# Method

In this section, we give a brief overview of the fine-tuning BERT (Devlin et al., 2019) and feature extraction

## Fine-tuning BERT

BERT for sequence classification takes one segments (sequences of tokens) as input. The segment is re-presented as a single input sequence to BERT with two special tokens (one from the beginning and one at the end): . need to be less or equal to which is a parameter that controls the maximum sequence length during training. At the output, the representation is fed into an output layer for the sentiment analysis.

Adam (Kingma and Ba, 2015) is the optimization of BERT with the following parameters: and . BERT trains with 0.1 dropout rate on all layers and attention weights with activation function as GELU.

## Feature Extraction

TMUNLP or something like that

# Performance Evaluation

## Experimental Setting

We use *bert-base-uncased* as the pre-trained model for both BERT model and BERT tokenizer from *transformer* package. We re-implement BERT with RoBERTa setting (Liu et al., 2019). We primarily follow the original BERT optimization hyper-parameters given in Section 3, except is set as 0.98 to improve stability when training.

The maximum sequence length is 512 tokens where padding or truncating at the end of segment. We run BERT model in a single GPU RTX 2080 TI. Since it only have 11GB GPU RAM, 6 sequences is trained per batch  
As an out, the last hidden layer of which represent a whole sentence will concatenate with the output layer of *feature extraction* step.

We use *tmunlp*package to extract top 50 features affected to each label.

## Data

## Results and Discussion

# Ref

Kingma and Ba, 2015

Devlin et al., 2019

Liu et al., 2019