# Final Project Presentation: GLUE Benchmark

Tristin Johnson, Robert Hilly, Divya Parmar

DATS 6312 - Natural Language Processing

December 9th, 2021

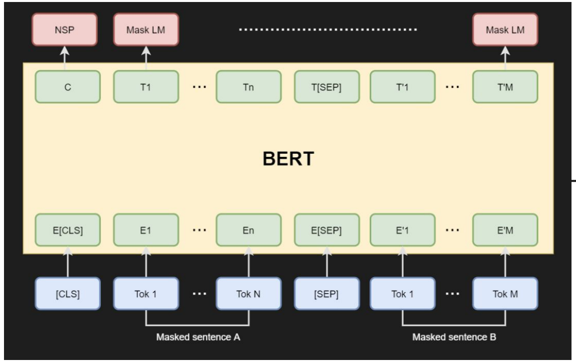
# Introduction

- General Language Understanding Evaluation (GLUE) Benchmark
    - Collection of resources for training, evaluating, and analyzing natural language understanding systems
- 11 different datasets consisting of sentence or sentence-pair tasks
- Wide range of natural language processing tasks
    - Sentiment Analysis, Textual Similarity/Entailment, Question-Answering
- “The ultimate goal of GLUE is to drive research in the development of general and robust natural language understanding systems”

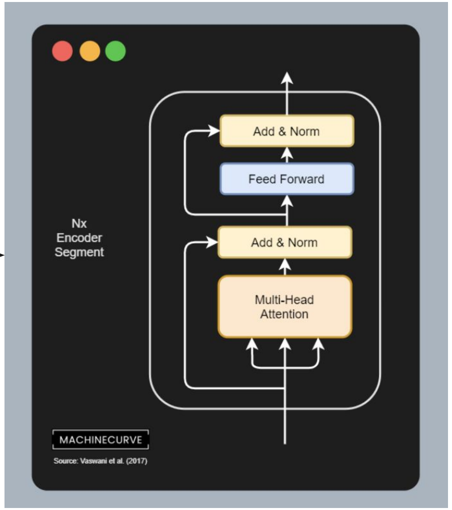
# ALBERT: A Lite BERT

- ALBERTs goal is to reduce the number of trainable parameters
- Two Key Differences:
    a. **Embeddings are factorized:** parameters of the embedding and hidden state are decomposed into 2 smaller matrices
    b. **ALBERT uses cross-layer parameter sharing**: parameters of MHA and Feed Forward Segments are shared

BERT Architecture



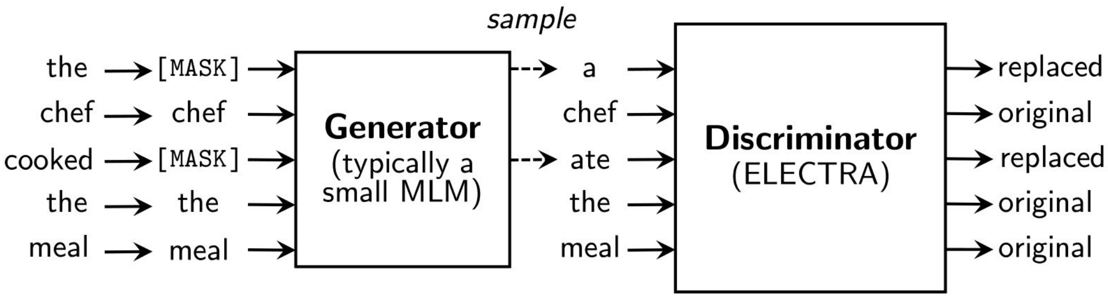

ALBERT Architecture



# ELECTRA

- Inspired by GANs
- Involves two transformer models:
  - **Generator:** replaces tokens and is trained as a Masked Language Model
  - **Discriminator:** tries to identify which tokens were replaced by the generator in the sequence

ELECTRA Architecture



# LSTM

- LSTMs have a good hold over memorizing certain patterns from input
- All relevant information is kept and irrelevant information is discarded
- LSTM Parameters:
  - Vocab Size: 64
  - Neurons: 256
  - Dropout: 0.3
  - Activation Function: Sigmoid

LSTM Architecture

```
SentimentAnalysisLSTM(
  (embedding): Embedding(1501, 64)
  (lstm1): LSTM(64, 256, num_layers=2, batch_first=True)
  (lstm2): LSTM(64, 256, num_layers=2, batch_first=True)
  (dropout): Dropout(p=0.3, inplace=False)
  (linear): Linear(in_features=256, out_features=1, bias=True)
  (act): Sigmoid()
)
```

# The Corpus of Linguistic Acceptability (CoLA)

- Custom Transformer Model Parameters:
  - Batch Size: 32
  - LR: 0.001
  - Epochs: 5
  - Optimizer: AdamW
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Cross Entropy

- LSTM Model Parameters:
  - Batch Size: 32
  - LR: 0.0001
  - Epochs: 20
  - Optimizer: Adam
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Binary Cross Entropy

| CoLA | **ELECTRA** | **ALBERT** | **Custom ELECTRA** | **Custom ALBERT** | **LSTM** |
|---|---|---|---|---|---|
| **MCC (Test)** | 0.5508 | 0.4187 | 0.212 | 0.1981 | 0.11951 |

# The Stanford Sentiment Treebank (SST)

- Custom Transformer Model Parameters:
  - Batch Size: 64
  - LR: 0.001
  - Epochs: 5
  - Optimizer: AdamW
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Binary Cross Entropy

- LSTM Model Parameters:
  - Batch Size: 64
  - LR: 0.0001
  - Epochs: 20
  - Optimizer: Adam
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Binary Cross Entropy

| SST | **ELECTRA** | **ALBERT** | **Custom ELECTRA** | **Custom ALBERT** | **LSTM** |
|---|---|---|---|---|---|
| **Accuracy (Test)** | 90.137% | 86.811% | 51.927% | 50.817% | 77.315% |

# Recognizing Textual Entailment (RTE)

- Custom Transformer Model Parameters:
  - Batch Size: 32
  - LR: 0.001
  - Epochs: 5
  - Optimizer: AdamW
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Cross Entropy

| RTE | **ELECTRA** | **ALBERT** | **Custom ELECTRA** | **Custom ALBERT** |
|---|---|---|---|---|
| **Accuracy (Test)** | 63.176% | 54.151% | 52.708% | 51.818% |

# Winograd Natural Language Inference (WNLI)

- Custom Transformer Model Parameters:
  - Batch Size: 32
  - LR: 0.001
  - Epochs: 5
  - Optimizer: AdamW
  - Scheduler: ReduceLROnPlateau
  - Loss Function: Cross Entropy

| WNLI | ELECTRA | ALBERT | Custom ELECTRA | Custom ALBERT |
|---|---|---|---|---|
| **Accuracy (Test)** | 56.828% | 57.38% | 56.338% | 56.338% |

# DistilBERT

- DistilBERT is a smaller version of BERT.
- Compared to BERT, the architecture of DistilBERT doesn’t have token-type embeddings and pooler that are present in BERT.
- Number of layers is reduced by a factor of 2.
- Requires less money, time, and compute to train compared to BERT.

# Microsoft Research Paraphrase Corpus (MRPC)

- A dataset of 5800 pairs of sentences that were extracted from news sources across the Internet.
- Each pair of sentences were then labeled by humans, indicating whether each pair is a paraphrase/is semantically similar.
- Goal is to train a model that can correctly predict whether each pair of sentences is a paraphrase/is semantically similar.

# Semantic Textual Similarity Benchmark (STS-B)

- A selection of datasets that contain text data from image captions, news headlines, and user forums.
- STS-B contains a pair of sentences and a label indicating the similarity between each sentence on the interval [1, 5].
- Here, the goal is to train a model that can discern the semantic similarity between two sentences.

# Quora Question Pairs (QQP)

- A collection of over 100,000 pairs of questions that have been asked on Quora.
- Similar to MRPC, humans labeled each pair of questions as being a “duplicate” or “not a duplicate”, based on whether the pair of questions have the same underlying meaning.
- Here, the goal is train a model that can interpret whether each pair of questions share the same meaning.

# Results of DistilBERT on MRPC, STS-B, and QQP

DistilBERT Results on Testing Set

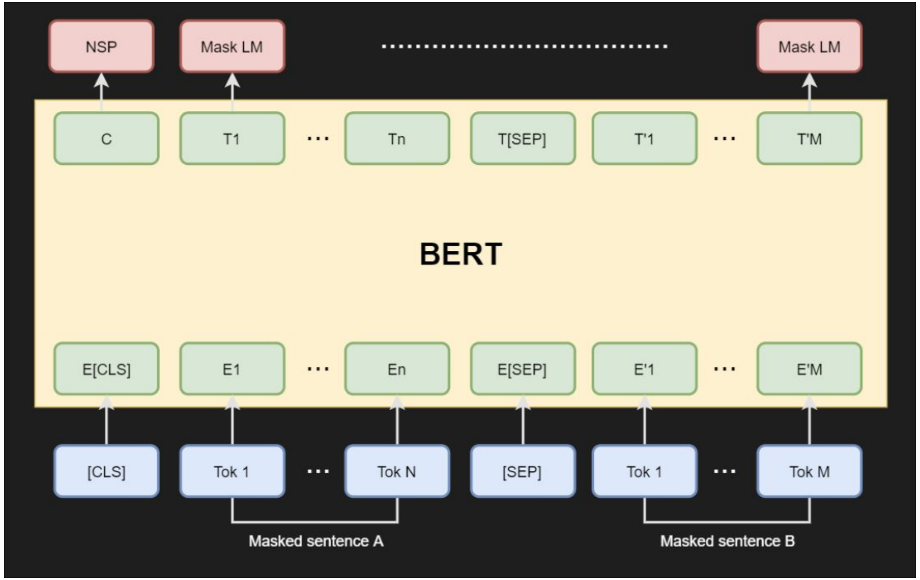
| Task | Accuracy | F1 Score | Pearson's R | Spearman |
|---|---|---|---|---|
| MRPC | 85.5% | 0.900 | NA | NA |
| SST-B | NA | NA | 87.3 | 86.9 |
| QQP | 88.2% | 0.843 | NA | NA |

# BERT

BERT (Bidirectional Encoder Representation for Transformers ) is the application of a bidirectional transformer to language modelling. It reads entire sequences of text at once, as opposed to left to right.

BERT is trained on Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM involves masking ~15 percent of inputs words and having the model predict, and NSP involves feeding an input sentence to predict the following one.

It is a very large transformer, with 110 million trainable parameters in its original form and 340 million trainable parameters in its large form.



# Multi-NLI Matched (MNLI-M)

This dataset contains ~400,000 train observations in the base MNLI and ~9000 validation examples in MNLI-M. Due to computational issues, only 8,000 train observations were used.

Each observation contains a premise ("Your gift is appreciated by each and every student who will benefit from your generosity.") and a hypothesis ("Hundreds of students will benefit from your generosity."). Here, the interest is in predicting if the hypothesis logically flows from the premise.

Each pair is either classified as entailment (it does flow), neutral, or contradiction (premise shows the opposite of hypothesis).

| |
|---|
| **Model: BERT**<br>**Epochs: 2**<br>**LR: 0.0005**<br>**Batch size: 8** |
| Accuracy: 35.45% |

# Multi-NLI Mismatched (MNLI-MisM)

This task is the exact same as the one prior, except it uses mismatched samples to test on. The model is trained on the same dataset as the prior task, so no new training is needed here. For this task, ~9000 new validation samples were used.

| |
|---|
| **Model: BERT**<br>**Epochs: 2**<br>**LR: 0.0005**<br>**Batch size: 8** |
| Accuracy: 35.22% |

# Diagnostics Main

This task contains sentence pairs in which we must check for entailment. Therefore, the model trained in MNLI can be applied to a new validation set of ~1,100 records. For this task, Matthew's Correlation is applied instead of accuracy.

However, the labels provided are incorrect, therefore this task was skipped.

# Question NLI (QNLI)

This dataset is derived from the Stanford Question Answering Dataset (SQuAD). This task contains a question and a sentence taken from a context paragraph, with the intent to see if the sentence answers the question.

This task contains ~100,000 train pairs and ~5,000 validation pairs. However, due to computational scale, only 18,000 training pairs were used.

| |
|---|
| **Model: BERT**<br>**Epochs 2:**<br>**LR: 4.7e-5 (variable by partial epoch)**<br>**Batch size: 4** |
| Accuracy: 79.736% |

# Conclusion

Overall, we were content with the results we received while completing each of the 11 GLUE tasks. From applying the BERT, DISTILBERT, ALBERT, ELECTRA and LSTM models to the datasets, we learned a lot about each of the models along with how each model performs against each other given the same task. Furthermore, it is safe to say that DistilBERT and ELECTRA were our two most promising models as they reported the highest and most consistent levels of performance throughout the GLUE Benchmark.

In the future, we would like to apply even more models to each of the tasks, and start to fine-tune the models that outperform others in specific natural language understanding problems, as this analysis would help with model decisions in the real world. All in all, we look forward to applying the knowledge we gained from this project to future datasets, and now have a better understanding of dealing with natural language processing tasks.

# References


1. “Transformers.” *Hugging Face*. https://huggingface.co/docs/transformers/index
2. “Sentiment Analysis using LSTM – PyTorch”. *Kaggle,* 8 June 2021, https://www.kaggle.com/arunmohan003/sentiment-analysis-using-lstm-pytorch
3. “Text Classification on GLUE”. *Google Colab*. https://colab.research.google.com/github/huggingface/notebooks/blob/master/examples/text_classification.ipynb#scrollTo=HFASsisvIrIb
4. Verma, Dhruv. “Fine-tuning Pre-Trained Transformer Models for Sentence Entailment.” *Towards Data Science,* 14 Jan. 2021. https://towardsdatascience.com/fine-tuning-pre-trained-transformer-models-for-sentence-entailment-d87caf9ec9db
5. Shekhar, Shraddha. “LSTM for Text Classification in Python.” *Analytics Vidhya,* 14 June 2021. https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/
6. Clark, Kevin. “ELECTRA.” *GitHub*. https://github.com/google-research/electra
7. Versloot, Christian. “ALBERT explained: A Lite Bert.” *Machine Curve,* 6 Jan. 2021. https://www.machinecurve.com/index.php/2021/01/06/albert-explained-a-lite-bert/
8. “Sanh et. al” "DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter." Mar. 2020. https://arxiv.org/pdf/1910.01108.pdf
9. “HuggingFace NLP Course” https://huggingface.co/course/chapter1/1?fw=pt
10. Bowman, Sam. “MultiNLI.” *Multinli*, https://cims.nyu.edu/~sbowman/multinli/.
11. Horev, Rani. “Bert Explained: State of the Art Language Model for NLP.” *Medium*, Towards Data Science, 17 Nov. 2018, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.
12. Lutkevich, Ben. “What Is Bert (Language Model) and How Does It Work?” *SearchEnterpriseAI*, TechTarget, 27 Jan. 2020, https://searchenterpriseai.techtarget.com/definition/BERT-language-model.