# Text embeddings

Ilya Fedorov

Lomonosov Moscow State University

November 5, 2020

# The first successful non-recurrent architecture for machine translation

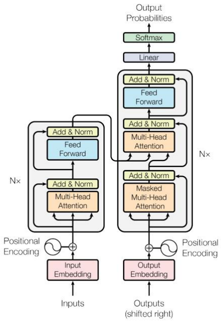
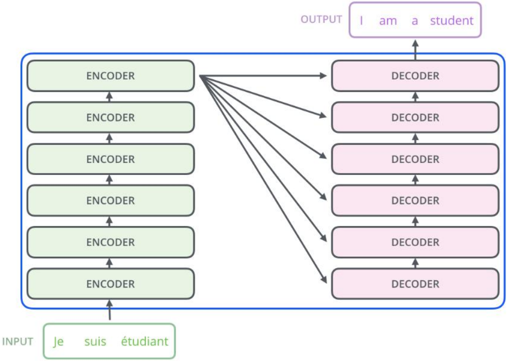



Figure 1: The Transformer - model architecture.

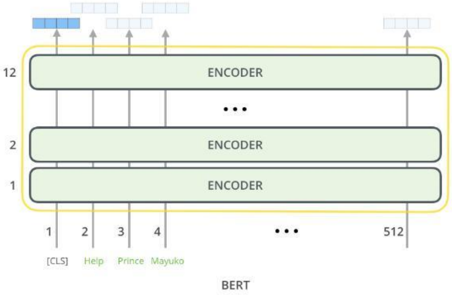
Attention Is All You Need - Vaswani et al. 2017

# Details



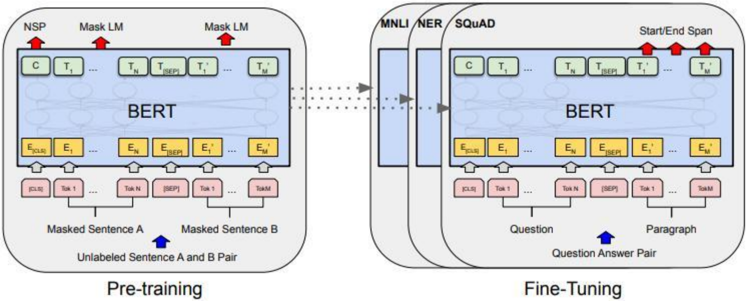

The Illustrated Transformer - Jay Alammar

# Bidirectional Encoder Representations from Transformers




The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) - Jay Alammar

# Illustration from the original paper




BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding - Devlin et al. 2018

# BERT Pre-Training

Two tasks:

1. Masked Language Modeling (MLM)
2. Next Sentence Prediction (NSP)

# Input Format

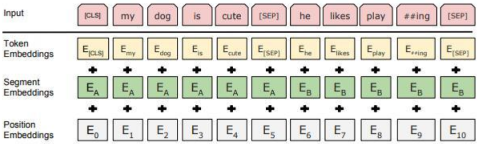



Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

- Token Embeddings - fixed derived from elsewhere word embeddings (e.g. WordPiece, Word2Vec, FastText, Glove etc)
- Segment Embeddings - learnabembeddingsle distinguisher between two sentences in the input
- Position Embeddings - to put the information about the word's position in the sentence

# Fine-Tuning

- Transfer learning
- Plug in the taskspecific inputs and outputs into BERT and finetune all the parameters end-to-end
- Compared to pre-training, fine-tuning is relatively inexpensive (All of the results in the main paper can be replicated in at most 1 hour on a single Cloud TPU, or a few hours on a GPU)

# The General Language Understanding Evaluation GLUE benchmark

| 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** |

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.

| Rank | Name | Model | URL | Score | CoLA | SST-2 | MRPC | STS-B | QQP | MNLI-m | MNLI-mm | QNLI | RTE | WNLI | AX |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | HFL iFLYTEK | MacALBERT + DKM | | 90.7 | 74.8 | 97.0 | 94.5/92.6 | 92.8/92.6 | 74.7/90.6 | 91.3 | 91.1 | 97.8 | 92.0 | 94.5 | 52.6 |
| 2 | Alibaba DAMO NLP | StructBERT + TAPT | | 90.6 | 75.3 | 97.3 | 93.9/91.9 | 93.2/92.7 | 74.8/91.0 | 90.9 | 90.7 | 97.4 | 91.2 | 94.5 | 49.1 |
| 3 | PING-AN Omni-Sinitic | ALBERT + DAAF + NAS | | 90.6 | 73.5 | 97.2 | 94.0/92.0 | 93.0/92.4 | 76.1/91.0 | 91.6 | 91.3 | 97.5 | 91.7 | 94.5 | 51.2 |
| 4 | ERNIE Team - Baidu | ERNIE | | 90.4 | 74.4 | 97.5 | 93.5/91.4 | 93.0/92.6 | 75.2/90.9 | 91.4 | 91.0 | 96.6 | 90.9 | 94.5 | 51.7 |
| 5 | T5 Team - Google | T5 | | 90.3 | 71.6 | 97.5 | 92.8/90.4 | 93.1/92.8 | 75.1/90.6 | 92.2 | 91.9 | 96.9 | 92.8 | 94.5 | 53.1 |
| 6 | Microsoft D365 AI & MSR AI & GATECH | 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 |
| 7 | Zihang Dai | Funnel-Transformer (Ensemble B10-10-10H1024) | | 89.7 | 70.5 | 97.5 | 93.4/91.2 | 92.6/92.3 | 75.4/90.7 | 91.4 | 91.1 | 95.8 | 90.0 | 94.5 | 51.6 |
| 8 | ELECTRA Team | ELECTRA-Large + Standard Tricks | | 89.4 | 71.7 | 97.1 | 93.1/90.7 | 92.9/92.5 | 75.6/90.8 | 91.3 | 90.8 | 95.8 | 89.8 | 91.8 | 50.7 |

https://gluebenchmark.com/leaderboard

# Conclusion

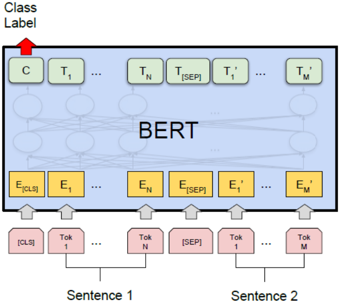
BERT

- Bidirectional model
- Can be pre-trained on a huge amount of unlabeled data
- Can be fine-tuned for the specific task and reach SOTA results
- There exists a lot of BERT modifications: ALBERT, RoBERTa, DistilBERT etc

# Semantic Textual Similarity

Semantic textual similarity deals with determining how similar two pieces of texts are. This can take the form of assigning a score from 1 to 5 (or be continuous in range [0, 1]). Related tasks are paraphrase or duplicate identification.

# Semantic Textual Similarity




(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

The original BERT can be used for that task, but...

# Computational overheads

… finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations ( 65 hours with modern V100) with BERT

What is the solution? The answer is - sentence embeddings.

# Sentence-BERT

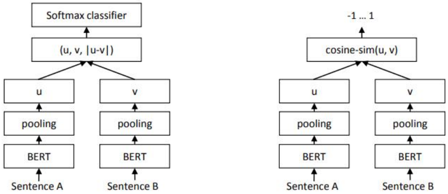



Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks - Nils Reimers and Iryna Gurevych, 2019

# Pooling strategies

- Mean
- Embedding of CLS token
- Max-Over-Time pooling

# Objective functions

- Classification Objective Function
  $o = softmax(W_t(u, v, |u - v|))$
- Regression Objective Function. The cosine similarity between the two sentence embeddings $u$ and $v$ is computed. We use mean squared-error loss as the objective function

- Triplet Objective Function. Given an anchor sentence $a$, a positive sentence $p$, and a negative sentence $n$, triplet loss tunes the network such that the distance between $a$ and $p$ is smaller than the distance between $a$ and $n$. So we minimize:
  $max(|s(a) - s(p)| - |s(a) - s(n)| + \varepsilon, 0)$

# Training Dataset

- SNLI (Stanford Natural Language Inference)
- Multi-Genre NLI

| Text | Judgments | Hypothesis |
|---|---|---|
| A man inspects the uniform of a figure in some East Asian country. | contradiction C C C C C | The man is sleeping |
| An older and younger man smiling. | neutral N N E N N | Two men are smiling and laughing at the cats playing on the floor. |
| A black race car starts up in front of a crowd of people. | contradiction C C C C C | A man is driving down a lonely road. |
| A soccer game with multiple males playing. | entailment E E E E E | Some men are playing a sport. |
| A smiling costumed woman is holding an umbrella. | neutral N N E C N | A happy woman in a fairy costume holds an umbrella. |

https://nlp.stanford.edu/projects/snli/

# Evaluation

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| Avg. GloVe embeddings | 55.14 | 70.66 | 59.73 | 68.25 | 63.66 | 58.02 | 53.76 | 61.32 |
| Avg. BERT embeddings | 38.78 | 57.98 | 57.98 | 63.15 | 61.06 | 46.35 | 58.40 | 54.81 |
| BERT CLS-vector | 20.16 | 30.01 | 20.09 | 36.88 | 38.08 | 16.50 | 42.63 | 29.19 |
| InferSent - Glove | 52.86 | 66.75 | 62.15 | 72.77 | 66.87 | 68.03 | 65.65 | 65.01 |
| Universal Sentence Encoder | 64.49 | 67.80 | 64.61 | 76.83 | 73.18 | 74.92 | **76.69** | 71.22 |
| SBERT-NLI-base | 70.97 | 76.53 | 73.19 | 79.09 | 74.30 | 77.03 | 72.91 | 74.89 |
| SBERT-NLI-large | 72.27 | **78.46** | **74.90** | 80.99 | 76.25 | **79.23** | 73.75 | 76.55 |
| SRoBERTa-NLI-base | 71.54 | 72.49 | 70.80 | 78.74 | 73.69 | 77.77 | 74.46 | 74.21 |
| SRoBERTa-NLI-large | **74.53** | 77.00 | 73.18 | **81.85** | **76.82** | 79.10 | 74.29 | **76.68** |

Table 1: Spearman rank correlation $\rho$ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as $\rho \times 100$. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

More of them in the original paper…

# Open source and easy to use

First download a pretrained model.

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('distilbert-base-nli-mean-tokens')
```

Then provide some sentences to the model.

```
sentences = ['This framework generates embeddings for each input sentence',
    'Sentences are passed as a list of string.',
    'The quick brown fox jumps over the lazy dog.']
sentence_embeddings = model.encode(sentences)
```

https://github.com/UKPLab/sentence-transformers