Transformers

Tu4n
@AIST
@VJAI

Agenda

1. Why we’re here ?
2. The evolution of Transformers
3. A deep look to Transformers architecture
4. Style Transformers yourself
5. Bring Transformers to our life

Why we're here?

Why Transformers?

- Encoder-Decoder architecture
- Seq2Seq[1] -> Transformers[2]
- NMT, NLU, Text generator ...
- Transfer learning and Pre-trained language model

[1] Sequence to Sequence Learning with Neural Networks (https://arxiv.org/abs/1409.3215)
[2] Attention is All you need (https://arxiv.org/pdf/1706.03762.pdf)

The evolution of Transformers

- From seq2seq to Transformers

  http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

- Transformers derivatives: BERT, GPT, (T5)

  https://github.com/thunlp/PLMpapers

- Transformers ++: Transformer-XL, AlBert, …

A deep look to Transformers architecture

- Illustrated Transformers (http://jalammar.github.io/illustrated-transformer/)
- Examples:
  - BERT
  - GPT
  - ALBERT
  - Transformer XL

Style Transformers yourself

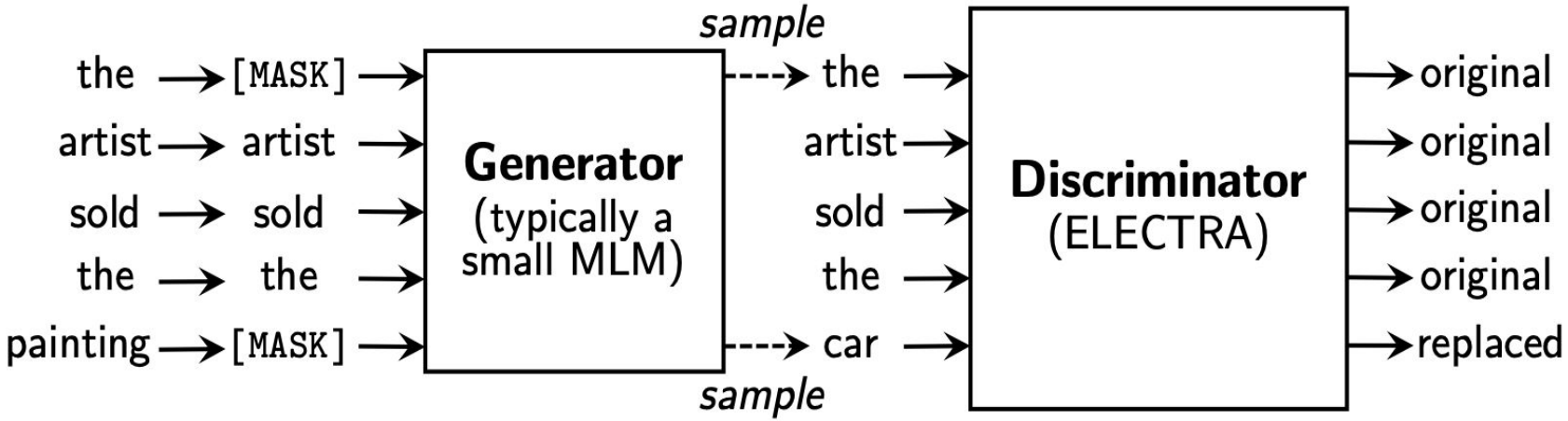
- Stack or recurrent[1]
- Shared parameters or not[1]
- Relative positional encodings[2]
- Objectives
  - Language model
  - Denoising[3]
  - Permutation
  - Discriminator[4]

[1] ALBERT (https://arxiv.org/abs/1909.11942)
[2] Transformer-XL (https://arxiv.org/abs/1901.02860)
[3] XLNet (https://arxiv.org/abs/1906.08237)
[4] ELECTRA (https://openreview.net/forum?id=r1xMH1BtvB)

Objective functions

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
I.i.d. noise, mask tokens	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

Objective functions



Style Transformers yourself

- Self-Attention ++
  - Adaptive Attention Span[1]
  - Hasing Attention[2]
  - Reversible Trans[2]
- Caching[3]
- Encoder (+ Decoder)
- The ambition of T5[4]

[1] Adaptive Attention Span in Transformers (https://arxiv.org/pdf/1905.07799.pdf)
[2] Reformer (https://arxiv.org/pdf/2001.04451.pdf)
[3] Transformer-XL (https://arxiv.org/abs/1901.02860)
[4] **T**ext-**t**o-**T**ext **T**ransfer **T**ransformer (https://arxiv.org/pdf/1910.10683.pdf)

Bring Transformers to our life

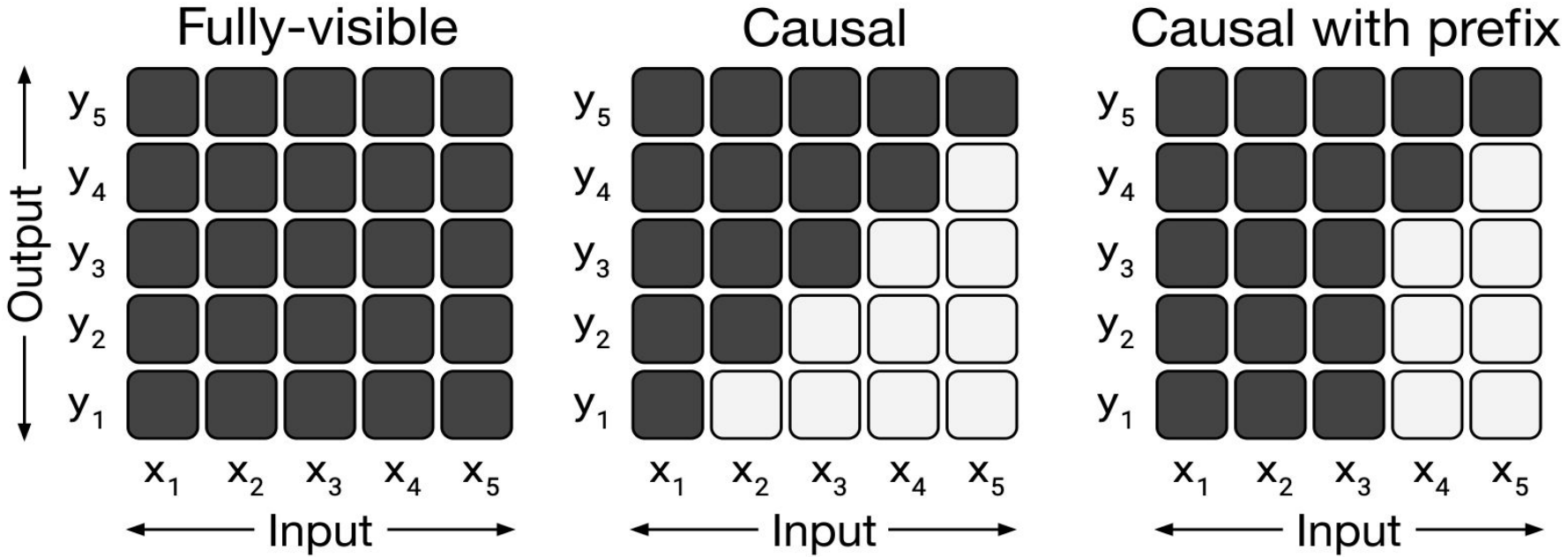
- Train Transformers from the scratch
- Fine-tuning
- Compress / Distillation

Train Transformers from the scratch

- Build the vocabulary
  - Word dictionary ordered by appearance frequency
  - BPE, using SentencePiece[1]
  - Merge vocab: Use adaptive embeddings[2]
- Prepare the inputs
  - Separate with training*
  - Group samples by length*
  - Understand the attention masks
  - Pair order

[1] https://github.com/google/sentencepiece
[2] Adaptive input representations for NLM (https://arxiv.org/pdf/1809.10853.pdf)
* from speaker with love

Understand the attention masks

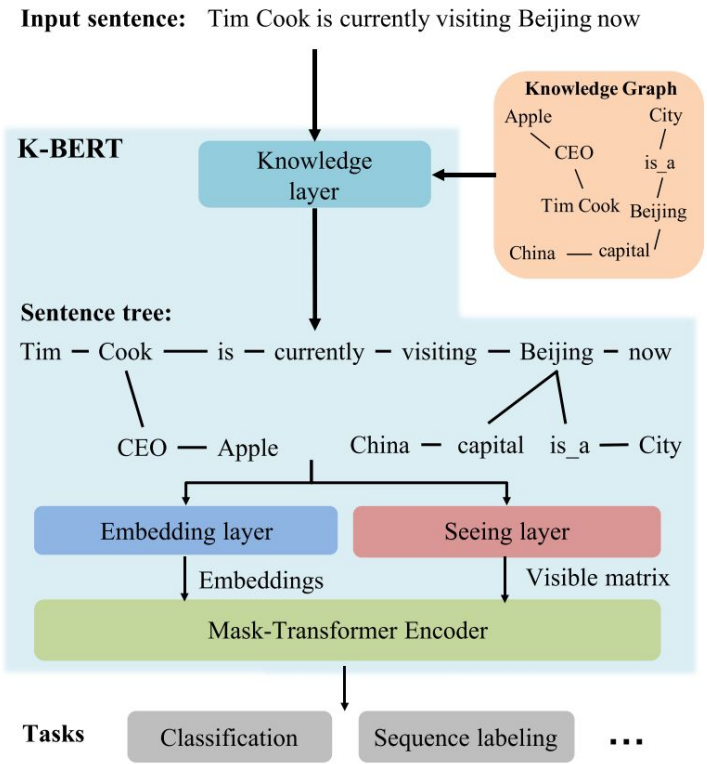


Train Transformers from the scratch

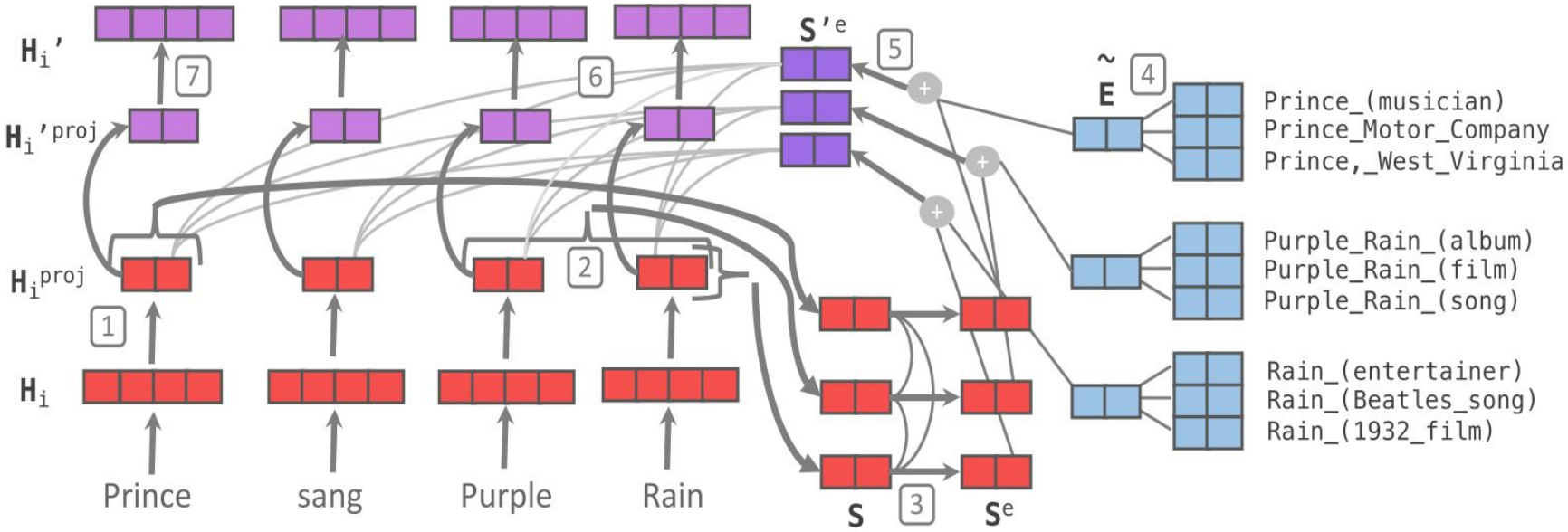
- Training
  - Bigger, longer is better[1][2]
  - Distributed Data Parallel[3]
  - Training with large batchsize[4]
  - Using NVIDIA Mixed Precision[5]
  - Using mean training losses*
- Using external KB:
  - Using additional text information[6]
  - Using contextual representations[7]

[1][2] RoBERTa (https://arxiv.org/pdf/1907.11692.pdf), T5 (https://arxiv.org/pdf/1910.10683.pdf)
[3] http://www.telesens.co/2019/04/04/distributed-data-parallel-training-using-pytorch-on-aws/
[4] LAMB (https://arxiv.org/pdf/1904.00962.pdf)
[5] https://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/index.html
[6] K-Bert (https://arxiv.org/abs/1909.07606)
[7] Knowledge enhanced contextual word representations (https://arxiv.org/abs/1909.04164)
* from speaker with love

K-Bert


Input sentence: Tim Cook is currently visiting Beijing now
K-BERT
Knowledge layer
Knowledge Graph
Apple
CEO
Tim Cook
City
is_a
Beijing
China
capital
Sentence tree:
Tim Cook is currently visiting Beijing now
CEO Apple
China capital is_a City
Embedding layer
Seeing layer
Embeddings
Visible matrix
Mask-Transformer Encoder
Tasks
Classification
Sequence labeling
...

Knowledge enhanced contextual word representations



Using sentencepiece to build vocab

Using tokenizer

Using Nvidia apex amp (fp16)

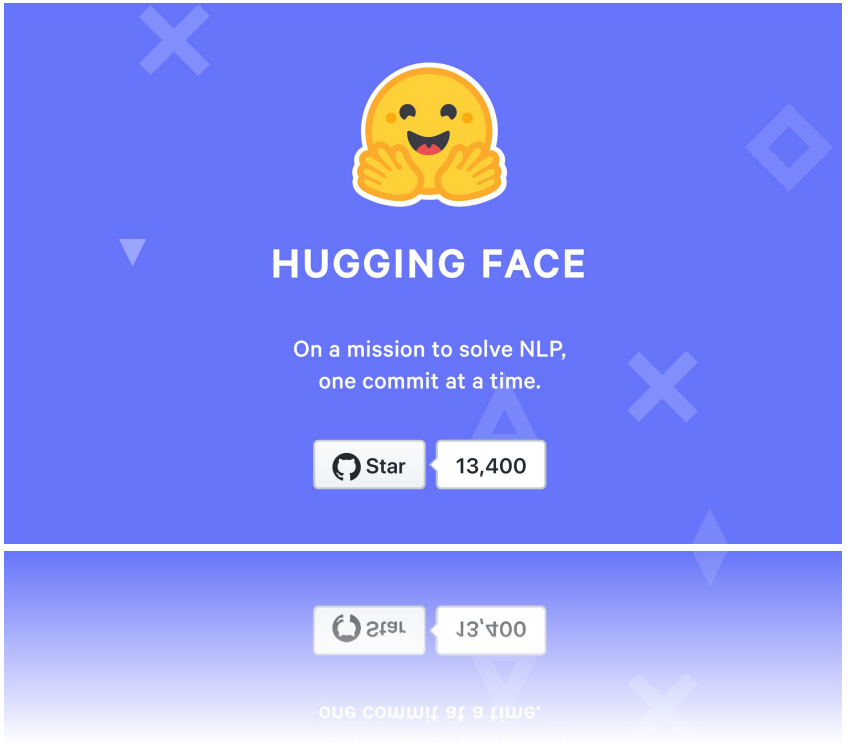
Fine-tuning

Papers to read:

- Universal Language Model Fine-tuning for Text Classification (https://arxiv.org/pdf/1801.06146.pdf)
- How to Fine-Tune BERT for Text Classification? (https://arxiv.org/abs/1905.05583)
- To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks (https://arxiv.org/abs/1903.05987)
- To Tune or Not to Tune? How about the best of both worlds? (https://arxiv.org/abs/1907.05338)
- T5 (https://arxiv.org/pdf/1910.10683.pdf)

Fine-tuning

- Sequence classification / regression / multi-label classification
- Tokens classification
- QA
- Masked LM / LM (used for training)



Call Bert Model

Getting Bert output to further use

Compress / Distillation

- Pruning
- Weight Factorization
- Knowledge Distillation
- Weight Sharing
- Quantization

http://mitchgordon.me/machine/learning/2019/11/18/all-the-ways-to-compress-BERT.html