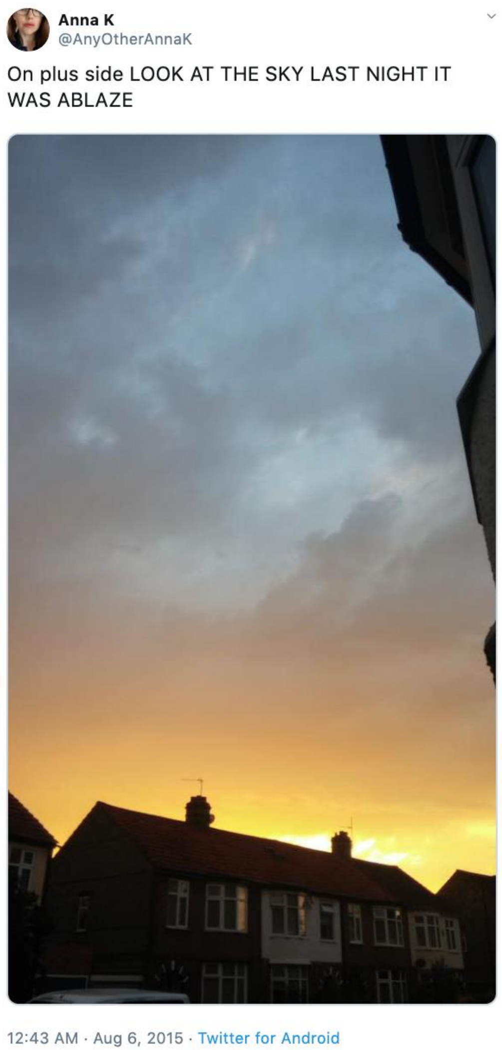
Project: Part II
Natural Language Processing with Disaster Tweets

Pattern Recognition & Machine Learning Laboratory
Ha-Na Jo
Dec. 06, 2022

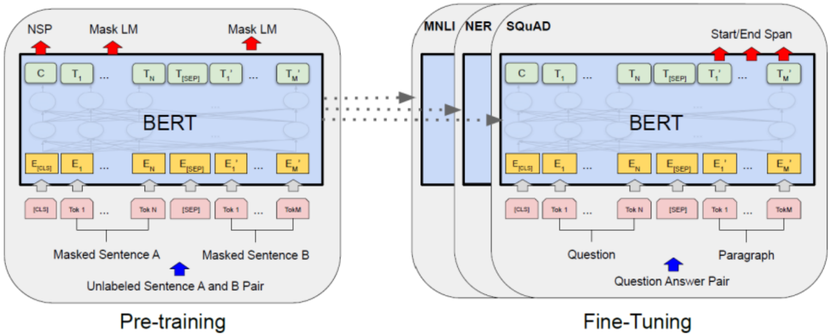
Data Description

- **Goal**
  - ➢ **Predicting whether a given tweet is about a real disaster or not**
- **Necessity**
  - ➢ **Twitter has become an important communication channel in times of emergency**
  - ➢ **The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time**
  - ➢ **But it's not always clear whether a person's words are actually announcing a disaster**
- **Data format**
  - ➢ **Train.csv: the training set**
    - **Including id, text, location, keyword, and target**
  - ➢ **Test.csv: the test set**
    - **Including id, text, location, keyword**
  - ➢ **Sample_submission.csv: a sample submission file in the correct format**

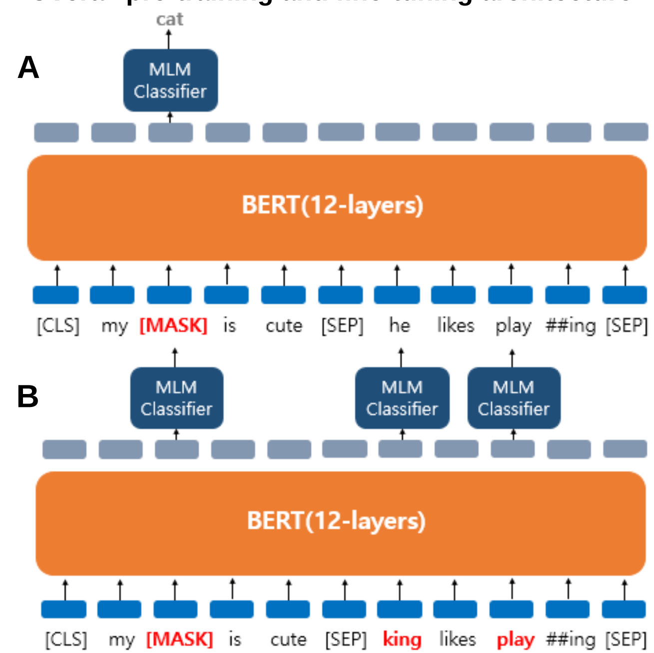


Baseline Model 1

- **BERT**
  - **Goal**
    - **Natural language processing**
  - **Architecture**
    - **Transformer encoder 12 or 24 layers**
      - **Word embedding with context**
    - **Pre-training**
      - **Using massive data without labels**
      - **Masked language model (MLM)**
        - » **Text random masking**
      - **Next sentence prediction (NSP)**
        - » **Train with randomly concatenated sentences**
    - **Fine-tuning**
      - **Using additional tasks with labels**
      - **Single text classification**
        - » **ex) emotion classification**
      - **Tagging**
      - **Question answering**



Overall pre-training and fine-tuning architecture



Example of pre-training (A) MLM (B) NSP

Results 1

- **Implementation environment**
  - Hardware
    - 64GB of RAM
    - GPU with 16GB of RAM
  - Software
    - Anaconda environment (local)
    - Python: 3.9.7
- **Model**
  - Pre-trained BERT
    - Epoch: 3
- **Result**
  - Learning time: 7m
  - 95.21 % accuracy for the training (+ 40.41 %)
    - Baseline model: 54.8 %
  - 83.39 % accuracy for the validation (+ 27.29 %)
    - Baseline model: 56.1 %

```
[9]  ✓  7m 25.8s                                                    Python
...  Epoch 1/3
     381/381 [==============================] - 164s 374ms/step - loss:
     0.4309 - accuracy: 0.8181 - val_loss: 0.3688 - val_accuracy: 0.8418
     Epoch 2/3
     381/381 [==============================] - 144s 377ms/step - loss:
     0.2624 - accuracy: 0.8980 - val_loss: 0.3966 - val_accuracy: 0.8404
     Epoch 3/3
     381/381 [==============================] - 138s 363ms/step - loss:
     0.1292 - accuracy: 0.9521 - val_loss: 0.4559 - val_accuracy: 0.8339
```

Baseline Model 2

- **DistilBERT**
  - **Goal**
    - **Smaller parameter counts**
    - **Faster speed**
  - **Architecture**
    - **Student model**
      - **Remove NSP training**
      - **Remove token-type embedding**
      - **Reduced to Layer 1/2**
    - **Distillation**
      - **Soft target loss (L_{ce})**
        - » **Comparison between teacher and student model's soft target**
      - **Hard target loss (L_{mlm})**
        - » **Masked language model's loss**
      - **Cosine embedding loss (L_{cos})**
        - » **Aligning teacher and student model's hidden vector**
  - **Results**
    - **40% size, 160% speed, 97% capability**

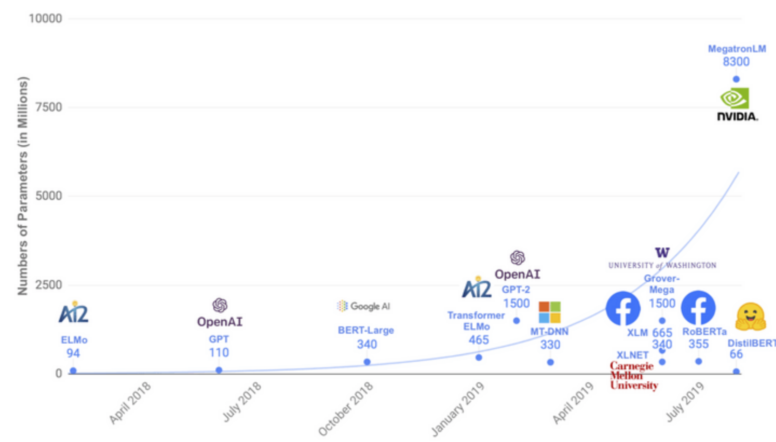


Figure 1: **Parameter counts of several recently released pretrained language models.**

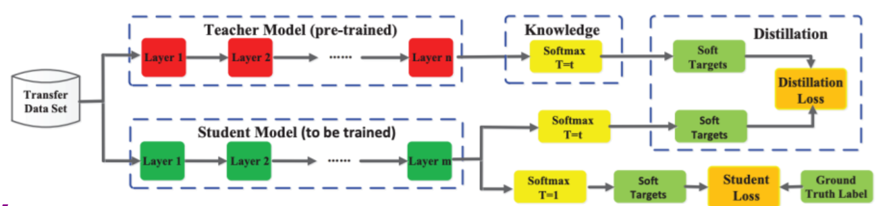


Fig. 5 The specific architecture of the benchmark knowledge distillation (Hinton et al., 2015).

J. Devlin *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Computation and Language*, 2018.

Results 2

- **Implementation environment**
  - Hardware
    - 64GB of RAM
    - GPU with 16GB of RAM
  - Software
    - Anaconda environment (local)
    - Python: 3.9.7
- **Model**
  - Pre-trained DistilBERT
    - Epoch: 3
- **Result**
  - Learning time: 1m 30s
  - 90.39 % accuracy for the training (+ 35.59 %)
    - Baseline model: 54.8 %
  - 85.04 % accuracy for the validation (+ 28.94 %)
    - Baseline model: 56.1 %
    - BERT: 83.39 %

```
Epoch 1/3
452/452 [==============================] - ETA: 0s - loss: 0.5037 - accuracy: 0.7724
Epoch 1: val_loss improved from inf to 0.36113, saving model to mycheckpoint
452/452 [==============================] - 33s 63ms/step - loss: 0.5037 - accuracy: 0.7724 - val_loss: 0.3611 - val_accuracy: 0.8583
Epoch 2/3
452/452 [==============================] - ETA: 0s - loss: 0.3502 - accuracy: 0.8555
Epoch 2: val_loss did not improve from 0.36113
452/452 [==============================] - 26s 58ms/step - loss: 0.3502 - accuracy: 0.8555 - val_loss: 0.3755 - val_accuracy: 0.8504
Epoch 3/3
451/452 [============================>.] - ETA: 0s - loss: 0.2550 - accuracy: 0.9041
Epoch 3: val_loss did not improve from 0.36113
452/452 [==============================] - 26s 57ms/step - loss: 0.2553 - accuracy: 0.9039 - val_loss: 0.4381 - val_accuracy: 0.8504
```