Day 7 - Contextual Embeddings, Pretrained Models, and Transfer Learning

Advanced Text as Data: Natural Language Processing
Essex Summer School in Social Science Data Analysis

Burt L. Monroe (Instructor) & Sam Bestvater (TA)
Pennsylvania State University

August 4, 2021
Transfer Learning

1 - **Semi-supervised** training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

**Semi-supervised Learning Step**

**Model:**
BERT

**Dataset:**
Predict the masked word (language modeling)

2 - **Supervised** training on a specific task with a labeled dataset.

**Supervised Learning Step**

**Model:**
(pre-trained in step #1)

**Dataset:**

<table>
<thead>
<tr>
<th>Email message</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy these pills</td>
<td>Spam</td>
</tr>
<tr>
<td>Win cash prizes</td>
<td>Spam</td>
</tr>
<tr>
<td>Dear Mr. Atreides, please find attached...</td>
<td>Not Spam</td>
</tr>
</tbody>
</table>
Transfer Learning

- Transductive transfer learning
  - Domain adaptation
  - Cross-lingual learning

- Inductive transfer learning
  - Multi-task learning
  - Sequential transfer learning

Source: Becker, et al. 2020, Modern Approaches to Natural Language Processing
Feature Extraction / Contextualized Word Embeddings

Pretraining with source data

1 B Word Benchmark

ELMo
Shallowly bidirectional Language Model (LSTM)

Feature Extraction with target data

Adopted model
question answering
classification
sequence labeling
....

Source: Becker, et al. 2020, *Modern Approaches to Natural Language Processing*
CoVe (SalesForce, 2017)

Pretrained on machine translation. Keep the encoder and reuse for other task.
(ELMo, AllenNLP 2018)
ELMo combines a forward and backward language model
Embedding of “stick” in “Let’s stick to” - Step #2

1- Concatenate hidden layers

2- Multiply each vector by a weight based on the task

3- Sum the (now weighted) vectors

ELMo embedding of “stick” for this task in this context
*Kitaev and Klein, ACL 2018  (see also Joshi et al., ACL 2018)
Fine-Tuning (ULMFiT - Howard & Ruder 2018)

Pretraining with source data

Wikitext-103
28,595 preprocessed articles and 103 million words

ULMFiT
AWD-LSTM

Fine-Tuning with target data

Adopted model
question answering
classification
sequence labeling
...

Source: Becker, et al. 2020, Modern Approaches to Natural Language Processing
Fine-Tuning (ULMFiT - Howard & Ruder 2018)

Techniques like “gradual unfreezing” help to prevent “catastrophic forgetting.”

Source: Becker, et al. 2020, Modern Approaches to Natural Language Processing
Use a transformer! GPT (Open-AI / Radford, et al., 2018)

Transformer architecture dramatically sped up training, allowing for deeper models (12 layers in the original GPT), and bigger training data (Books Corpus in the GPT used just the decoder, on an LM task.)
BERT (Google, 2018)
Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.
<table>
<thead>
<tr>
<th>Input</th>
<th>[CLS]</th>
<th>my</th>
<th>dog</th>
<th>is</th>
<th>cute</th>
<th>[SEP]</th>
<th>he</th>
<th>likes</th>
<th>play</th>
<th>'#ing'</th>
<th>[SEP]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token Embeddings</strong></td>
<td>$E_{[CLS]}$</td>
<td>$E_{my}$</td>
<td>$E_{dog}$</td>
<td>$E_{is}$</td>
<td>$E_{cute}$</td>
<td>$E_{[SEP]}$</td>
<td>$E_{he}$</td>
<td>$E_{likes}$</td>
<td>$E_{play}$</td>
<td>$E_{'#ing'}$</td>
<td>$E_{[SEP]}$</td>
</tr>
<tr>
<td><strong>Segment Embeddings</strong></td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_A$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
<td>$E_B$</td>
</tr>
<tr>
<td><strong>Position Embeddings</strong></td>
<td>$E_0$</td>
<td>$E_1$</td>
<td>$E_2$</td>
<td>$E_3$</td>
<td>$E_4$</td>
<td>$E_5$</td>
<td>$E_6$</td>
<td>$E_7$</td>
<td>$E_8$</td>
<td>$E_9$</td>
<td>$E_{10}$</td>
</tr>
</tbody>
</table>

BERT input representation
BERT - so what?

- BERT is a pretrained language model that can be fine-tuned to a specific task.

1. **Semi-supervised** training on large amounts of text (books, wikipedia, etc.).
   - The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

2. **Supervised** training on a specific task with a labeled dataset.

   ![Diagram](source: Alammar (2018))
Pretraining #1: Masked Language Modeling

Use the output of the masked word’s position to predict the masked word.

Randomly mask 15% of tokens.

Input

Possible classes: All English words

0.1% Aardvark
... ...
10% Improvisation
... ...
0% Zyzyva

BERT

FFNN + Softmax
Pretraining #2: Next Sentence Prediction

Predict likelihood that sentence B belongs after sentence A

Tokenized Input

Input

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]

Sentence A

Sentence B
Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different downstream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG
(b) Single Sentence Classification Tasks: SST-2, CoLA
(c) Question Answering Tasks: SQuAD v1.1
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER
The output of each encoder layer along each token's path can be used as a feature representing that token.

But which one should we use?
What is the best contextualized embedding for “Help” in that context?
For named-entity recognition task CoNLL-2003 NER

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Embedding</th>
<th>Dev F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Layer</td>
<td></td>
<td>91.0</td>
</tr>
<tr>
<td>Last Hidden Layer</td>
<td>12</td>
<td>94.9</td>
</tr>
<tr>
<td>Sum All 12 Layers</td>
<td>12</td>
<td>95.5</td>
</tr>
<tr>
<td>Second-to-Last Hidden Layer</td>
<td>11</td>
<td>95.6</td>
</tr>
<tr>
<td>Sum Last Four Hidden</td>
<td>12</td>
<td>95.9</td>
</tr>
<tr>
<td>Concat Last Four Hidden</td>
<td>9 10 11 12</td>
<td>96.1</td>
</tr>
</tbody>
</table>

Help
Highlights in pretrained models

- 2018 - ELMo (AllenNLP) - Contextualized word embeddings
- 2018 - ULMFit (fast.ai) - Fine-tuning a pretrained model
- 2018 - GPT - Use a transformer (decoder) - autoregressive
- 2018 - BERT (Google) - Bidirectional transformer encoder / auto-encoder, MLM/NSP
- Feb 2019 - GPT-2 (OpenAI) - 1.5 billion parameters
- Apr 2019 - ERNIE (Baidu) - masked phrases/entities in pretraining
- Jun 2019 - XLNet (CMU/Google) - permuted language modeling
- Jul 2019 - RoBERTa (Facebook) - pretraining differences, more training, more data
- Sep 2019 - ALBERT (Google) - Parameter-reduction techniques
Highlights in pretrained models

- Aug 2019 - StructBERT (Alibaba) - BERT masked LM + unshuffling scrambled word and sentence order.
- Sep 2019 - MegatronLM (Nvidia) - parallelism in pretraining, 8.3b parameter ~GPT-2, 3.9b ~BERT
- Oct 2019 - T5 (Google) - unifying text-to-text framework
- Jan 2020 - Reformer (Google) - locality-sensitive hashing allows context windows of 1m words
- Feb 2020 - Meena (Google) - 2.6B parameter chatbot
- Feb 2020 - Turing NLG (Microsoft) - 17 billion parameters
- May 2020 - GPT-3 (OpenAI) - 175 billion parameters
- May 2020 - ELECTRA (Stanford) - Efficiency from different pretraining
- Jul 2020 - DeBERTa (Microsoft) - “disentangled attention”
- May 2021 - OmniNET (Google) - “omnidirectional attention”
ERNIE (Baidu, 2019)

3 Methods

We introduce ERNIE and its detailed implementation in this section. We first describe the model’s transformer encoder, and then introduce the knowledge integration method in Section 3.2. The comparisons between BERT and ERNIE are shown visually in Figure 1.

3.1 Transformer Encoder

ERNIE use multi-layer Transformer (Vaswani et al., 2017) as basic encoder like previous pre-training model such as GPT, BERT and XLM. The Transformer can capture the contextual information for each token in the sentence via self-attention, and generates a sequence of contextual embeddings.

For Chinese corpus, we add spaces around every character in the CJK Unicode range and use the WordPiece (Wu et al., 2016) to tokenize Chinese sentences. For a given token, its input representation is constructed by summing the corresponding token, segment and position embeddings. The first token of every sequence is the special classification embedding ([CLS]).

3.2 Knowledge Integration

We use prior knowledge to enhance our pretrained language model. Instead of adding the knowledge embedding directly, we proposed a multi-stage knowledge masking strategy to integrate phrase and entity level knowledge into the Language representation. The different masking level of a sentence is described in Figure 2.

3.2.1 Basic-Level Masking

The first learning stage is to use basic level masking. It treats a sentence as a sequence of basic Language unit, for English, the basic language unit is word, and for Chinese, the basic language unit is Chinese Character. In the training process, we randomly mask 15 percents of basic language units, and using other basic units in the sentence as inputs, and train a transformer to predict the mask units. Based on basic level mask, we can obtain a basic word representation. Because it is trained on a random mask of basic semantic units, high level semantic knowledge is hard to be fully modeled.

3.2.2 Phrase-Level Masking

The second stage is to employ phrase-level masking. Phrase is a small group of words or characters together acting as a conceptual unit. For English, we use lexical analysis and chunking tools to get the boundary of phrases in the sentences, and use some language dependent segmentation tools to get the word/phrase information in other language such as Chinese. In phrase-level mask stage, we also use basic language units as training input, unlike random basic units mask, this time we randomly select a few phrases in the sentence, mask and predict all the basic units in the same phrase. At this stage, phrase information is encoded into the word embedding.

3.2.3 Entity-Level Masking

The third stage is entity-level masking. Name entities contain persons, locations, organizations, products, etc., which can be denoted with a proper

Figure 1: The different masking strategy between BERT and ERNIE

Training objective. BERT Masked LM + masked phrases and entities
StructBERT (Alibaba, 2019)

Training objective. BERT Masked LM + unshuffling scrambled words and sentences.
T5 (Google, 2019)

Unifying tasks as all text-to-text

- "translate English to German: That is good."
- "cola sentence: The course is jumping well."
- "sts-b 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."
- "Das ist gut."
- "not acceptable"
- "3.8"
- "six people hospitalized after a storm in attala county."

Trained on C4, new “Colossal Clean Crawled Corpus.” - 11b param
GPT-3 (Google, 2019)

Zero-, one-, and few-shot learning - 175 billion parameters
ALBERT (Google, 2019)

What's the core idea of this paper?

- It is not reasonable to further improve language models by making them larger because of memory limitations of available hardware, longer training times, and unexpected degradation of model performance with the increased number of parameters.
- To address this problem, the researchers introduce the ALBERT architecture that incorporates two parameter-reduction techniques:
  - factorized embedding parameterization, where the size of the hidden layers is separated from the size of vocabulary embeddings by decomposing the large vocabulary-embedding matrix into two small matrices;
  - cross-layer parameter sharing to prevent the number of parameters from growing with the depth of the network.
- The performance of ALBERT is further improved by introducing the self-supervised loss for sentence-order prediction to address BERT’s limitations with regard to inter-sentence coherence.

What's the key achievement?

- With the introduced parameter-reduction techniques, the ALBERT configuration with 18x fewer parameters and 1.7x faster training compared to the original BERT-large model achieves only slightly worse performance.
- The much larger ALBERT configuration, which still has fewer parameters than BERT-large, outperforms all of the current state-of-the-art language modes by getting:
  - 89.4% accuracy on the RACE benchmark;
  - 89.4 score on the GLUE benchmark; and
  - An F1 score of 92.2 on the SQuAD 2.0 benchmark.
XLNet (CMU/Google 2019)

Pretraining: Permuted Language Modeling

Figure 1: Illustration of the permutation language modeling objective for predicting $x_3$ given the same input sequence $x$ but with different factorization orders.
RoBERTa (Facebook, 2019)

What's the core idea of this paper?

- The Facebook AI research team found that BERT was significantly undertrained and suggested an improved recipe for its training, called RoBERTa:
  - More data: 160GB of text instead of the 16GB dataset originally used to train BERT.
  - Longer training: increasing the number of iterations from 100K to 300K and then further to 500K.
  - Larger batches: 8K instead of 256 in the original BERT base model.
  - Larger byte-level BPE vocabulary with 50K subword units instead of character-level BPE vocabulary of size 30K.
  - Removing the next sequence prediction objective from the training procedure.
  - Dynamically changing the masking pattern applied to the training data.

What's the key achievement?

- RoBERTa outperforms BERT in all individual tasks on the General Language Understanding Evaluation (GLUE) benchmark.
- The new model matches the recently introduced XLNet model on the GLUE benchmark and sets a new state of the art in four out of nine individual tasks.
<table>
<thead>
<tr>
<th></th>
<th>BERT</th>
<th>RoBERTa</th>
<th>DistilBERT</th>
<th>XLNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size (millions)</strong></td>
<td>Base: 110</td>
<td>Base: 110</td>
<td>Base: 66</td>
<td>Base: ~110</td>
</tr>
<tr>
<td></td>
<td>Large: 340</td>
<td>Large: 340</td>
<td></td>
<td>Large: ~340</td>
</tr>
<tr>
<td><strong>Training Time</strong></td>
<td>Base: 8 x V100 x 12 days*</td>
<td>Large: 1024 x V100 x 1 day; 4-5 times more than BERT.</td>
<td>Base: 8 x V100 x 3.5 days; 4 times less than BERT.</td>
<td>Large: 512 TPU Chips x 2.5 days; 5 times more than BERT.</td>
</tr>
<tr>
<td></td>
<td>Large: 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Outperforms state-of-the-art in Oct 2018</td>
<td>2-20% improvement over BERT</td>
<td>3% degradation from BERT</td>
<td>2-15% improvement over BERT</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.</td>
<td>160 GB (16 GB BERT data + 144 GB additional)</td>
<td>16 GB BERT data. 3.3 Billion words.</td>
<td>Base: 16 GB BERT data Large: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.</td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>BERT (Bidirectional Transformer with MLM and NSP)</td>
<td>BERT without NSP**</td>
<td>BERT Distillation</td>
<td>Bidirectional Transformer with Permutation based modeling</td>
</tr>
</tbody>
</table>
DeBERTa (Microsoft 2020)

Disentangled Attention

What’s the key achievement?

- Compared to the current state-of-the-art method RoBERTa-Large, the DeBERTA model trained on half the training data achieves:
  - an improvement of +0.9% in accuracy on MNLI (91.1% vs. 90.2%),
  - an improvement of +2.3% in accuracy on SQuAD v2.0 (90.7% vs. 88.4%),
  - an improvement of +3.6% in accuracy on RACE (86.8% vs. 83.2%)
- A single scaled-up variant of DeBERTa surpasses the human baseline on the SuperGLUE benchmark for the first time (89.9 vs. 89.8). The ensemble DeBERTa is the top-performing method on SuperGLUE at the time of this publication, outperforming the human baseline by a decent margin (90.3 versus 89.8).
ELECTRA (Stanford, 2020)

Training objective. Replaced token detection.

Outperforms BERT w/ similar parameters, matches RoBERTa and XLNet w/ 25% compute.
Arms Race

GPT-3 - 175000!
Overview / Taxonomy
Pretraining Model Architectures

A.2 Pre-training Procedure

For pre-training, we adopt the mechanism of contrastive learning with masked language modeling. In each step, we randomly mask 15% of tokens and reconstruct them. We use a batch size of 256 sequences (256 sentences, which is done for the “next sentence prediction” task). The training process is performed on 4 TPU-nodes, and the learning rate is 1e-4. The embedding layer is updated with Adam optimizer. The training is performed for 1,000,000 steps, which is approximately 40 million words.

Next Sentence Prediction

Label

Figure 3: Taxonomy of PTMs with Representative Examples

Pre-trained Models for Natural Language Processing: A Survey

Xipeng Qiu*, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang
Pretraining Tasks

MT: Machine Translation
LM: Language Modeling
MLM: Masked Language Modeling
PLM: Permutated Language Modeling
DAE: Denoising Autoencoder

CTL: Contrastive Learning
   RTD: Replaced Token Detection
NSP: Next Sentence Prediction
SOP: Sentence Order Prediction

Pre-trained Models for Natural Language Processing: A Survey
Xipeng Qiu*, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang
Variants

PTMs

Extensions

Strategies

Tuning

Pre-Training

Architectures

QIU XP, et al.

Early Exit

Model Compression

Domain-Specific

Multi-Modal

Language-Specific

Multilingual

Knowledge-Enriched

ERNIE(THU) [76], KnowBERT [77], K-BERT [78], SentiLR [79], KEPLER [80]

WKLM [57], CoLAKE [81]

XLU

mBERT [16], Unicoder [82], XLM [46], XLM-R [62], MultiFit [83]

XLG

MASS [41], mBERT [61], XNLG [60]

ERNIE(Baidu) [84], BERT-wwm-Chinese [85], NEZHA [86], ZEN [87], BERTje [88]

CamemBERT [89], FlauBERT [90], RobBERT [91]

Image

ViLBERT [92], LXMERT [93], VisualBERT [94], B2T2 [95], VL-BERT [96]

Video

VideoBERT [97], CBT [98]

Speech

SpeechBERT [99]

SentiLR [79], BioBERT [100], SciBERT [101], PatentBERT [102]

Model Pruning

CompressingBERT [103]

Quantization

Q-BERT [104], Q8BERT [105]

Parameter Sharing

ALBERT [63]

Distillation

DistilBERT [106], TinyBERT [107], MiniLM [108]

Module Replacing

BERT-of-Theseus [109]

DecBERT [110], RightTool [111], FastBERT [112], PABEE [113], Liao et al. [114], Sun et al. [115]

SentEE/TokEE [116]

Pre-trained Models for Natural Language Processing: A Survey

Xipeng Qiu*, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai & Xuanjing Huang
Benchmarks / leaderboards
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
<th>Data example</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoLA</td>
<td>Is the sentence grammatical or ungrammatical?</td>
<td>&quot;This building is than that one.&quot; = Ungrammatical</td>
<td>Matthews</td>
</tr>
<tr>
<td>SST-2</td>
<td>Is the movie review positive, negative, or neutral?</td>
<td>&quot;The movie is funny, smart, visually inventive, and most of all, alive.&quot; = 0.93056 (Very Positive)</td>
<td>Accuracy</td>
</tr>
</tbody>
</table>
| MRPC      | Is the sentence B a paraphrase of sentence A?                               | A) "Yesterday, Taiwan reported 35 new infections, bringing the total number of cases to 418." 
B) "The island reported another 35 probable cases yesterday, taking its total to 418." = A Paraphrase | Accuracy / F1        |
| STS-B     | How similar are sentences A and B?                                          | A) "Elephants are walking down a trail." 
B) "A herd of elephants are walking along a trail." = 4.6 (Very Similar) | Pearson / Spearman   |
| QQP       | Are the two questions similar?                                              | A) "How can I increase the speed of my internet connection while using a VPN?" 
B) "How can Internet speed be increased by hacking through DNS?" = Not Similar | Accuracy / F1        |
| MNLI-mm   | Does sentence A entail or contradict sentence B?                            | A) "Tourist Information offices can be very helpful." 
B) "Tourist Information offices are never of any help." = Contradiction | Accuracy             |
| QNLI      | Does sentence B contain the answer to the question in sentence A?           | A) "What is essential for the mating of the elements that create radio waves?" 
B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = Answerable | Accuracy             |
| RTE       | Does sentence A entail sentence B?                                          | A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." 
B) "Yunus supported more than 50,000 Struggling Members." = Entailed | Accuracy             |
| WNLI      | Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun? | A) "Lily spoke to Donna, breaking her concentration." 
B) "Lily spoke to Donna, breaking Lily's concentration." = Incorrect Referent | Accuracy             |
<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>URL Score ColA SST-2</th>
<th>MRPC</th>
<th>STS-B</th>
<th>QQP</th>
<th>MNLI-m</th>
<th>MNLI-mm</th>
<th>QNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AliceMind &amp; DIRL + CLEVER</td>
<td>91.0 75.3 97.7 93.9/91.9 93.5/93.1 75.6/90.8</td>
<td>91.7</td>
<td>91.5</td>
<td>97.4</td>
<td>92.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ERNIE Team - Baidu</td>
<td>90.9 74.3 97.8 93.9/91.8 93.0/92.6 75.2/90.9</td>
<td>91.9</td>
<td>91.4</td>
<td>97.3</td>
<td>92.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DeBERTA Team - Microsoft</td>
<td>DeBERTA / TuringNLRv4</td>
<td>90.8 71.5 97.5 94.0/92.0 92.9/92.6 76.2/90.8</td>
<td>91.9</td>
<td>91.6</td>
<td>99.2</td>
<td>93.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>HFL, IFLYTEK</td>
<td>MacALBERT + DLM</td>
<td>90.7 74.8 97.0 94.5/92.6 92.8/92.6 74.7/90.6</td>
<td>91.3</td>
<td>91.1</td>
<td>97.8</td>
<td>92.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PING-AN Omni-Sentic</td>
<td>ALBERT + DAAF + NAS</td>
<td>90.6 73.5 97.2 94.0/92.0 93.0/92.4 76.1/91.0</td>
<td>91.6</td>
<td>91.3</td>
<td>97.5</td>
<td>91.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>liangzhuge</td>
<td>Deberta + adv (ensemble)</td>
<td>90.4 72.7 97.3 92.7/90.3 93.2/92.9 76.5/90.6</td>
<td>91.7</td>
<td>91.5</td>
<td>96.4</td>
<td>92.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TS Team - Google</td>
<td>TS</td>
<td>90.3 71.6 97.5 92.8/90.4 93.1/92.8 75.1/90.6</td>
<td>92.2</td>
<td>91.9</td>
<td>96.9</td>
<td>92.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Microsoft D365 AI &amp; MSR AI &amp; GATECH</td>
<td>MTDNN-SMART</td>
<td>89.9 69.5 97.5 92.7/91.6 92.9/92.5 73.9/90.2</td>
<td>91.0</td>
<td>90.8</td>
<td>99.2</td>
<td>89.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Huawei Noah's Ark Lab</td>
<td>NEZHA-Large</td>
<td>89.8 71.7 97.3 93.3/91.0 92.4/91.9 75.2/90.7</td>
<td>91.5</td>
<td>91.3</td>
<td>96.2</td>
<td>90.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Zihang Dai</td>
<td>Funnel-Transformer (Ensemble B10-10-10H1024)</td>
<td>89.7 70.5 97.5 93.4/91.2 92.6/92.3 75.4/90.7</td>
<td>91.4</td>
<td>91.1</td>
<td>95.8</td>
<td>90.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ELECTRA Team</td>
<td>ELECTRA-Large + Standard Tricks</td>
<td>89.4 71.7 97.1 93.1/90.7 92.9/92.5 75.6/90.8</td>
<td>91.3</td>
<td>90.8</td>
<td>95.8</td>
<td>89.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Microsoft D365 AI &amp; UMD</td>
<td>FreeLB-DeBERTa (ensemble)</td>
<td>88.4 68.0 96.8 93.1/90.8 92.3/92.1 74.8/90.3</td>
<td>91.1</td>
<td>90.7</td>
<td>95.6</td>
<td>88.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Junjie Yang</td>
<td>HIRE-DeBERTa</td>
<td>88.3 68.6 97.1 93.0/90.7 92.4/92.0 74.3/90.2</td>
<td>90.7</td>
<td>90.4</td>
<td>95.5</td>
<td>87.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Facebook AI</td>
<td>RoBERTa</td>
<td>88.1 67.8 96.7 92.3/89.8 92.2/91.9 74.3/90.2</td>
<td>90.8</td>
<td>90.2</td>
<td>95.4</td>
<td>88.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Microsoft D365 AI &amp; MSR AI</td>
<td>MTDNN-ensemble</td>
<td>87.6 68.4 96.5 92.7/90.3 91.1/90.7 73.7/89.9</td>
<td>87.9</td>
<td>87.4</td>
<td>96.0</td>
<td>86.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>GLUE Human Baselines</td>
<td>GLUE Human Baselines</td>
<td>87.1 66.4 97.8 86.3/80.8 92.7/92.6 59.5/80.4</td>
<td>92.0</td>
<td>92.8</td>
<td>91.2</td>
<td>93.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Adrian de Wynter</td>
<td>Bort (Alexa AI)</td>
<td>84.6 63.9 96.2 94.1/92.3 89.2/88.3 66.0/85.9</td>
<td>88.1</td>
<td>87.8</td>
<td>92.3</td>
<td>82.</td>
<td></td>
</tr>
</tbody>
</table>
## SuperGLUE Tasks

<table>
<thead>
<tr>
<th>Name</th>
<th>Identifier</th>
<th>Download</th>
<th>More Info</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcoverage Diagnostics</td>
<td>AX-b</td>
<td></td>
<td></td>
<td>Matthew's Corr</td>
</tr>
<tr>
<td>CommitmentBank</td>
<td>CB</td>
<td></td>
<td></td>
<td>Avg. F1 / Accuracy</td>
</tr>
<tr>
<td>Choice of Plausible Alternatives</td>
<td>COPA</td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>Multi-Sentence Reading Comprehension</td>
<td>MultiRC</td>
<td></td>
<td></td>
<td>F1a / EM</td>
</tr>
<tr>
<td>Recognizing Textual Entailment</td>
<td>RTE</td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>Words in Context</td>
<td>WiC</td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>The Winograd Schema Challenge</td>
<td>WSC</td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>BoolQ</td>
<td>BoolQ</td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>Reading Comprehension with Commonsense Reasoning</td>
<td>ReCoRD</td>
<td></td>
<td></td>
<td>F1 / Accuracy</td>
</tr>
<tr>
<td>Winogender Schema Diagnostics</td>
<td>AX-g</td>
<td></td>
<td></td>
<td>Gender Parity / Accuracy</td>
</tr>
</tbody>
</table>

[DOWNLOAD ALL DATA]
### Leaderboard Version: 2.0

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Model</th>
<th>URL</th>
<th>Score</th>
<th>BoolQ</th>
<th>CB</th>
<th>COPA</th>
<th>MultiRC</th>
<th>ReCoRD</th>
<th>RTE</th>
<th>WIC</th>
<th>WSC</th>
<th>AX-b</th>
<th>AX-q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ERINIE Team - Baidu</td>
<td>ERINIE 3.0</td>
<td></td>
<td>90.6</td>
<td>91.0</td>
<td>98.6/99.2</td>
<td>97.4</td>
<td>88.6/632</td>
<td>94.7/94.2</td>
<td>92.6</td>
<td>77.4</td>
<td>97.3</td>
<td>68.6</td>
<td>92.7/94.7</td>
</tr>
<tr>
<td>2</td>
<td>Zirui Wang</td>
<td>TS + Meena, Single Model (Meena Team - Google Brain)</td>
<td></td>
<td>90.4</td>
<td>91.4</td>
<td>95.8/97.6</td>
<td>98.0</td>
<td>88.3/630</td>
<td>94.2/93.5</td>
<td>93.8</td>
<td>77.9</td>
<td>96.6</td>
<td>69.1</td>
<td>92.7/91.9</td>
</tr>
<tr>
<td>3</td>
<td>DeBERTa Team - Microsoft</td>
<td>DeBERTa / TuringNLUv4</td>
<td></td>
<td>90.3</td>
<td>90.4</td>
<td>95.7/97.6</td>
<td>98.4</td>
<td>88.2/639</td>
<td>94.5/94.1</td>
<td>93.2</td>
<td>77.5</td>
<td>95.0</td>
<td>66.7</td>
<td>93.3/93.8</td>
</tr>
<tr>
<td>4</td>
<td>SuperGLUE Human Baselines</td>
<td>SuperGLUE Human Baselines</td>
<td></td>
<td>89.8</td>
<td>89.0</td>
<td>95.8/98.9</td>
<td>100.0</td>
<td>81.8/919</td>
<td>91.7/91.3</td>
<td>93.6</td>
<td>80.0</td>
<td>100.0</td>
<td>76.6</td>
<td>99.3/99.7</td>
</tr>
<tr>
<td>5</td>
<td>TS Team - Google</td>
<td>TS</td>
<td></td>
<td>89.3</td>
<td>91.2</td>
<td>93.9/96.8</td>
<td>94.8</td>
<td>88.1/633</td>
<td>94.1/93.4</td>
<td>92.5</td>
<td>76.9</td>
<td>93.8</td>
<td>65.6</td>
<td>92.7/91.9</td>
</tr>
<tr>
<td>6</td>
<td>Huawei Noahs Ark Lab</td>
<td>NEZHA-Plus</td>
<td></td>
<td>86.7</td>
<td>87.8</td>
<td>94.4/96.0</td>
<td>93.6</td>
<td>84.6/55.1</td>
<td>90.1/98.9</td>
<td>89.1</td>
<td>74.6</td>
<td>93.2</td>
<td>58.0</td>
<td>87.1/74.4</td>
</tr>
<tr>
<td>7</td>
<td>Alibaba PAIJBNU</td>
<td>PAI Albert</td>
<td></td>
<td>86.1</td>
<td>88.1</td>
<td>92.4/96.4</td>
<td>91.8</td>
<td>84.6/54.7</td>
<td>89.0/88.3</td>
<td>88.8</td>
<td>74.1</td>
<td>93.2</td>
<td>75.6</td>
<td>98.3/99.2</td>
</tr>
<tr>
<td>8</td>
<td>Infosys : DAWN : AI Research</td>
<td>RoBERTa-CETS</td>
<td></td>
<td>86.0</td>
<td>88.5</td>
<td>93.2/95.2</td>
<td>91.2</td>
<td>86.4/58.2</td>
<td>89.0/89.3</td>
<td>89.9</td>
<td>72.9</td>
<td>89.0</td>
<td>61.8</td>
<td>88.9/81.5</td>
</tr>
<tr>
<td>9</td>
<td>Tencent Jarvis Lab</td>
<td>RoBERTa (ensemble)</td>
<td></td>
<td>85.9</td>
<td>88.2</td>
<td>92.5/95.6</td>
<td>90.8</td>
<td>84.4/53.4</td>
<td>91.5/91.0</td>
<td>87.9</td>
<td>74.1</td>
<td>91.8</td>
<td>57.6</td>
<td>88.9/75.6</td>
</tr>
<tr>
<td>10</td>
<td>Zhejiang Technology</td>
<td>RoBERTa-m6-adv</td>
<td></td>
<td>85.7</td>
<td>87.1</td>
<td>92.4/95.6</td>
<td>91.2</td>
<td>85.1/54.3</td>
<td>91.7/91.3</td>
<td>88.1</td>
<td>72.1</td>
<td>91.8</td>
<td>58.5</td>
<td>91.0/78.1</td>
</tr>
<tr>
<td>11</td>
<td>Facebook AI</td>
<td>RoBERTa</td>
<td></td>
<td>84.6</td>
<td>87.1</td>
<td>90.5/95.2</td>
<td>90.6</td>
<td>84.4/52.5</td>
<td>90.6/90.0</td>
<td>88.2</td>
<td>69.9</td>
<td>89.0</td>
<td>57.9</td>
<td>91.0/78.1</td>
</tr>
<tr>
<td>12</td>
<td>Anvar Sharafudinov</td>
<td>AllLabs Team, Transformers</td>
<td></td>
<td>82.6</td>
<td>88.1</td>
<td>91.6/94.8</td>
<td>86.8</td>
<td>85.1/54.7</td>
<td>82.8/79.8</td>
<td>88.9</td>
<td>74.1</td>
<td>78.8</td>
<td>100.0/100.0</td>
<td>100.0/100.0</td>
</tr>
<tr>
<td>13</td>
<td>Rakesh Radhakrishnan Menon</td>
<td>ADAPET (ALBERT) - few-shot</td>
<td></td>
<td>85.4</td>
<td>76.2/35.7</td>
<td>86.1/85.5</td>
<td>75.0</td>
<td>53.5</td>
<td>85.6</td>
<td>-0.4</td>
<td>100.0/50.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SQuAD benchmark

SQuAD 2.0
The Stanford Question Answering Dataset

What is SQuAD?

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Human Performance Stanford University (Rajpurkar &amp; Jia et al. ’18)</th>
<th>86.831</th>
<th>89.452</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IE-Net (ensemble) RICOH_SRCB_DML</td>
<td>90.939</td>
<td>93.214</td>
</tr>
<tr>
<td>2</td>
<td>FFPNet (ensemble) Ant Service Intelligence Team</td>
<td>90.871</td>
<td>93.183</td>
</tr>
<tr>
<td>3</td>
<td>IE-NetV2 (ensemble) RICOH_SRCB_DML</td>
<td>90.860</td>
<td>93.100</td>
</tr>
<tr>
<td>4</td>
<td>SA-Net on Albert (ensemble) QIANXIN</td>
<td>90.724</td>
<td>93.011</td>
</tr>
<tr>
<td>5</td>
<td>SA-Net V2 (ensemble) QIANXIN</td>
<td>90.679</td>
<td>92.948</td>
</tr>
</tbody>
</table>

SQuAD 1.1. The previous version of the SQuAD dataset.
## RACE Reading Comprehension Dataset

The RACE dataset is a large-scale RoActing Comprehension dataset collected from English Examinations that are created for middle school and high school students.

### Leaderboard

<table>
<thead>
<tr>
<th>Model</th>
<th>Report Time</th>
<th>Institute</th>
<th>RACE</th>
<th>RACE-M</th>
<th>RACE-H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Reading Performance</td>
<td>Apr 15, 2017</td>
<td>CMU</td>
<td>94.5</td>
<td>95.4</td>
<td>94.2</td>
</tr>
<tr>
<td>Amazon Mechanical Turk</td>
<td>Apr 15, 2017</td>
<td>CMU</td>
<td>73.3</td>
<td>85.1</td>
<td>69.4</td>
</tr>
<tr>
<td>ALBERT-SingleChoice + transfer learning (ensemble)</td>
<td>Nov 06, 2020</td>
<td>Tencent Cloud Xiaowei &amp; Tencent Cloud Ti-ONE</td>
<td>91.4</td>
<td>93.6</td>
<td>90.5</td>
</tr>
<tr>
<td>Megatron-BERT (ensemble)</td>
<td>Mar 13, 2020</td>
<td>NVIDIA Research</td>
<td>90.9</td>
<td>93.1</td>
<td>90.0</td>
</tr>
<tr>
<td>ALBERT-SingleChoice + transfer learning</td>
<td>Nov 06, 2020</td>
<td>Tencent Cloud Xiaowei &amp; Tencent Cloud Ti-ONE</td>
<td>90.7</td>
<td>92.8</td>
<td>89.8</td>
</tr>
<tr>
<td>ALBERT + DUMA (ensemble)</td>
<td>Mar 18, 2020</td>
<td>SJTU &amp; Huawei Noah's Ark Lab</td>
<td>89.8</td>
<td>92.6</td>
<td>88.7</td>
</tr>
<tr>
<td>Megatron-BERT</td>
<td>Mar 13, 2020</td>
<td>NVIDIA Research</td>
<td>89.5</td>
<td>91.8</td>
<td>88.6</td>
</tr>
<tr>
<td>ALBERT (ensemble)</td>
<td>Sep 26, 2019</td>
<td>Google Research &amp; TTIC</td>
<td>89.4</td>
<td>91.2</td>
<td>88.6</td>
</tr>
<tr>
<td>UnifiedQA</td>
<td>May 02, 2020</td>
<td>AI2 &amp; UW</td>
<td>89.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALBERT + DUMA</td>
<td>Feb 08, 2020</td>
<td>SJTU &amp; Huawei Noah's Ark Lab</td>
<td>88.0</td>
<td>90.9</td>
<td>86.7</td>
</tr>
<tr>
<td>T5</td>
<td>May 02, 2020</td>
<td>Google</td>
<td>87.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALBERT</td>
<td>Sep 26, 2019</td>
<td>Google Research &amp; TTIC</td>
<td>86.5</td>
<td>89.0</td>
<td>85.5</td>
</tr>
<tr>
<td>RoBERTa + MMM</td>
<td>Oct 01, 2019</td>
<td>MIT &amp; Amazon Alexa AI</td>
<td>85.0</td>
<td>89.1</td>
<td>83.3</td>
</tr>
<tr>
<td>RoBERTa + MMM</td>
<td>Aug 30, 2019</td>
<td>MIT &amp; Amazon Alexa AI</td>
<td>85.0</td>
<td>89.1</td>
<td>83.3</td>
</tr>
</tbody>
</table>

**Report your results:** If you have new results, please send Qihe (qihe@cs.cmu.edu) or Guokun (guokun@cs.cmu.edu) an email with the link to your paper!
Sam - Text Classification with BERT