M | logits.ai Sign in Get started

XLNet — SOTA pre-training method that outperforms BERT

XLNet: Generalized Autoregressive Pretraining for Language Understanding

Ankur Bohra in logits.ai Follow
Jun 25 · 5 min read ★

NLP Research is growing fast, and in less than nine months, we have XLNet, a new state of the art pre-training method that outperforms BERT[1] in more than 20 tasks. XLNet was proposed by the researchers from Carnegie Mellon University and Google Brain team, the same team behind the Transformer-XL paper. XLNet outperforms BERT on 20 tasks and achieves state-of-the-art results on 18 tasks, including question answering, natural language inference, sentiment analysis, and document ranking.

What is pre-training in machine learning?

Pre-training is training a model on another task before training it on the actual task. E.g., let's say you want to do sentiment analysis, you start by training the model for language modeling first.

Why do you need pre-training?

Shortage of human-labeled training data for a given task is one of the biggest challenges in machine learning. So the idea here is to pre-train a model on a large text corpus like Wikipedia on some other task which doesn't require human-labeled data. This pre-trained machine learning model learns to capture information from the task, e.g. in the case of NLP, the model learns to understand language. We then reuse the same model and fine-tune the model for the actual task like sentiment analysis using the small labeled dataset we have.

What is a transformer based model?

Transformer-based models are the models which are based on the architecture Transformer model. Transformer model used the concept of attention instead of the previously prevalent recurrent model for natural language processing.

What is an autoencoding language model?

Autoencoding model is a model that aims to reconstruct the original data from corrupted input.

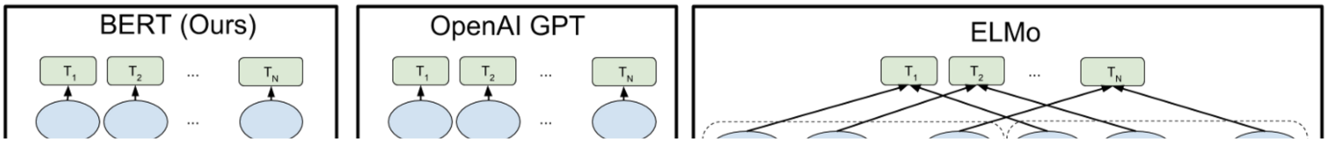
What is an autoregressive language model?

Autoregressive model is a unidirectional model which aims to predict data from past input.
Before we try to understand XLNet, let's take a quick look at the BERT (autoencoding) model.

. . .

BERT — Bidirectional Encoder Representations from Transformers.

BERT comes under the category of autoencoding (AE) language model.
BERT is the first transformer based model to bring deep bi-directional context, unlike OpenAI GPT and Transformer-XL.



BERT is deeply bidirectional, OpenAI GPT is unidirectional, and ELMo is shallowly bidirectional.

What is a bi-directional context?
To get a context of a word, instead of using just the previous or the next words, we use both the previous and the next words to understand the context.
Let's take an example of the word **bank**. Bank can have different meaning depending on the context of the sentence. E.g., "Bank account" and "Bank of the river." Now, let's say you want to find the context of the word bank in the sentence "I accessed my bank account." The bi-directional context would consider the words from both the sides "I accessed my" and "account." In the uni-directional context, you could only look from one side.

Training Objective

BERT is trained with two objectives

1. **Masked Language Model**

   15% of the words are randomly chosen which are then masked. The model is then trained to predict the words which were masked.

2. **Next Sentence Prediction**

   BERT can take two input sequences as input and predict if the second sentence is following the first.

Advantages

1. **Bi-directional context**

   No model before BERT was able to pre-train the model with deep bi-directional context. In a traditional unidirectional model — which uses previous words to predict the next word — If you try to apply the same idea in both directions, you would indirectly end up exposing the word to the model.

Limitations

1. **Pretrain-finetune-discrepancy**

   During pre-training, BERT masks a certain portion of the input sequence and then tries to predict the masked token. But when you are fine-tuning the model for a particular task, you never mask the input tokens, this generates a pretrain-finetune-discrepancy.

2. **Assumes the predicted tokens are independent**

   As the predicted tokens are masked in the input, BERT is not able to model the joint probability. In other words, BERT assumes the predicted tokens are independent of each other given the unmasked tokens.

3. **Fixed length context**

   BERT model cannot handle more than 512 input tokens.

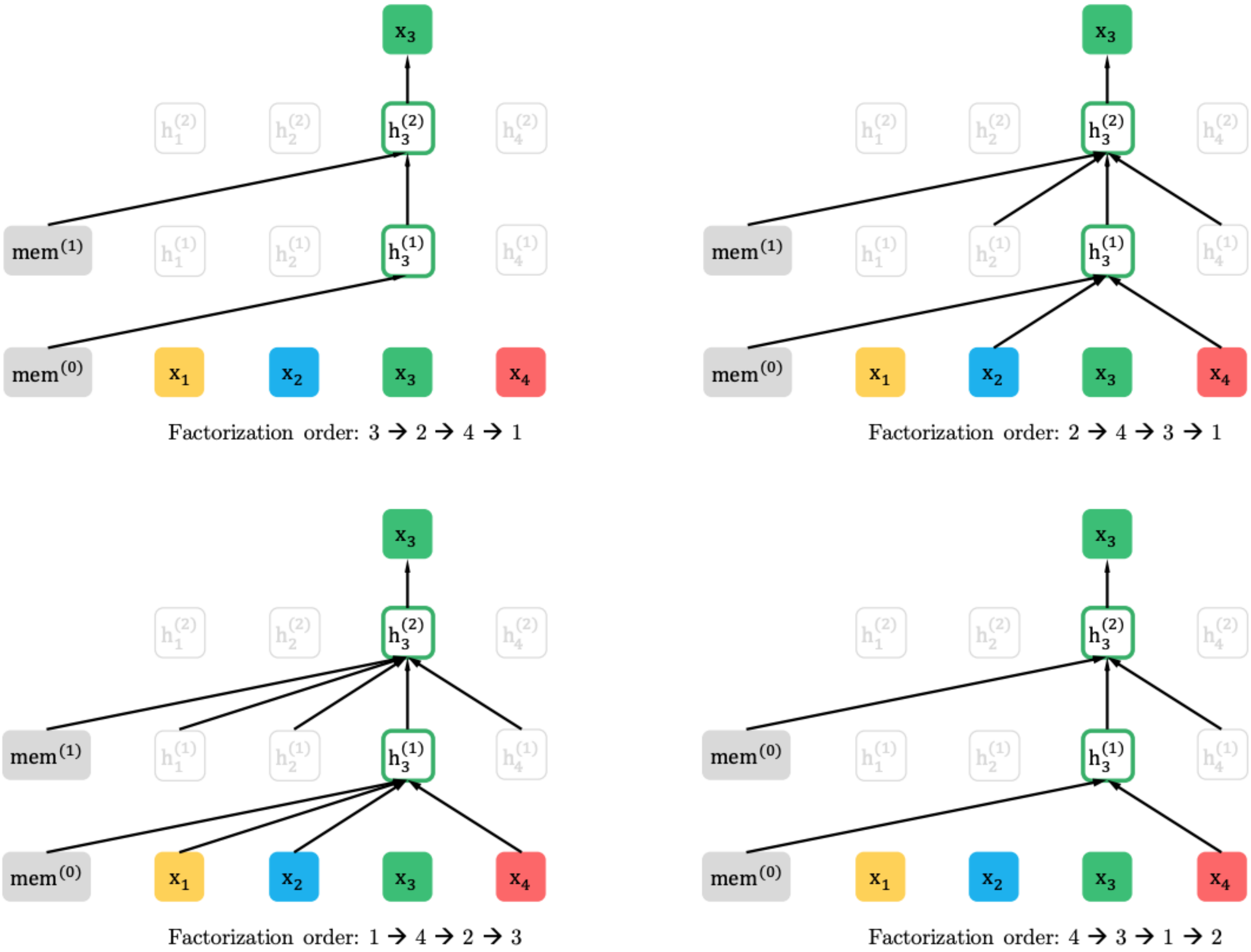
4. **Not good at sequential text generation**

   During pre-training, as the BERT's objective is not to predict the next word from the previous words, it is not good at sequential text generation. On the other hand, autoregressive models are pre-trained to generate the next word, and thus, they are better at sequential text generation.

Transformer-XL (autoregressive model) doesn't suffer from the same limitations as BERT. Even though Transformer-XL model doesn't suffer from the same limitations, it doesn't capture the bi-

How does XLNet add **Bi-directional context** to an autoregressive model?

XLNet proposes a novel retraining objective that maximizes the expected log-likelihood of a sequence concerning **all possible permutations of the factorization order.** The context for each position can consist of tokens from both left and right. Each position learns to utilize contextual information from all positions, thus capturing a bidirectional context.


x_3
$h_1^{(2)}$ $h_2^{(2)}$ $h_3^{(2)}$ $h_4^{(2)}$
mem$^{(1)}$ $h_1^{(1)}$ $h_2^{(1)}$ $h_3^{(1)}$ $h_4^{(1)}$
mem$^{(0)}$ x_1 x_2 x_3 x_4

Factorization order: 3 → 2 → 4 → 1

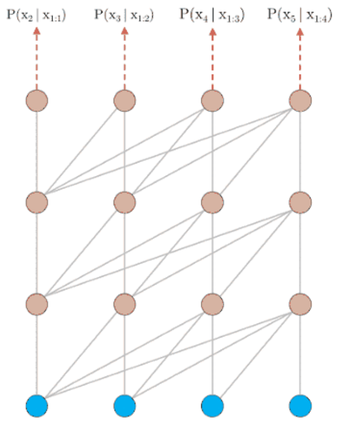
Factorization order: 2 → 4 → 3 → 1

Factorization order: 1 → 4 → 2 → 3

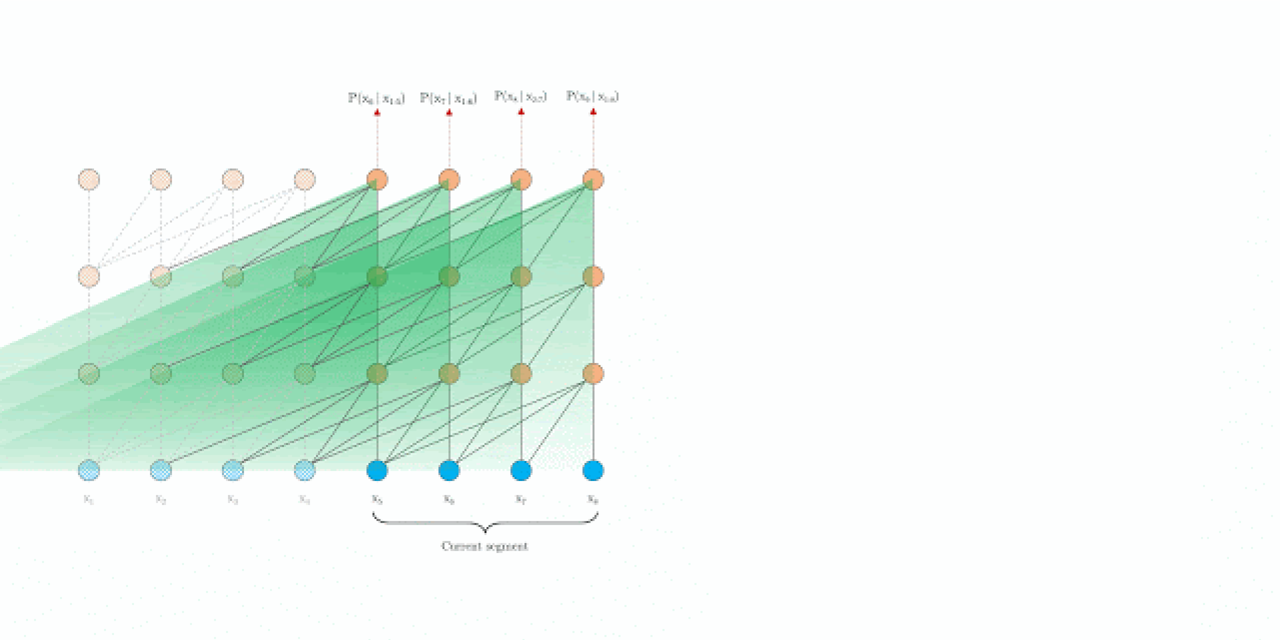
Factorization order: 4 → 3 → 1 → 2

Objective: Permutation Language Modeling

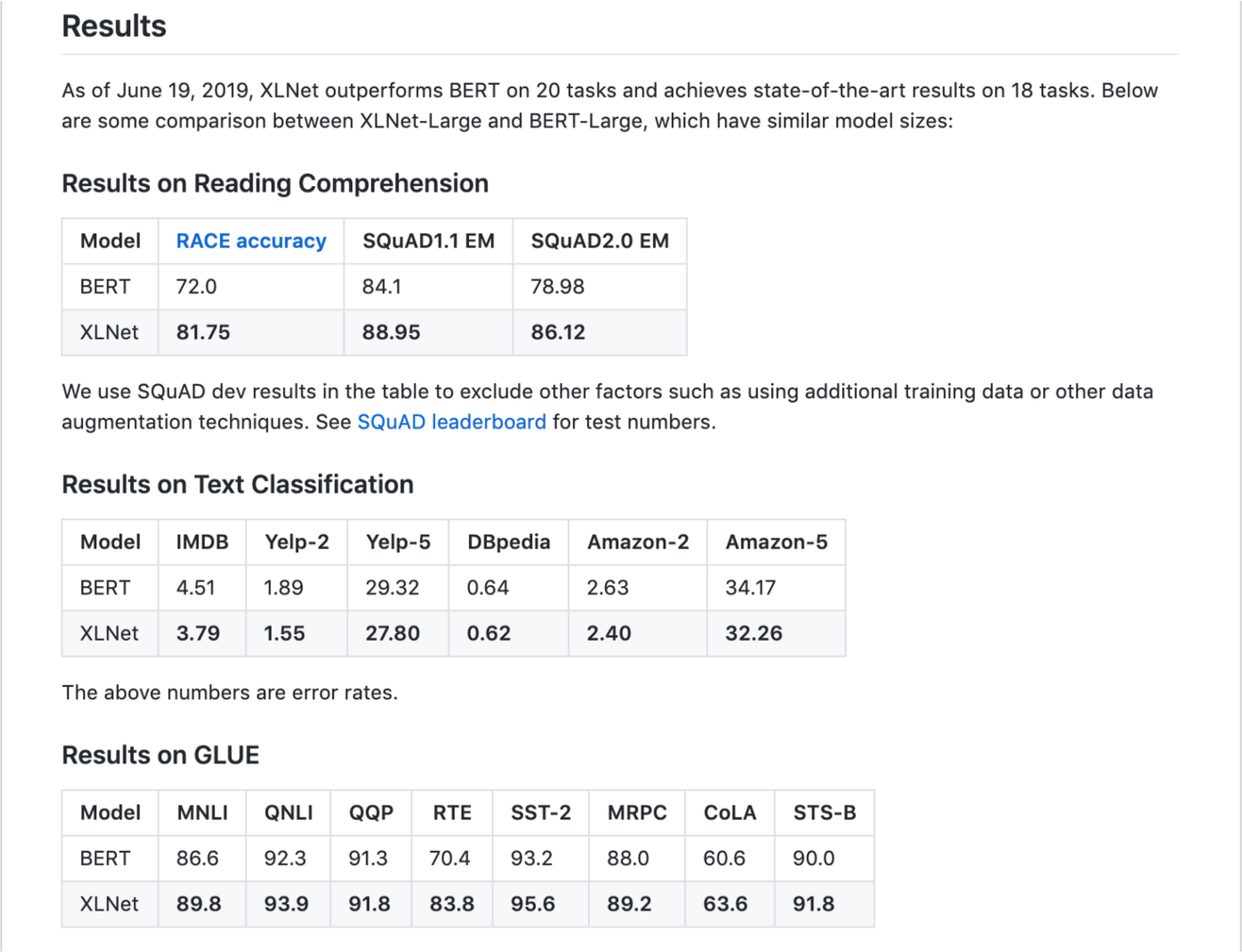
XLNet also integrates the **segment recurrence mechanism** and **relative encoding scheme** of **Transformer-XL** into pretraining, which improves the performance, especially for tasks involving a more extended text sequence.



Segment-level recurrence at training time.



Segment-level recurrence at evaluation time.



Results

As of June 19, 2019, XLNet outperforms BERT on 20 tasks and achieves state-of-the-art results on 18 tasks. Below are some comparison between XLNet-Large and BERT-Large, which have similar model sizes:

Results on Reading Comprehension

Model	RACE accuracy	SQuAD1.1 EM	SQuAD2.0 EM
BERT	72.0	84.1	78.98
XLNet	**81.75**	**88.95**	**86.12**

We use SQuAD dev results in the table to exclude other factors such as using additional training data or other data augmentation techniques. See SQuAD leaderboard for test numbers.

Results on Text Classification

Model	IMDB	Yelp-2	Yelp-5	DBpedia	Amazon-2	Amazon-5
BERT	4.51	1.89	29.32	0.64	2.63	34.17
XLNet	**3.79**	**1.55**	**27.80**	**0.62**	**2.40**	**32.26**

The above numbers are error rates.

Results on GLUE

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B
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**

Results

The entire tensorflow implementation code for the paper is available on **GitHub.**

. . .

[1]: XLNet isn't the only model to outperform BERT. **Microsoft's MT-DNN (Multi-Task Deep Neural Network)** model outperforms BERT on 9 out of 11 benchmark NLP tasks. **Facebook Research's XLM — Cross-lingual pre-training method** outperforms BERT on all GLUE tasks with SOTA on XNLI and unsupervised MT but both of these models are based on BERT.

Machine Learning Deep Learning NLP Sota Transfer Learning

24 claps

WRITTEN BY

Ankur Bohra

Follow

logits.ai

logits.ai

logits.ai

Follow

Write the first response

More From Medium

Also tagged Transfer Learning Related reads Related reads

M | logits.ai

Sign in

Get started

Using Transfer Learning to Detect Malaria Diseases

Satsawat...

Jun 18 · 5 mi...

268

State-of-the-art Multilingual Lemmatization

Erick...

Mar 1...

225

Using FastAI to Analyze Yelp Reviews and Predict User Ratings (Polarity)

Sho F...

Mar 1...

73