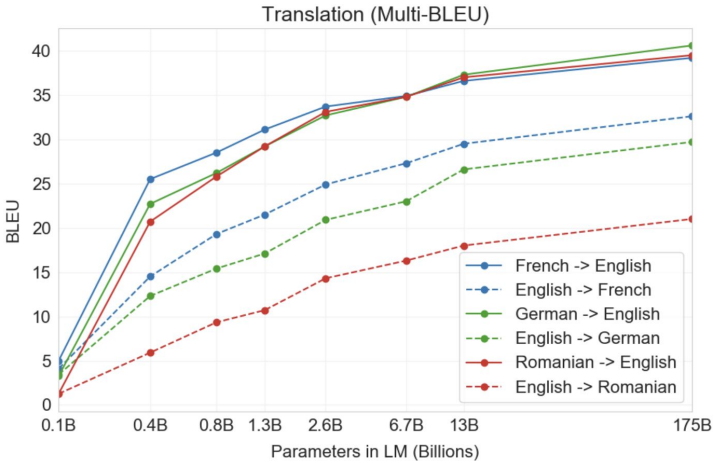


Result of approaches

Setting	LAMBADA (acc)	LAMBADA (ppl)
SOTA	68.0[a]	8.63[b]
GPT-3 Zero-Shot	**76.2**	**3.00**
GPT-3 One-Shot	**72.5**	**3.35**
GPT-3 Few-Shot	**86.4**	**1.92**

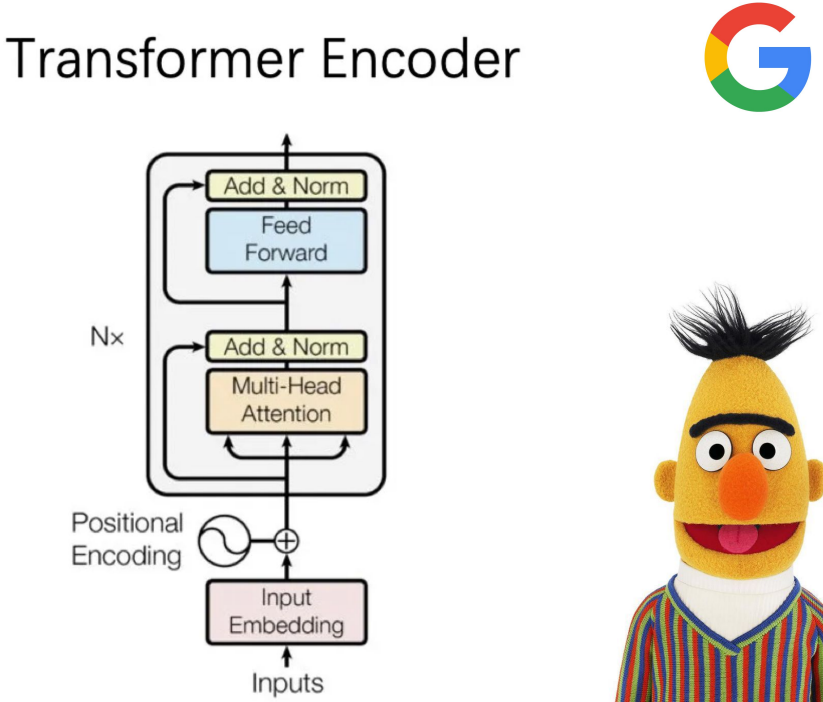


Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5



Bidirectional Encoder Representations from Transformers

- presented in 2018
- 110m trainable params in base version and 340m params in large version
- goal was to make model similar to GPT-1
- Based on transformer-encoder architecture



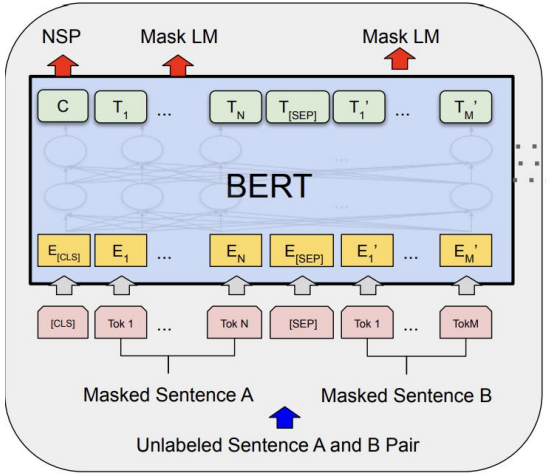
Bert pre-training proccess

1. Masked Language Model (NLM)

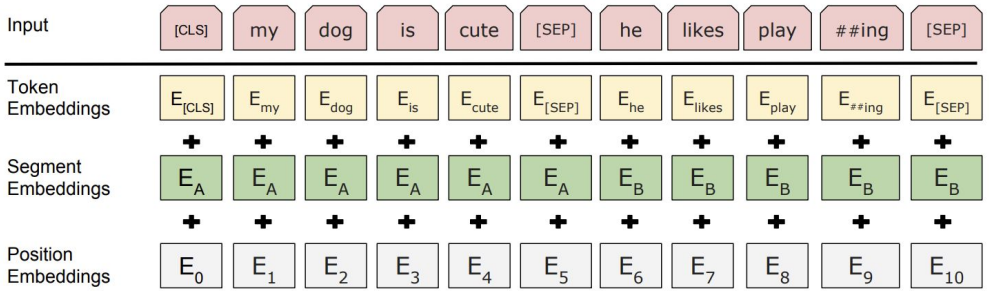
   Tokens are randomly selected from the text to be masked with a special token ([MASK]), and the goal of BERT is to correctly predict these masked tokens

2. Next Sentence Prediction(NSP)

   The model receives two sentences (A and B) as input, and its task is to determine whether the second sentence is a logical continuation of the first.



Pre-training

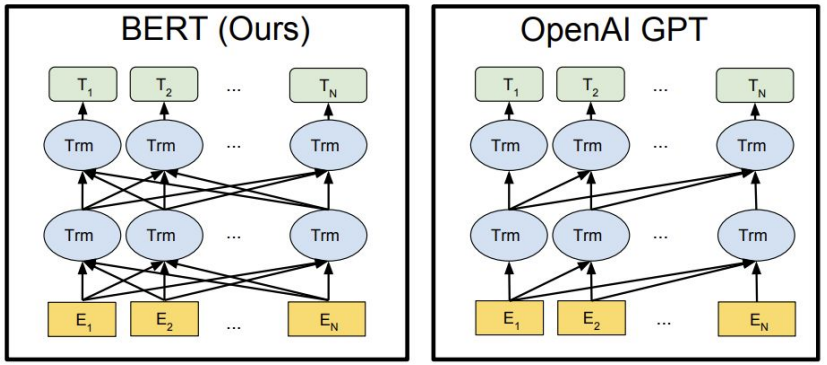


Bert specification

- Bert Base
  - 12 blocks of Transformer-encoder
  - 768 embeddings size
  - 12 heads of attention
  - 110m of trainable params
- Bert Large
  - 24 blocks of Transformer-encoder
  - 1024 embeddings size
  - 16 heads of attention
  - 340m of trainable params
- WordPiece tokenizer (30k vocabulary size)
- Training Dataset: BooksCorpus, like in GPT-1,(800m words) + English Wikipedia(2,500m words)
- batch size 256 and context size 512 tokens
- Adam optimizer
- 40 epochs (4 days on 16(64) TPUs)

Compare GPT и Bert

Bert	GPT-1
Training Dataset = 3.3B words	Training Dataset = 800M words
1 batch has 128k words	1 batch has 32k words
Lr depends on fine-tuning task	Same lr for each fine-tuning task



System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

T5(Text-to-Text Transfer Transformer)

- Presented in 2019
- Based on full transformer architecture
- The largest model has 11B trainable params

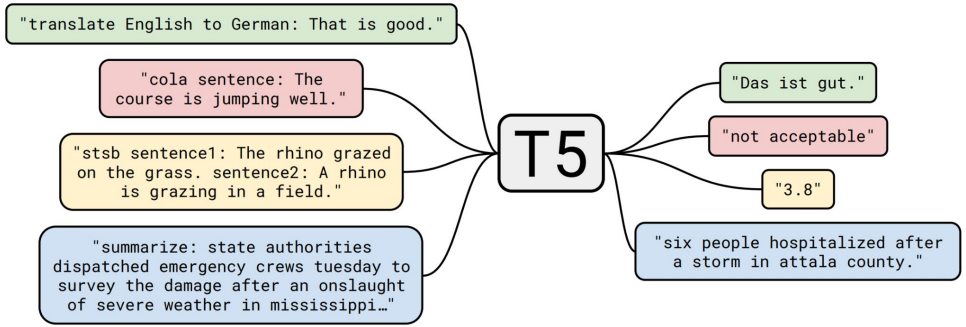
Training process

1. Pre-training process

   Span-based-denoising autoencoder. (Extracting random subsequences (spans) from texts and replacing them with special tokens, after which the model is trained to predict the original text of these subsequences.)

2. Fine-tuning for specific task:

   Adding the “task type” to the beginning of the input



T5 specification

- Vocabulary size 32k
- 24 decoder and 24 encoder blocks
- 16 heads of attention
- 1024 embeddings size
- batch size 128 with context size 512 tokens
- Training Dataset: Colossal Clean Crawled Corpus

It is well suited for tasks related to machine translation, text summarization, answers to questions

Compare GPT-3 and T5

	GPT3	**TP**
Плюсы	Мощная генерация текста Гибкое применение с помощью промтов	Универсальность Структурированные ответы
Минусы	Иногда непредсказуемые ответы	Сложность формулировки
Сценарии применения	Генерация разнообразного контента, чат-бот	Суммаризация текста, машинный перевод, ответы на вопросы

Sources

- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. OpenAI.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May 24). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.
- Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020, July 22). Language Models are Few-Shot Learners. arXiv:2005.14165.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2019, September 23). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683.
- Lester, B., Al-Rfou, R., & Constant, N. (2021, September 2). The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691.