Google Explore CSR @ Perth

Explainable Sentiment Analysis

Siwen Luo
School of Computer Science
The University of Western Australia
siwen.luo@uwa..edu.au

Natural Language Processing

- ✔ A subfield of linguistics, computer science, and artificial intelligence concerned with the **interactions between machine and human language.**
- ✔ The result is a machine **capable of "understanding" the contents of documents**, including the **contextual nuances** of the language within them.
- ✔ Including Speech Recognition, N.L.Understanding (e.g. QA, **text classification**), and N.L.Generation (e.g. chatbots, reports).

"I miss you" doesn't equal "Let's get back together".

Source : https://en.wikipedia.org/wiki/Natural_language_processing

Text Classification

✔ Goal : to automatically classify the text documents into one or more defined categories. (label = **classes** or categories, data = **text**)

✔ Examples

- Understanding audience **sentiment** from social media
- Detection of spam and non-spam emails
- Auto tagging of customer queries
- Categorization of news articles into defined topics

Sentiment Analysis

= The Detection of Attitudes

*"Sentiment analysis is the operation of **understanding the intent or emotion behind a given piece of text**. It is part of text classification, but it is useful for extracting structured information"*

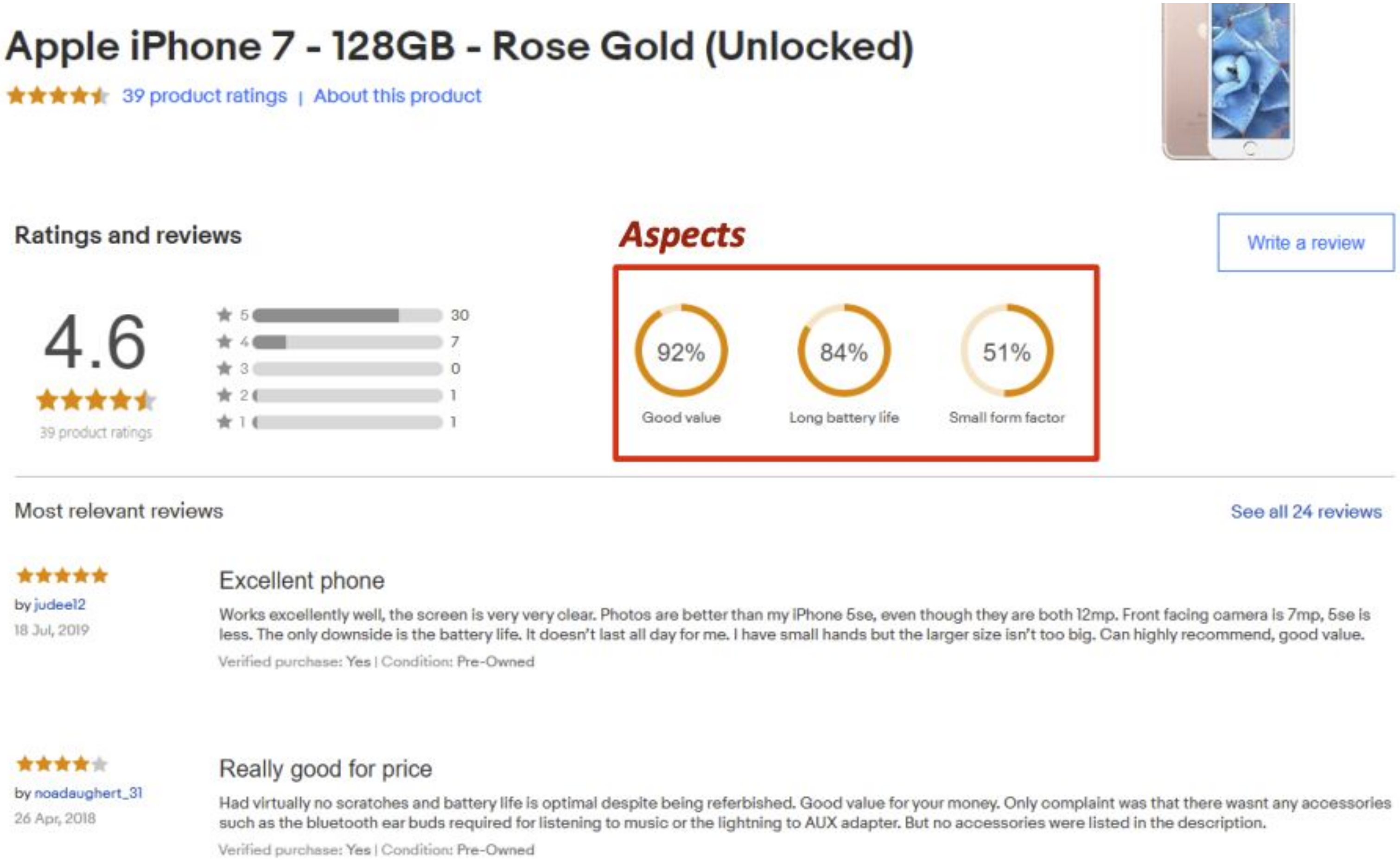
Different Names of a 'Sentiment Analysis'

- *Opinion extraction*
- *Opinion mining*
- *Sentiment mining*
- *Subjectivity analysis*

Knowing Why (= Explainability) is Important in Sentiment Analysis.

Source : University of Sydney COMP5046 NLP Lecture note (by Dr. Caren Han)

Sentiment Analysis Examples

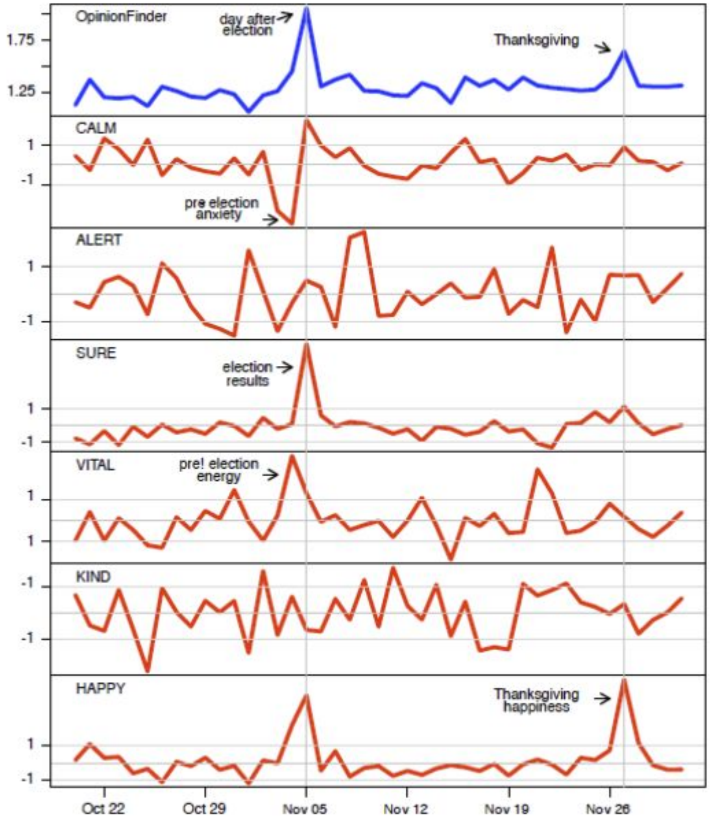


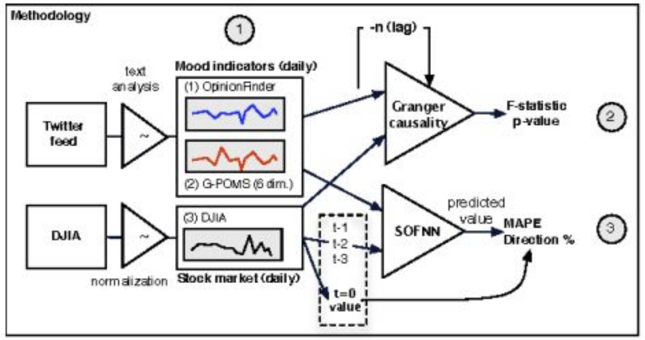
- *Is this review positive or negative?*
- *What do people think about the new phone?*

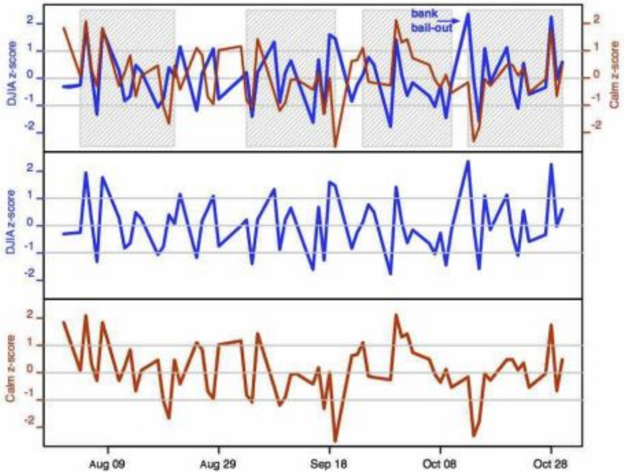
Source : University of Sydney COMP5046 NLP Lecture note (by Dr. Caren Han)

Sentiment Analysis Examples

Twitter mood predicts the stock market (Bollen et al. 2011)







- *How is the changes of investors mood in Twitter?*
- *Predict market trends from sentiment*

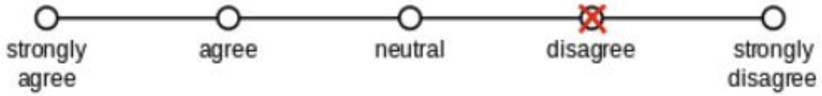
Source : University of Sydney COMP5046 NLP Lecture note (by Dr. Caren Han)

Sentiment Analysis Tasks

- *Basic Task: Is the attitude of this text positive or negative?*

- *More complex task: Rank the attitude of this text from 1 to 5*

Likert Scale (1 to 5)



Fashion Review Dataset

- *Advanced task: Detect the target, source, or complex* ***attitude types***

Source : University of Sydney COMP5046 NLP Lecture note (by Dr. Caren Han)

Sentiment Classification

- ✔ Typically, people have used **Naïve Bayes** or **Support Vector Machines (SVM)** in the past. [Mohammad et al. 2013]
- ✔ **Artificial Neural Nets** are also becoming more popular now. [Nogueira dos Santos & Gatti, 2014]
- ✔ Although these **Neural Nets** show a quantitative improvement over previous approaches, they are **not often** accompanied with a thorough analysis of the **qualitative differences**. [Barnes et al. 2019]
- ✔ Only fewer applications of **pretrained language models** such as **BERT** have been observed for sentiment classification. [Gao et al. 2019]

Challenges:
- It is difficult to identify explainable reasons of the sentiment prediction from Deep learning Neural Nets or pretrained language models.

Research Aim

- ✔ ***Task : Explainable Text Classification Model for Fashion Review Dataset***

- ✔ *Aim : to discover* ***interpretation of sentiment analysis*** *predicted from deep learning Neural Nets and pretrained models.*

Pretrained Model - BERT

Devlin, Jacob, et al. "**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**." *Proceedings of the 2019 Conference of the NAACL.* 2019. [Paper]

Background

- Pretrained language model : effective for improving many NLP tasks
- Two strategies for downstream tasks:
  - ✔ ***Feature-based** : uses **task-specific architectures** that include the pre-trained representations as additional features (e.g. ELMO).*
  - ✔ ***Fine-tuning:** introduces **minimal task-specific parameters** and is trained on the downstream tasks by simply fine-tuning all pretrained parameters. (e.g. GPT)*
- Both methods use **unidirectional** language models during pretraining to learn general language representations.

Addressed Problems

- **Restriction on the choice of architectures** that can be used during pretraining.
- It could be **harmful for token-level tasks** (e.g. QA): crucial to **incorporate context from both directions**.

Pretrained Model - BERT

BERT is a model, already pre-trained on massive datasets that breaks several records for how well models can handle language-based tasks.

Research Aim

- To improve fine-tuning based approach by proposing BERT: **Bidirectional Encoder Representations from Transformers.**
- Use **Masked Language Model (MLM)** : randomly masks some of the tokens from the input **to predict the original vocabulary id** of the masked word **based only on its context.** → enables a deep bidirectional Transformer.
- Use a **Next Sentence Prediction (NSP)** task : **jointly** pretrains text-pair representations.

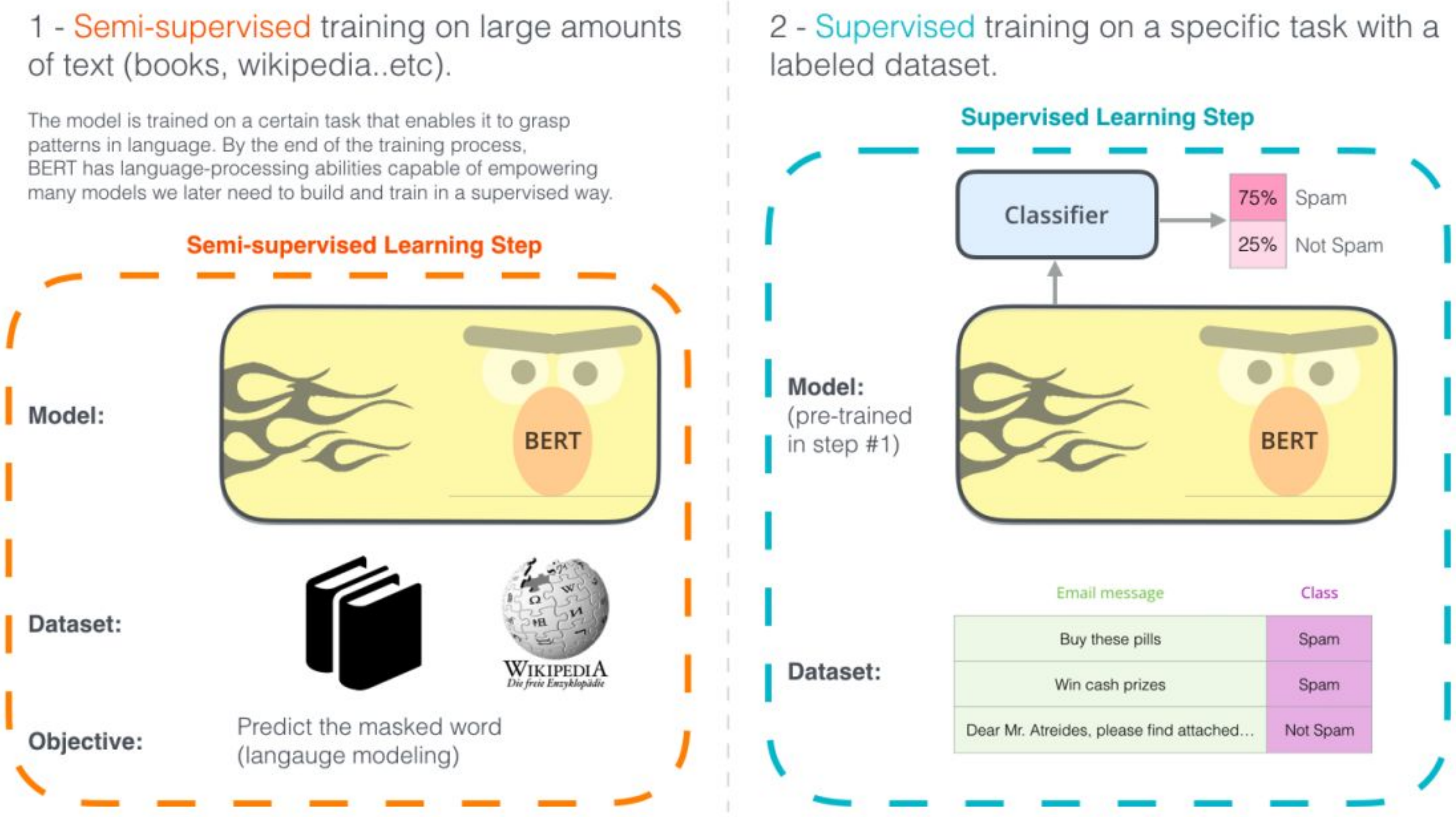
Contribution

- Demonstrate the **importance of bidirectional** pretraining for language representations.
- Achieve **state-of-the-art performance** on a large suite of sentence-level and token-level tasks, outperforming many task-specific architectures. (SOTA for eleven NLP tasks)

Pretrained Model - BERT

Methodology

- Two steps:
  - ✔ ***Pre-training* : trained on unlabeled data** over different pre-training tasks.
  - ✔ ***Fine-tuning*: initialized with the pre-trained parameters,** and all the parameters are fine-tuned using labeled data from the downstream tasks.



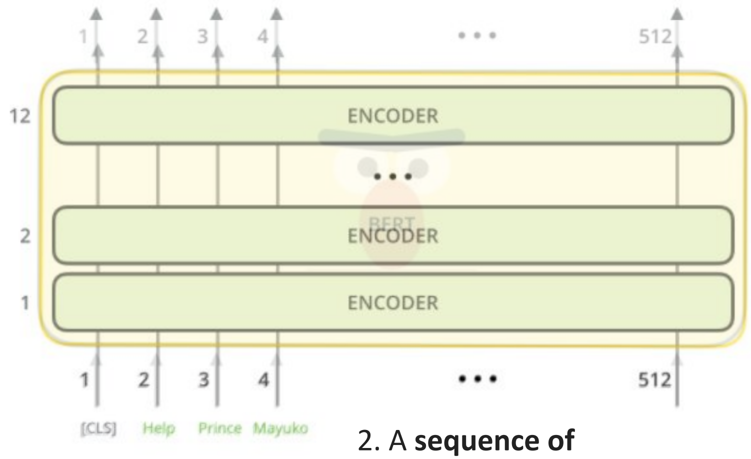
Pretrained Model - BERT

Methodology

- Model Architecture : a bidirectional **Transformer** encoder stack. [Illustrated BERT] [Illustrated Transformer]
  - ✔ ***BERT base*** (**L**ayer=12, **H**idden size = 768, Self-**A**ttention head=12, Total **P**arameters=110M)
  - ✔ ***BERT large*** (L=24, H=1024, A=16, P=340M)

An example of using BERT in a binary classification task

Model Input



1. The first input token is supplied with a special **[CLS]** token

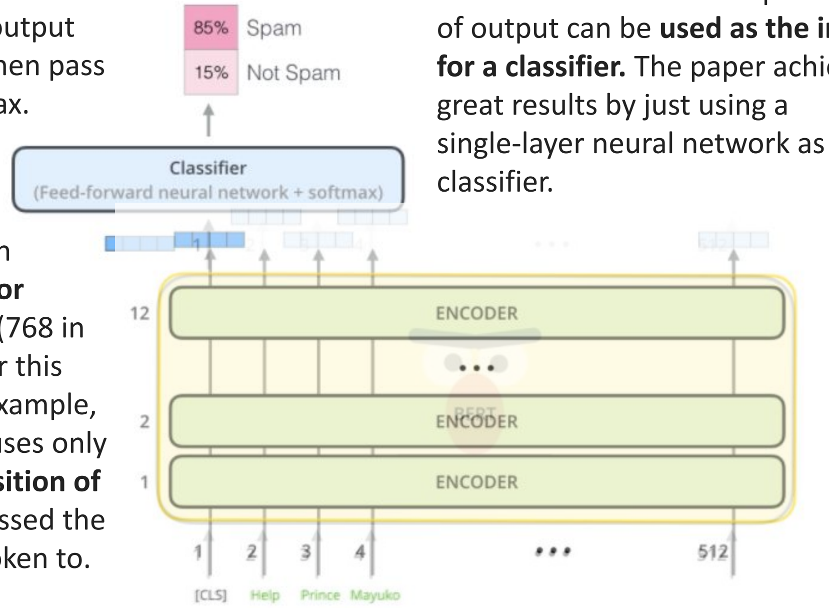
2. A **sequence of words** as input which keep flowing up the stack

3. Each layer applies **self-attention** and passes its results through a **feed-forward network,** and then hands it off to the **next encoder.**

4. In terms of architecture, this has been identical to the Transformer up until this model input.

Model Output

7. For a multiclass classification task, it needs to **tweak the classifier network** to have more output neurons that then pass through softmax.



6. The vector on the first position of output can be **used as the input for a classifier.** The paper achieves great results by just using a single-layer neural network as the classifier.

5. Each position **outputs a vector of *hidden size*** (768 in BERT Base). For this classification example, the model focuses only on the **first position of output** that passed the special **[CLS]** token to.

Pretrained Model - BERT

Methodology

- **Pre-training BERT:**

✔ **Task #1: Masked LM (MLM)*:***

- **Purpose :** to train a deep bidirectional representation instead of unidirectional method.
- **Method** :
  - Mask 15% of all WordPiece tokens in each sequence at random, and then predict those masked tokens. (Bert uses WordPiece embeddings with a 30k token vocabulary.)
  - The final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary.

✔ **Task #2: Next Sentence Prediction (NSP)*:***

- **Purpose :** to train a model that understands sentence relationships.
- **Method** : Use a next sentence prediction task. Specifically, given a pair of sentence A and B (following sentence of A), and predict the binary label on B. (IsNext / NotNext).

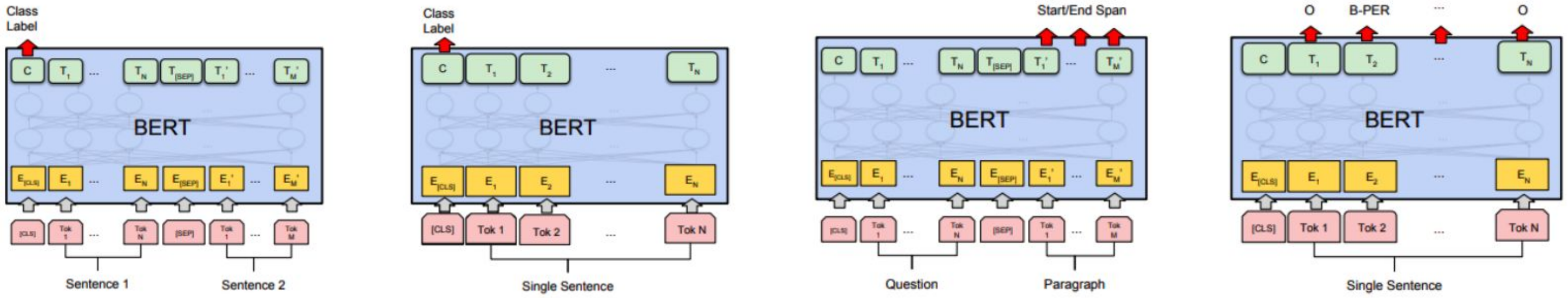
Pretrained Model - BERT

Methodology

✔ **Pre-training data:**

- **Method :** Use the BooksCorpus (800M words) and English Wikipedia (2,500M words) for the pretraining corpus. (A common way of language model pretraining)

- **Fine-training BERT:**
  - **Method** : Simply plug in the task-specific inputs and outputs into BERT and fine-tune all the parameters.
  - **NLP Tasks for experiments** : (e.g.) Sentence Pair Classification, Single Sentence Classification, Question Answering, and Single Sentence Tagging Tasks



(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

Pretrained Model - BERT

Experiments

- **GLUE Test Results**
  - Both BERT base and BERT large outperform.
  - BERT large significantly outperforms BERT base across all tasks, especially those with very little training data.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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**

- **Results on SQuAD 1.1, SQuAD 2.0, and SWAG**
  - SOTA performances.

- **Ablation Studies**

✔ **Effect of Pre-training Tasks**

- **No Next Sentence Prediction (No NSP):** hurts performance significantly on QNLI, MNLI, and SQuAD 1.1.
- **No NSP & No MLM (Mask) & LTR (Left-to-Right) :** performs worse than the MLM model on all tasks.

✔ **Effect of Model Size:** BERT large outperforms BERT base across all four datasets.

✔ **Feature-based Approach with BERT**

- Named Entity Recognition (NER) task on CoNLL2003 : BERT large performs competitively with SOTA model.

Conclusion

- BERT demonstrates empirical improvements on a broad set of NLP tasks using unsupervised pretraining model.

Dataset & Source

Dataset : Women’s E-commerce Clothing Review Dataset
https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews

Base Code:
https://colab.research.google.com/drive/1ptHEmph8rrHBC9GRr058dZAD21-SyBvi?usp=sharing

Colab :
https://colab.research.google.com/drive/1R_Q2EOVN0jpcjeMPZIiUbTQMDKkK8_9b?usp=sharing