

The Journey from LSTM to BERT

All slides are my own. Citations provided for borrowed images

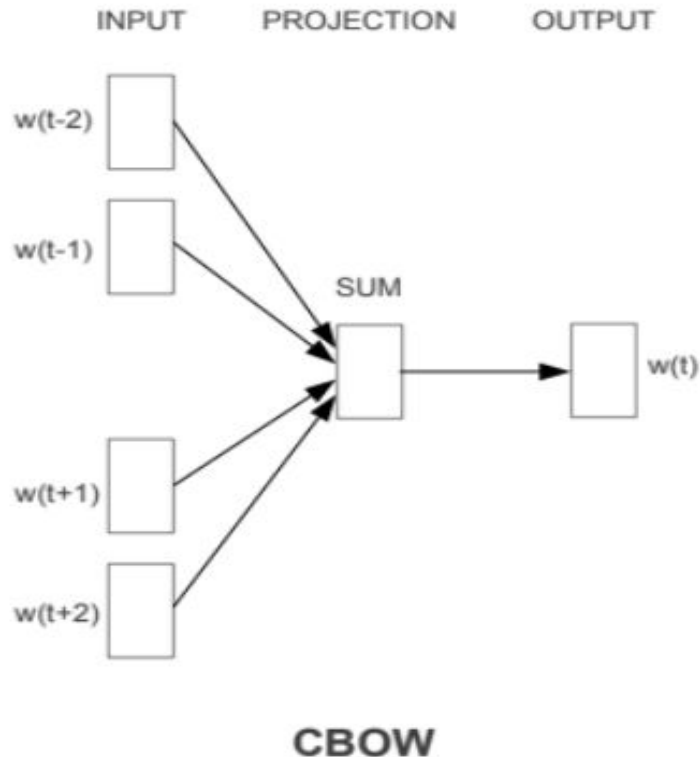
Kolluru Sai Keshav
PhD Scholar

Concepts

- Self-Attention
 - Pooling
 - Attention (Seq2Seq, Image Captioning)
 - Structured Self-Attention in LSTMs
 - Transformers
- **LM-based pretraining**
 - ELMo
 - ULMiFit
 - GPT
- GLUE Benchmark
- BERT
- Extensions: Roberta, ERNIE

Word2Vec

Vaibhav: similar to MLM

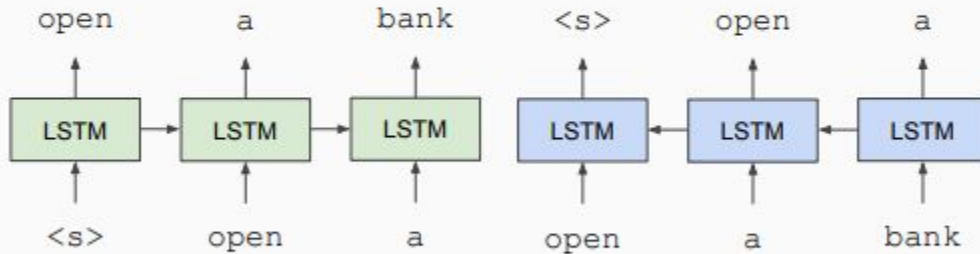


- Converts words to vectors such that similar words are located near to each other in the vector space
- Made possible using CBOW (Continuous Bag of Words) objective
- Words in the context are used to predict the middle word
- Words with similar contexts are embedded close to each other
“A word is known by the company it keeps”

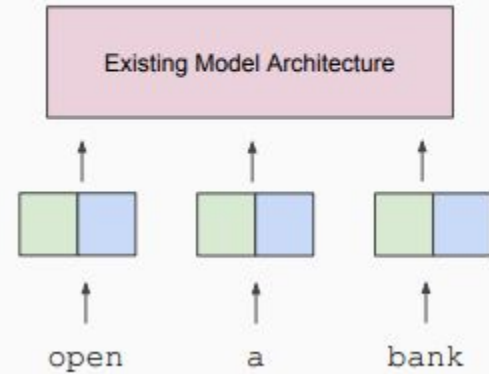
Contextualized Word Representations (ELMo)

- Bidirectional language modelling using separate forward and backward LSTMs
- Issue: Both LSTMs are not coupled with one another

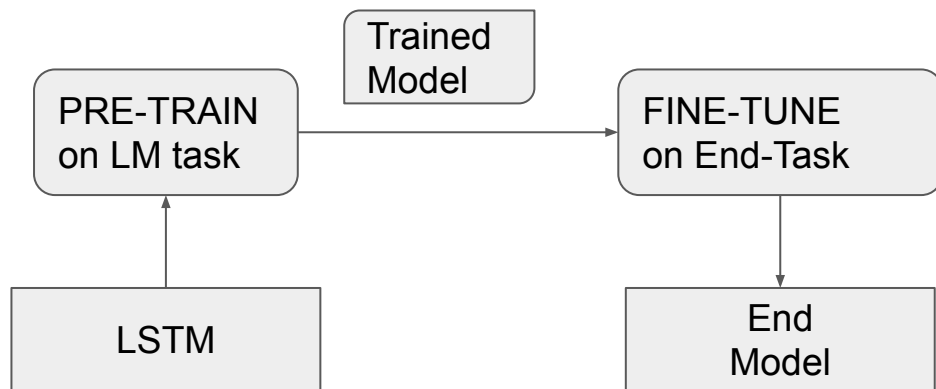
Train Separate Left-to-Right and Right-to-Left LMs



Apply as “Pre-trained Embeddings”



Universal Language Model Fine-tuning for Text Classification

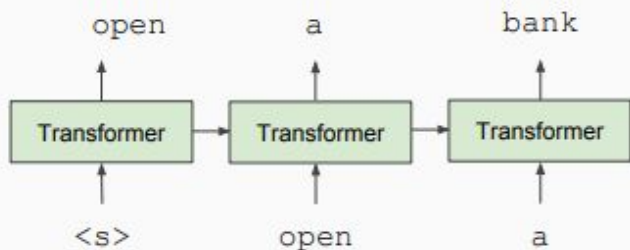


- Uses the same architecture for both pretraining and finetuning
- ELMo is added as additional component to existing task-specific architectures
- Introduced the Pretrain-Finetune paradigm for NLP
- Similar to pretraining ResNet on ImageNet and finetune on specific tasks
- Pretrained using Language modelling task
- Finetuned on End-Task (such as Sentiment Analysis)

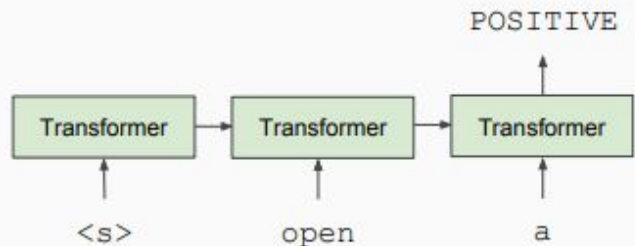
Generative Pre-training

- GPT - Uses Transformer decoder instead of LSTM for Language Modeling
- GPT-2 - Trained on larger corpus of text (40 GB) Model size:1.5 B parameters
- Can generate text given initial prompt - “unicorn” story, economist interview

Train Deep (12-layer) Transformer LM



Fine-tune on Classification Task



Unicorn Story

SYSTEM PROMPT
(HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL
COMPLETION
(MACHINE-
WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Concepts

- Self-Attention
 - Pooling
 - Attention (Seq2Seq, Image Captioning)
 - Structured Self-Attention in LSTMs
 - Transformers
- LM-based pretraining
 - ELMo
 - ULMiFit
 - GPT
- GLUE Benchmark
- **BERT**
- Extensions: Roberta, ERNIE

BERT : Masked language modelling

- GPT-2 is unidirectional. Tasks like classification - we already know all the words - using unidirectional model is sub-optimal
- But language modeling objective is inherently unidirectional

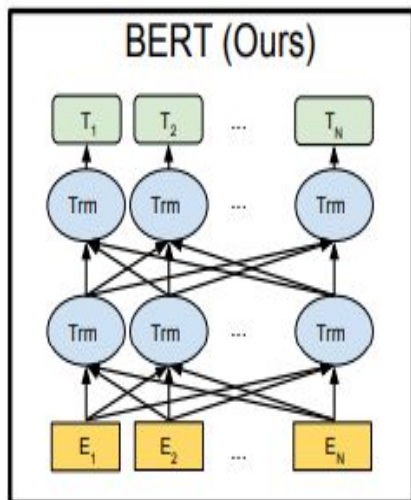
- **Solution:** Mask out $k\%$ of the input words, and then predict the masked words
 - We always use $k = 15\%$

the man went to the [MASK] to buy a [MASK] of milk

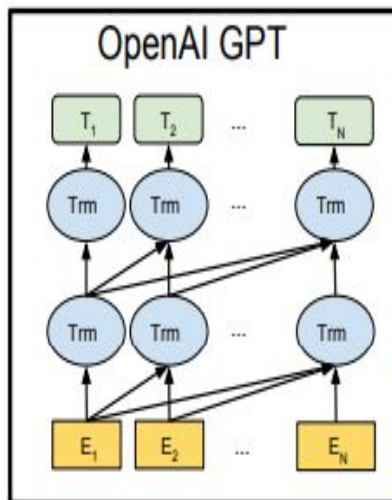
store gallon

↑ ↑

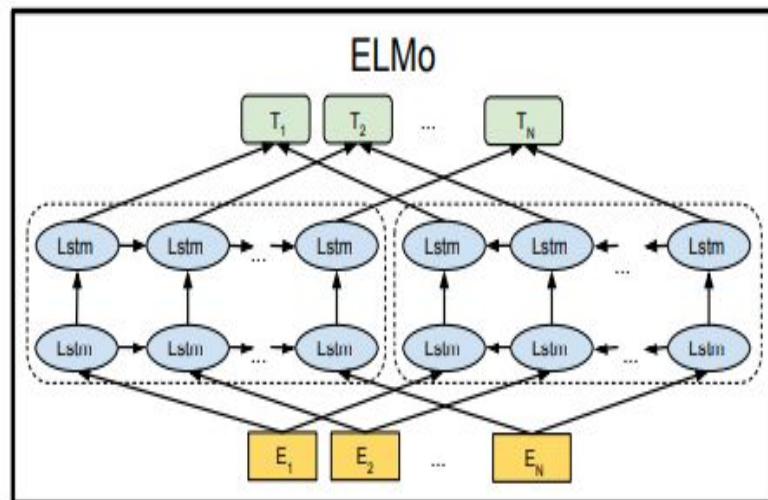
BERT vs. OpenAI-GPT vs. ELMo



Bidirectional

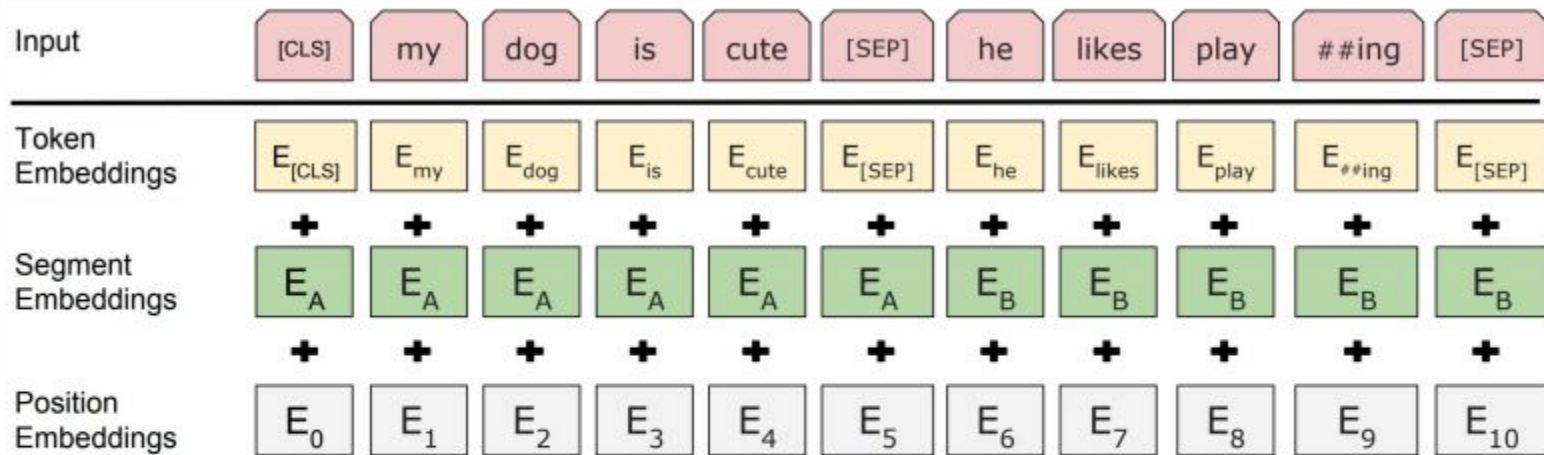


Unidirectional



De-coupled
Bidirectionality

Input Representation



- Use 30,000 WordPiece vocabulary on input.
- Each token is sum of three embeddings

Word-Piece tokenizer

Atishya, Siddhant: UNK tokens

- Middle ground between character level and word level representations
- tweeting → tweet + ##ing
- xanax → xa + ##nax
- Technique originally taken from paper for Japanese and Korean languages from a speech conference

- Given a training corpus and a number of desired tokens D , the optimization problem is to select D wordpieces such that the resulting corpus is minimal in the number of wordpieces when segmented according to the chosen wordpiece model.

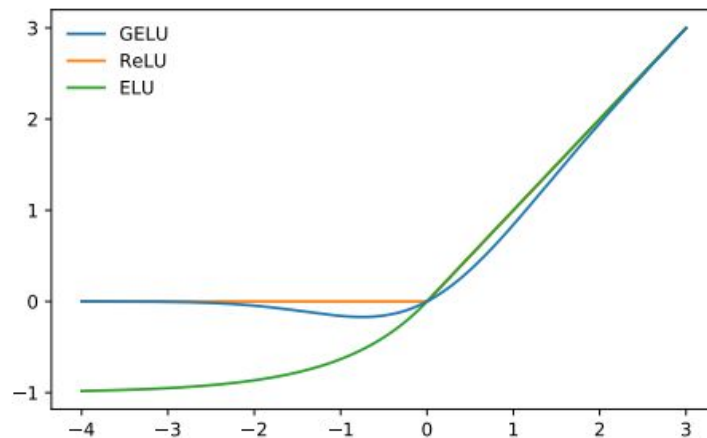
Schuster, Mike, and Kaisuke Nakajima. "Japanese and korean voice search." *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012.

Misc Details

- Uses an activation function called GeLU - a continuous version of ReLU
- Multiplies the input with a stochastic one-zero map (in the expectation)

$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x).$$

$$0.5x(1 + \tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$



- Optimizer: A variant of the Adam optimizer where the learning rate first increases (Warm-up phase) and is then decayed

Practical Tips

- Proper modelling of input for BERT is extremely important
 - Question Answering: [CLS] Query [SEP] Passage [SEP]
 - Natural Language Inference: [CLS] Sent1 [SEP] Sent2 [SEP]
 - BERT cannot be used as a general purpose sentence embedder
- Maximum input length is limited to 512. Truncation strategies have to be adopted
- BERT-Large model requires random restarts to work
- Always PRE-TRAIN, on related task - will improve accuracy
- Highly optimized for TPUs, not so much for GPUs

Atishya: TPUs vs.
GPUs

Small Hyperparameter search

- Because of using a pre-trained model - we can't really change the model architecture any more
- Number of hyper-parameters are actually few:
 - Batch Size: 16, 32
 - Learning Rate: $3e-6$, $1e-5$, $3e-5$, $5e-5$
 - Number of epochs to run
- Compare to LSTMs where we need to decide number of layers, the optimizer, the hidden size, the embedding size, etc...
- This greatly simplifies using the model

Implementation for fine-tuning

- Using BERT requires 3 modules
 - Tokenization, Model and Optimizer
- Originally developed in Tensorflow
- HuggingFace ported it to Pytorch and to-date remains the most popular way of using BERT (18K stars)
- Tensorflow 2.0 also has a very compact way of using it - from TensorflowHub
 - But fewer people use it, so support is low
- My choice - use HuggingFace BERT API with Pytorch-Lightning
 - Lightning provides a Keras-like API for Pytorch

Concepts

- Self-Attention
 - Pooling
 - Attention (Seq2Seq, Image Captioning)
 - Structured Self-Attention in LSTMs
 - Transformers
- LM-based pretraining
 - ELMo
 - ULMiFit
 - GPT
- **GLUE Benchmark**
- BERT
- Extensions: Roberta, ERNIE

Evaluating Progress: [GLUE-benchmark](#)

Corpus	Train	Dev	Test	Task	Metric	Domain
Single-Sentence Tasks						
CoLA	10k	1k	1.1k	acceptability	Matthews	linguistics literature
SST-2	67k	872	1.8k	sentiment	acc.	movie reviews
Similarity and Paraphrase Tasks						
MRPC	4k	N/A	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.5k	1.4k	sentence similarity	Pearson/Spearman	misc.
QQP	400k	N/A	391k	paraphrase	acc./F1	social QA Questions
Inference Tasks						
MNLI	393k	20k	20k	NLI	acc. (match/mismatch)	misc.
QNLI	108k	11k	11k	QA/NLI	acc.	Wikipedia
RTE	2.7k	N/A	3k	NLI	acc.	misc.
WNLI	706	N/A	146	coreference/NLI	acc.	fiction books

Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-Benchmark, which is a regression task. MNLI has three classes while all other classification tasks are binary.

DecaNLP - a forgotten benchmark

Leaderboard

Rank	Model	decaScore	Breakdown by Task						
			SQuAD	IWSLT	CNN/DM	MNLI	SST	QA-SRL	QA-ZRE
1 June 20, 2018	MQAN <i>Salesforce Research</i>	590.5	SQuAD	74.4	QA-SRL	78.4			
			IWSLT	18.6	QA-ZRE	37.6			
			CNN/DM	24.3	WOZ	84.8			
			MNLI	71.5	WikiSQL	64.8			
			SST	87.4	MWSC	48.7			
2 May 18, 2018	Sequence-to- sequence baseline <i>Salesforce Research</i>	513.6	SQuAD	47.5	QA-SRL	68.7			
			IWSLT	14.2	QA-ZRE	28.5			
			CNN/DM	25.7	WOZ	84.0			
			MNLI	60.9	WikiSQL	45.8			
			SST	85.9	MWSC	52.4			

- Spans 10 tasks
- Question Answering (SQUAD)
- Summarization (CNN/DM)
- Natural Language Inference (MNLI)
- Semantic Parsing (WikiSQL)
-
- Interesting choice of tasks but did not pick up steam
- Model designers had to manually communicate the results
- GLUE had an automatic system

Surprising effectiveness of BERT

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
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.⁸ BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

BERT as Feature Extractor

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	93.1
Fine-tuning approach		
BERT _{LARGE}	96.6	92.8
BERT _{BASE}	96.4	92.4
Feature-based approach (BERT _{BASE})		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Table 7: CoNLL-2003 Named Entity Recognition results. Hyperparameters were selected using the Dev set. The reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

Ablation Study

Tasks	Dev Set				
	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
BERT _{BASE}	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the BERT_{BASE} architecture. “No NSP” is trained without the next sentence prediction task. “LTR & No NSP” is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. “+ BiLSTM” adds a randomly initialized BiLSTM on top of the “LTR + No NSP” model during fine-tuning.

Self-Supervised Learning



Yann LeCun shared a photo.

30 April 2019 · 🌐

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of its input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

Concepts

- Self-Attention
 - Pooling
 - Attention (Seq2Seq, Image Captioning)
 - Structured Self-Attention in LSTMs
 - Transformers
- LM-based pretraining
 - ELMo
 - ULMiFit
 - GPT
- GLUE Benchmark
- BERT
- **Extensions: Roberta, ERNIE**

Roberta: A Robustly Optimized BERT Pretraining Approach

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	94.6/89.4	90.2	96.4
BERT _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB \rightarrow 160GB of text) and pretrain for longer (100K \rightarrow 300K \rightarrow 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT_{LARGE}. Results for BERT_{LARGE} and XLNet_{LARGE} are from Devlin et al. (2019) and Yang et al. (2019), respectively. Complete results on all GLUE tasks can be found in the Appendix.

ERNIE: A Continual Pre-Training Framework for Language Understanding

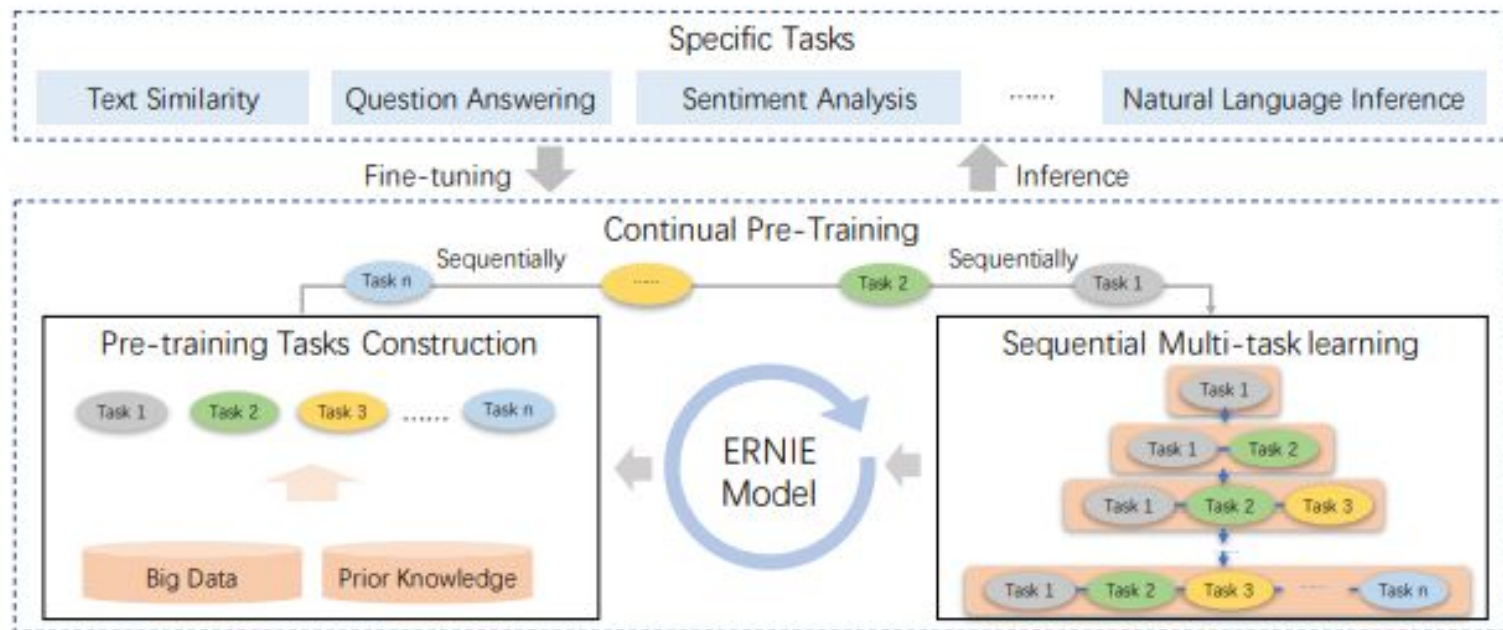
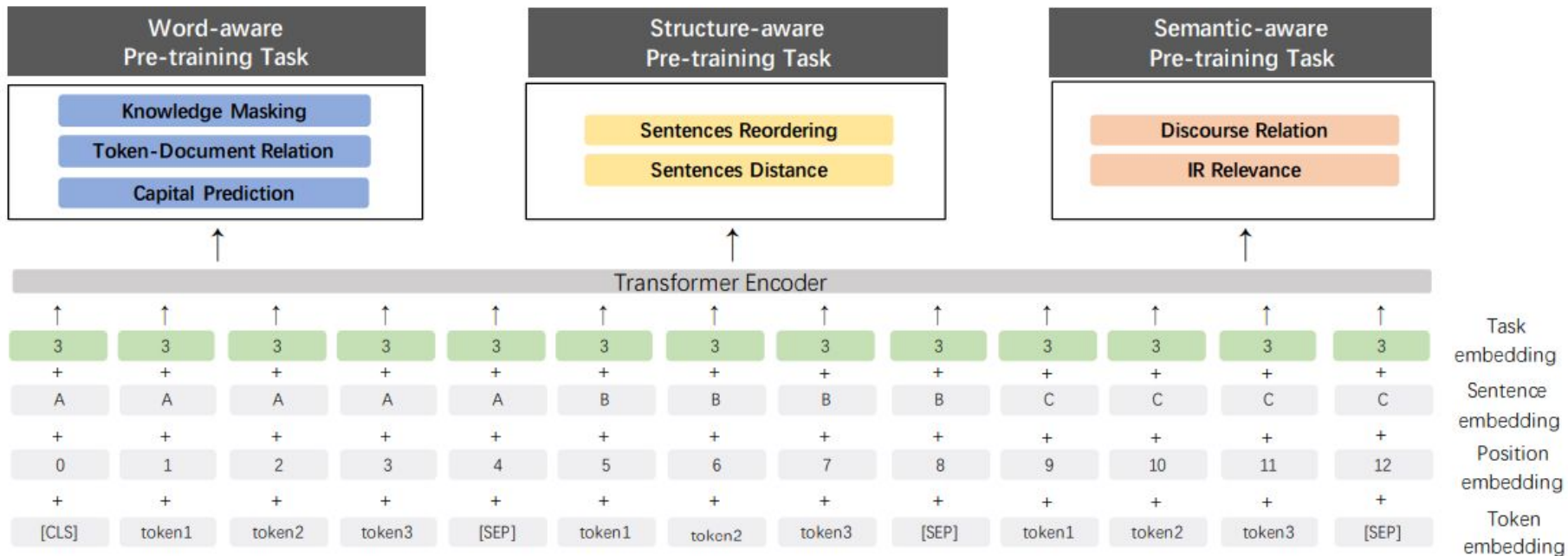











Figure 1: The framework of ERNIE 2.0, where the pre-training tasks can be incrementally constructed, the models are pre-trained through continual multi-task learning, and the pre-trained model is fine-tuned to adapt to various language understanding tasks.

Pre Training tasks in ERNIE



Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	ERNIE Team - Baidu	ERNIE		90.1	72.2	97.5	93.0/90.7	92.9/92.5	75.2/90.8	91.2	90.6	98.0	90.4	94.5	49.4
2	Microsoft D365 AI & MSR AI	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
3	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
+	4 王玮	ALICE v2 large ensemble (Alibaba D		89.5	71.3	97.1	93.9/91.9	93.0/92.5	74.8/91.0	90.7	90.4	99.2	87.4	91.8	48.4
5	XLNet Team	XLNet (ensemble)		89.5	70.2	97.1	92.9/90.5	93.0/92.6	74.7/90.4	90.9	90.9	99.0	88.5	92.5	48.4
6	ALBERT-Team Google	LanguALBERT (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
7	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
8	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
9	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3

Snapshot taken on 24th December, 2019

Review of Reviews

- (Sankalan, Vaibhav) Using image as input: **VL-BERT**
- (Sankalan) Using KB facts as input (KB-QA): **Retrieval+Concatenation**
- Using BERT as a KB: **E-BERT**
- (Atishya) Inter-dependencies between masked tokens: **XL-Net**
- (Rajas) Freeze layers while fine-tuning: **Adapter-BERT**
 - 0.4% accuracy drop adding only 3.6% parameters
- (Rajas) Pre-training over multiple tasks: **ERNIE** (with a curriculum)
- (Shubham) Fine-training over multiple tasks: **MT-DNN, SMART**

Review of Reviews

- (Pratyush) Masking using NER: **ERNIE**
- (Jigyasa) Model Compression: **DistilBERT, MobileBERT**
 - Reduces size of BERT by 40%, improves inference by 60% while achieving 99% of the results
- (Saransh) Using BERT for VQA: **LXMBERT**
- (Siddhant) Analyzing BERT: *Bertology*
 - Though post-facto and not axiomatic
- (Soumya) Issue with breaking negative affixes: *Whole-word masking*
- (Vipul) Pre-training on supervised tasks: **Universal Sentence Repr.**
- (Lovish) Introducing language embeddings: **mBART, T5** (task-embedding)
- (Pavan) Text-Generation tasks: **GPT-2, T5, BART**