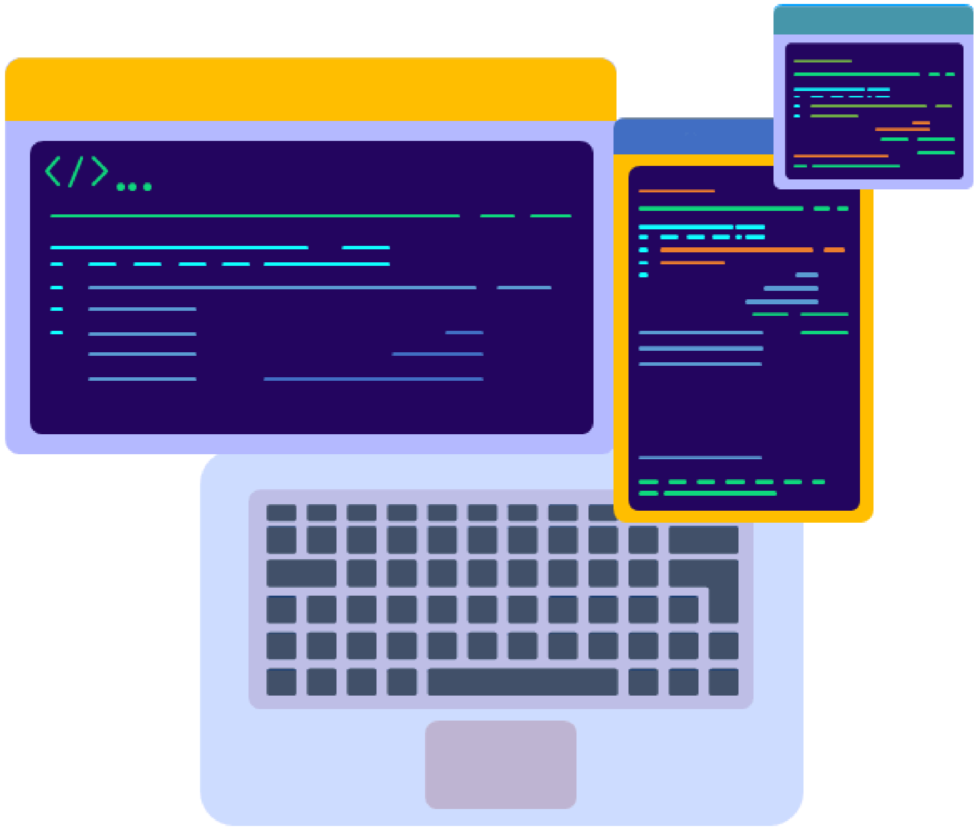
WestlakeNLP

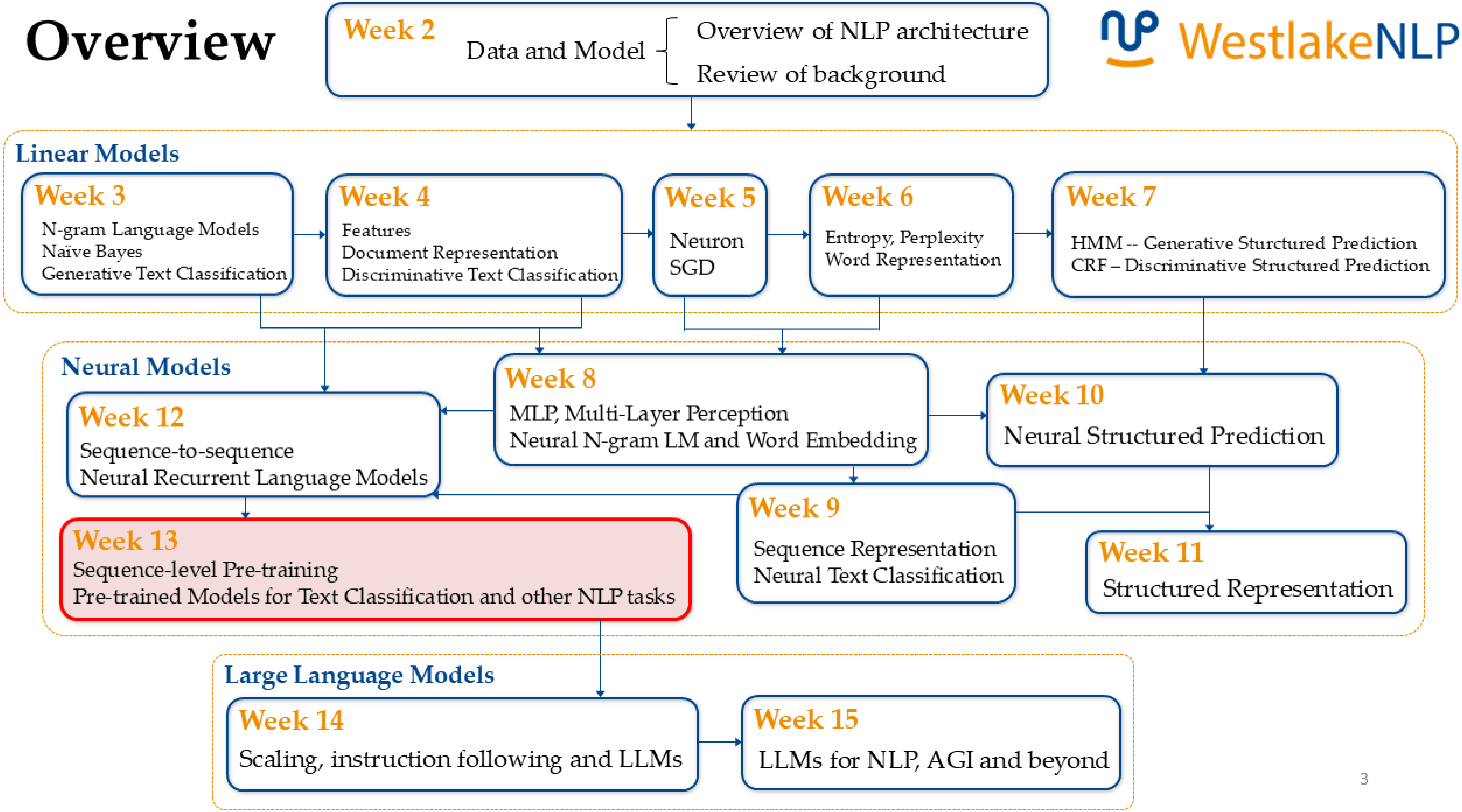
# Natural Language Processing




**Yue Zhang**
**Westlake University**

西湖大學
WESTLAKE UNIVERSITY

# Overview




WestlakeNLP

Week 13

# Transformer Pre-training

# Content

# Content

# Transformer Pre-training

WestlakeNLP

- **ELMo**(Embeddings from Language Models) shows the promise of pretraining a sequence encoder
- The same recurrent LM objective can also train Transformer
- More objectives can be defined to train the **encoder**, **decoder**, or **encoder-decoder** structure
- The use can be beyond contextualized embedding

# GPT and Decoder-only Pre-training

WestlakeNLP

- Use a decoder-only Transformer as a recurrent language model
- Objective: given $W_{1:n} = w_1 w_2 \cdots w_n$

$$L = -\sum_{i=1}^{n} \log P(w_i \mid W_{1:i-1})$$

# GPT and Decoder-only Pre-training

WestlakeNLP

- Use a decoder-only Transformer as a recurrent language model
- Objective: given $W_{1:n} = w_1 w_2 \cdots w_n$

$$L = -\sum_{i=1}^{n} \log P(w_i \mid W_{1:i-1})$$

- Model architecture:

$$\mathbf{X} = [emb(w_0); ...; emb(w_n)] \ \mathbf{P} = [\textsc{PositionEncoding}(0); ...; \textsc{PositionEncoding}(n)]$$

$$\mathbf{H}^{\mathbf{0}} = \mathbf{X} + \mathbf{P}$$

$$\mathbf{H}^k = DecoderLayer(\mathbf{H}^{k-1}), k \in [1, K_d]$$

$$\mathbf{P}(w_i \mid W_{1:i-1}) = Softmax(\mathbf{W}\mathbf{h}_i^k)$$

- Note: No cross attention sublayer! $w_0 = w_{n+1} = < s >$

# GPT and Decoder-only Pre-training

WestlakeNLP

- Uses BPE to obtain subword vocabulary
- Trained on WebText (8M documents, 40GB text)
- Statistics:

| Model | #heads | #layers $K_d$ | hidden size $d_h$ | model size #params |
|---|---|---|---|---|
| GPT-2 | 12 | 12 | 768 | 117M |

# GPT and Decoder-only Pre-training

WestlakeNLP

- Application
  - $\mathbf{H}^{K_d}$ can be used for contextualized embedding
  - GPT gives a new way of usage —— fine-tuning

classification

$$P(c \mid W_{1:n}) = softmax(\mathbf{W}\mathbf{h}_n^{K_d})$$

loss

$$\mathcal{L}^{FT} = -\sum_{(W_i, c_i) \in D} \log P(c_i \mid W_i)$$

The whole set of Transformer parameters are adjusted!

# Content

# BERT: Bidirectional Encoder Representations from Transformer

WestlakeNLP

- Masked Language Model

| | |
|---|---|
| $n$-gram LM | recurrent LM |
| skip-gram LM | masked LM |

- Predict missing word in a sentence

“I went to the ______ for lunch”

$\Downarrow$

(café, canteen, restaurant, bar, …)

- Advantage: Context from both the left and right can be used.

# BERT: Bidirectional Encoder Representations from Transformer

- Use an encoder-only Transformer for masked language model
- Objective: given $W_{1:n} = w_1 w_2 \cdots w_n$, where $\mathcal{M}$ is set of masked words

$$\mathcal{L} = \sum_{i \in \mathcal{M}} -\log P(w_i | W_{1:n})$$

# BERT: Bidirectional Encoder Representations from Transformer

WestlakeNLP

- Model Architecture

$$\mathbf{X} = [emb(w_0); ...; emb(w_n)]$$

$$\mathbf{P} = [\textsc{PositionEncoding}(0); ...;$$
$$\textsc{PositionEncoding}(n)]$$

$$\mathbf{H^0} = \mathbf{X} + \mathbf{P}$$

$$\mathbf{H}^k = DecoderLayer(\mathbf{H}^{k-1}), k \in [1, K_e]$$

$$\mathbf{P}(w_i \mid W_{1:n}) = Softmax(\mathbf{W}\mathbf{h}_i^{K_e} + \mathbf{b})$$

Note: $w_0 = [\text{CLS}]$

# BERT: Bidirectional Encoder Representations from Transformer

- Model Architecture

$$\mathbf{X} = [emb(w_0); ...; emb(w_n)]$$

$$\mathbf{P} = [\textsc{PositionEncoding}(0); ...; \textsc{PositionEncoding}(n)]$$

$$\mathbf{H^0} = \mathbf{X} + \mathbf{P}$$

$$\mathbf{H}^k = DecoderLayer(\mathbf{H}^{k-1}), k \in [1, K_e]$$

$$\mathbf{P}(w_i \mid W_{1:n}) = Softmax(\mathbf{W}\mathbf{h}_i^{K_e} + \mathbf{b})$$

Note: $w_0 = [\text{CLS}]$

- Masking 15% input words
- Test time: no mask —— training-testing inconsistency
  - 10% masked words unmasked, still predict
  - 10% masked words randomly change to a different word

(to prevent model from simply copying unmasked words)

# BERT: Bidirectional Encoder Representations from Transformer

- Additional objective: next sentence prediction

  a sentence pair $W_1 W_2$:

$$[\text{CLS}] w_1^1 w_2^1 \cdots w_{|W_1|}^1 [\text{SEP}] w_1^2 w_2^2 \cdots w_{|W_2|}^2 [\text{SEP}]$$

  Predicts whether $W_2$ is the next sentence in data.

# BERT: Bidirectional Encoder Representations from Transformer

- Additional objective: next sentence prediction

  a sentence pair $W_1 W_2$:

$$[\text{CLS}]w_1^1 w_2^1 \cdots w_{|W_1|}^1 [\text{SEP}] w_1^2 w_2^2 \cdots w_{|W_2|}^2 [\text{SEP}]$$

  Predicts whether $W_2$ is the next sentence in data.

- Model architecture
  - add segment embedding (0/1) to word representation $\mathbf{X} + \mathbf{P}$
  - predicts binary class (next sentence of $W_1$ or not)

$$P(\text{true}|\ W_1 W_2) = softmax(\mathbf{W}'\mathbf{h}_{[\text{CLS}]}^{K_e} + \mathbf{b}')$$

# BERT: Bidirectional Encoder Representations from Transformer

- Use WordPiece to obtain subword vocabulary
  :alternative to BPE, using $\frac{P(w_1 w_2)}{P(w_1)P(w_2)}$ instead of $P(w_1 w_2)$ for merging
  - Trained on BooksCorpus (0.8B words) and English Wikipedia (2.5B words)

- Statistics

| Model | #heads | #layers $K_d$ | hidden size $d_h$ | model size #params |
|---|---|---|---|---|
| BERT$_{\text{BASE}}$ | 12 | 12 | 768 | 110M |
| BERT$_{\text{LARGE}}$ | 24 | 24 | 1024 | 340M |

# BERT: Bidirectional Encoder Representations from Transformer

- Application
  - Follows GPT on fine-tuning
  - Classification:

$$P(c \mid W_{1:n}) = softmax(\mathbf{W}\mathbf{h}_{[\mathrm{CLS}]}^{K_e} + \mathbf{b})$$

  loss

$$\mathcal{L}^{FT} = -\sum_{(W_i, c_i) \in D} \log P(c_i \mid W_i)$$

  - For structured prediction, use the last hidden layer as **H**
  - More tasks later…

# RoBERTa: A Robustly Optimized BERT Pretraining Approach

WestlakeNLP

- Variant of BERT
  - the same architecture as BERT
  - trained with more data ( BookCorpus 16GB, CC News 76G, OpenWebText 38GB, … Total **160GB**)
  - dynamically mask training instances in each batch
  - focus less on next sentence prediction
- Statistics

| Model | #heads | #layers | hidden size | model size |
|---|---|---|---|---|
| $\text{RoBERTa}_{\text{BASE}}$ | 12 | 12 | 768 | 125M |
| $\text{RoBERTa}_{\text{LARGE}}$ | 24 | 24 | 1024 | 355M |

# BART: Bidirectional and Auto-Regressive Transformer

- Use an encoder-decoder Transformer for denoising auto-encoder
- Objective: given a noisy $\mathbf{X}_{1:m}$, predict the original $\mathbf{Y}_{1:m}$

$$\mathcal{L} = -\sum_{i=1}^{m} P(y_i \mid \mathbf{X}_{1:m}, \mathbf{Y}_{<i})$$

# BART: Bidirectional and Auto-Regressive Transformer

- Use an encoder-decoder Transformer for denoising auto-encoder
- Objective: given a noisy $\mathbf{X}_{1:m}$, predict the original $\mathbf{Y}_{1:m}$

$$\mathcal{L} = -\sum_{i=1}^{m} P(y_i \mid \mathbf{X}_{1:m}, \mathbf{Y}_{<i})$$

- Model Architecture
  - Standard Transformer
  - Change ReLU activation to GeLU

$$\mathrm{GeLU}(x) = x \cdot \frac{1}{2}\left[1 + \mathrm{erf}(x/\sqrt{2})\right] \approx 0.5x\left(1 + \tanh\left[\sqrt{2/\pi}(x + 0.044715x^3)\right]\right)$$

# BART: Bidirectional and Auto-Regressive Transformer

WestlakeNLP

- Denoise Tasks

| Task | Input $\rightarrow$ Output |
|---|---|
| Token masking | ABC.DE. $\rightarrow$ A_C._E. |
| Token deletion | ABC.DE. $\rightarrow$ A.C.E |
| Text infilling (Span to mask) | ABC.DE $\rightarrow$ A_.D_E (BC; $\emptyset$) |
| Sentence permutation | ABC.DE. $\rightarrow$ DE.ABC. |
| Document rotation | ABC.DE. $\rightarrow$ C.DE.AB (start from C) |

- Token masking is the most useful.

# BART: Bidirectional and Auto-Regressive Transformer

- Uses BPE to obtain subword vocabulary (same as RoBERTa)
- trained data ( BookCorpus 16GB, CC News 76G, OpenWebText 38GB, … Total **160GB**, same as RoBERTa)
- Model Architecture

  Statistics

| Model | #heads | #encoder layers $K_e$ | #decoder layers $K_d$ | hidden size | model size |
|---|---|---|---|---|---|
| $\text{BART}_{\text{BASE}}$ | 16 | 6 | 6 | 768 | 125M |
| $\text{BART}_{\text{LARGE}}$ | 24 | 12 | 12 | 1024 | 355M |

# BART: Bidirectional and Auto-Regressive Transformer

- Application
  - Follows GPT and BERT on fine-tuning
    - Classification:
      use $h^{\text{dec}}_{<s>}$ for prediction
  - Structured prediction: use $\mathbf{H}^{\text{dec}}$ or $\mathbf{H}^{\text{dec}} \oplus \mathbf{H}^{\text{enc}}$ for hidden
  - Directly fine-tuned on sequence-to-sequence tasks

# Content

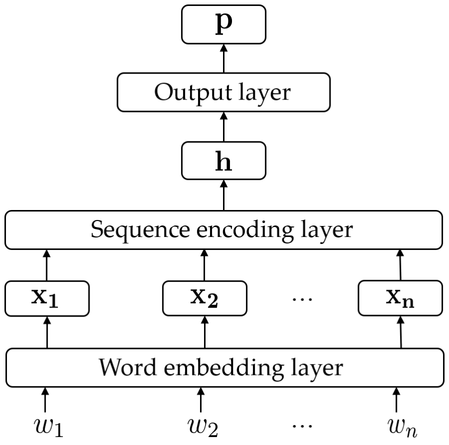
WestlakeNLP

# Using Pre-trained Transformer for Solving NLP Tasks

- Pre-training + Fine-tuning
  - Make use of pre-training knowledge (take a base model)
  - Inject task knowledge (tune it)
- Additional Model Structures
- Tasks—— NLI becomes easier!
  - Classification
  - Structured Prediction
  - Generation

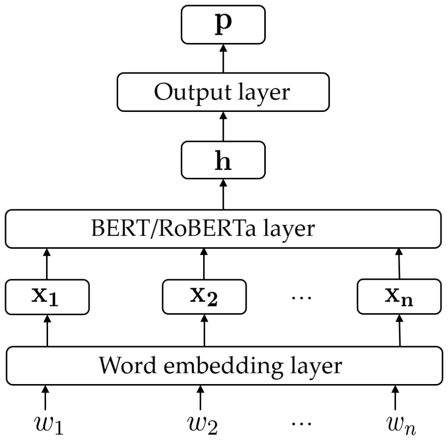
# Content

# Text Classification




Encoder model architecture

$\Rightarrow$




BERT/ RoBERTa model architecture

- Can use [CLS] for **h** or aggregated hidden for $W_{1:n}$
  - Take the whole BERT instead of word embeddings, as pre-trained parameters.
  - Fine-tune the whole BERT as fine-tuning word embeddings.

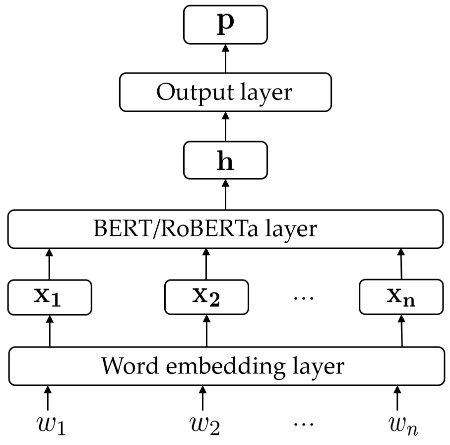
# Text Classification

WestlakeNLP

- Classifying Two Texts
- NLI
  - premise: $W_1 = w_1^1, w_2^1, \ldots, w_{n_1}^1$
  - hypothesis: $W_2 = w_1^2, w_2^2, \ldots, w_{n_2}^2$

$$X = [\mathrm{CLS}]w_1^1 \ldots w_{n_1}^1 [\mathrm{SEP}]w_1^2 \ldots w_{n_2}^2$$

$$Y = \text{entail} \;/\; \text{contradict} \;/\; \text{neutral}$$




BERT/ RoBERTa model architecture

# Text Classification

- Word Sense Disambiguation (WSD)

  *"He went to the bank and closed his account this morning"*

  - WordNet:

    bank$^1$: *sloping land (especially the slope beside a body of water)*

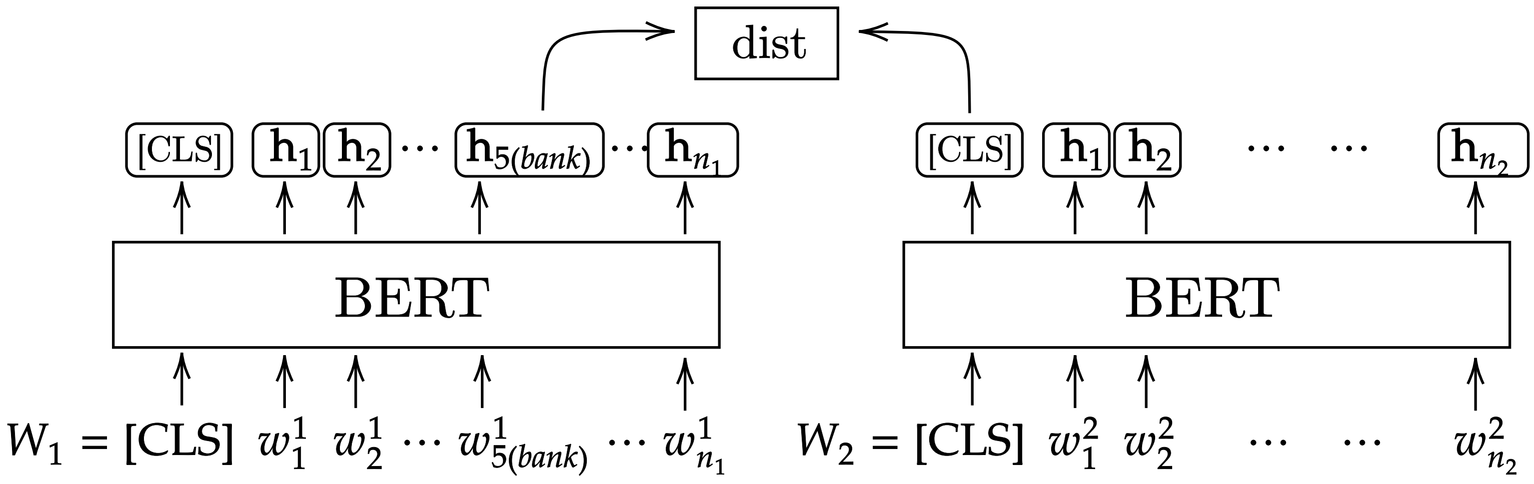
    bank$^2$: *banking company, banking concern, depository financial institution*

    bank$^3$: *a long ridge or pile*

  - Input: $W_1$, WordNet sense: $W_2$

# Text Classification

- Classify two texts




BERT/ RoBERTa model architecture

- Select the sense that has the highest score.

Ref: *Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders*

# Content

# Continued Pre-training

- done between pre-training and fine-tuning
- inject more knowledge into representation model before fine-tuning


Continued Pre-training

- domain-adaptive pre-training
  Train BERT on test domains (e.g. Biomedical, Computer Science, News reviews)
- Task-adaptive pre-training
  Train BERT on the task unlabeled data

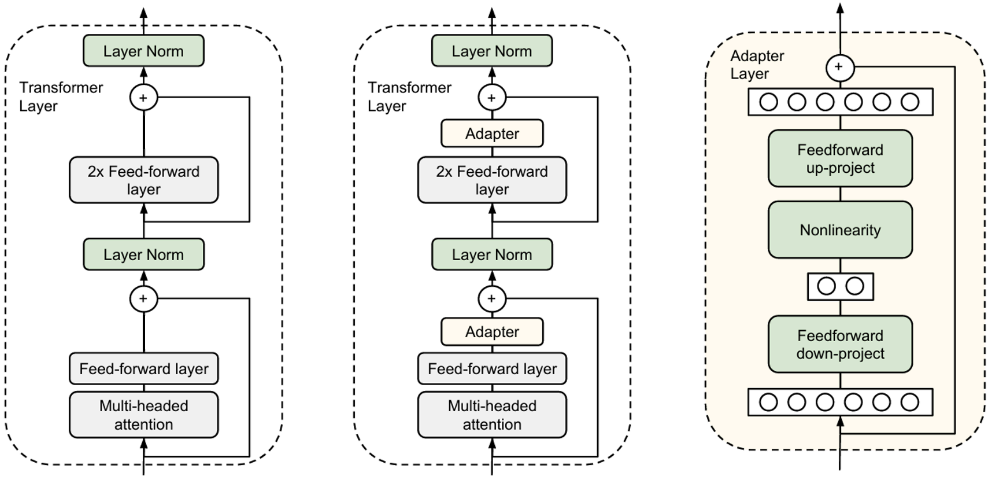
Ref: *Don't Stop Pretraining: Adapt Language Models to Domains and Tasks*

# Content

WestlakeNLP

# Adapters

- Light-weight, parameter efficient tuning
- Add additional structures to Transformer
- Instead of tuning all parameters, tune added parameters



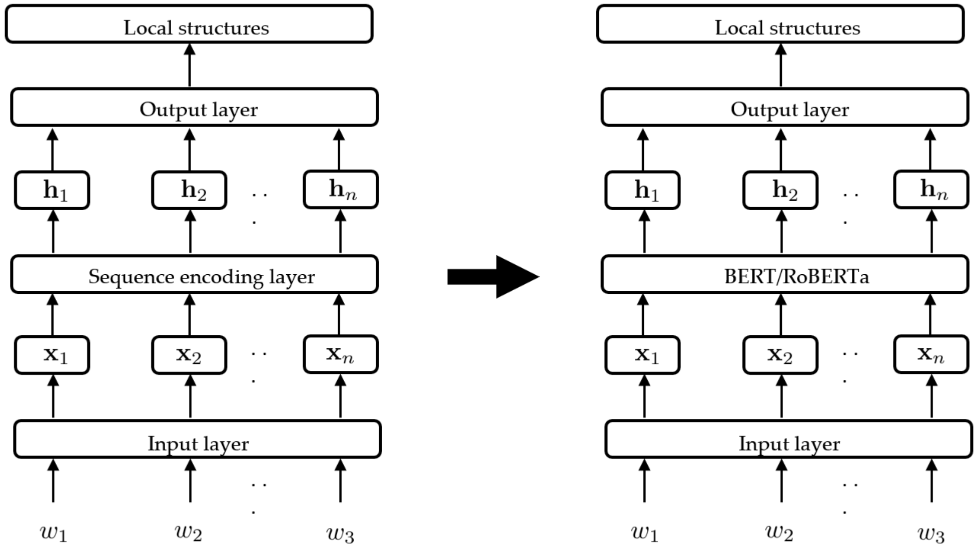

Transformer encoder | Transformer encoder with adapter | Adapter layer

Ref: *Parameter-Efficient Transfer Learning for NLP*

# Content

# Structured prediction

- useing BERT as pre-trained sequence encoder

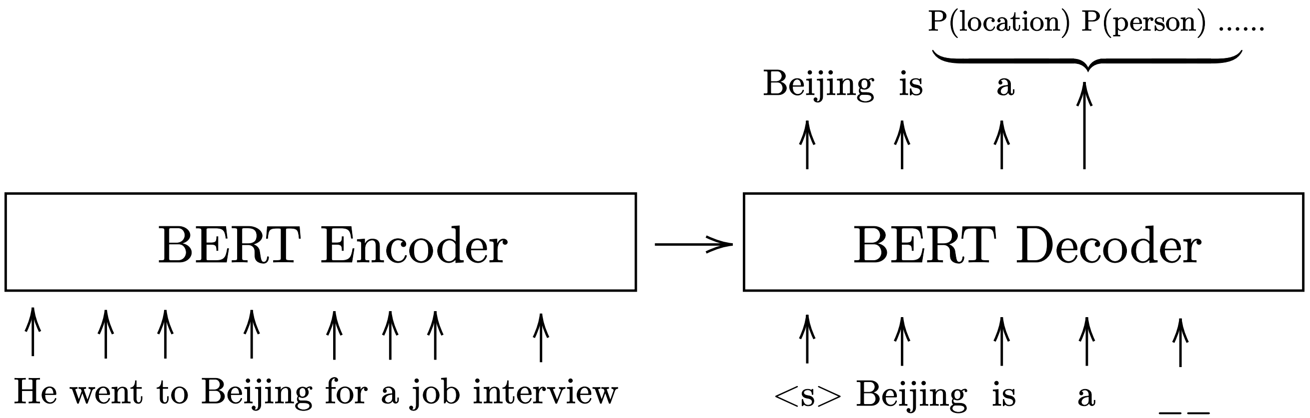



- Use the hidden states of BERT to replace Transformer
- Add the target sequence labels depending on input/output pairs
- Fine-tune the whole BERT as fine-tuning word embeddings

# Structured prediction

- Maximize pre-trained TLM utility by making the task close to pre-training

  Template-based BERT Encoder




Ref: *Template-Based Named Entity Recognition Using BART*

# Machine Reading Comprehension

WestlakeNLP

- Passage $W_1 = w_1^1 w_2^1 \ldots w_{n_1}^1$
- Question $W_2 = w_1^2 w_2^2 \ldots w_{n_2}^2$
- Input to BERT:

$$[\mathrm{CLS}] w_1^1 w_2^1 \ldots w_{n_1}^1 [\mathrm{SEP}] w_1^2 w_2^2 \ldots w_{n_2}^2$$

- Output of BERT:

$$\mathbf{h}_{[\mathrm{CLS}]}, \mathbf{h}_1^1, \mathbf{h}_2^1, \ldots, \mathbf{h}_{n_1}^1, \mathbf{h}_{[\mathrm{SEP}]}, \mathbf{h}_1^2, \mathbf{h}_2^2, \ldots, \mathbf{h}_{n_2}^2$$

- Predict on $\mathbf{h}$ beginning or end of answer span

# Machine Reading Comprehension

- SpanBERT
  - add span knowledge to BERT
  - mask whole spans (randomly sample span size, and then beginning)
  - predict span content vs. boundary tokens.
- Given $W_{1:n} = w_1 w_2 \ldots w_n$, span $W_b, \ldots, W_e (b, e \in [1, \ldots, n])$ for all words $w_i (i \in [b, \ldots, e])$.
  Predict $P\left(w_i \mid w_{\{b-1\}}, w_{\{e+1\}}, \textsc{PositionEncode}(i)\right)$
- Use masked language modeling and span prediction
- Gives improved machine reading comprehension results

Ref: *SpanBERT: Improving Pre-training by Representing and Predicting Spans*

# Machine Reading Comprehension

WestlakeNLP

- Input: a question and a relevant database table

| Student ID | Name | Class |
|---|---|---|
| 20230101 | Yue Zhang | 3A |
| 20230102 | Ting Wang | 3A |
| 20230103 | Ming Liu | 4B |

*"Which class is Yue Zhang from?"*

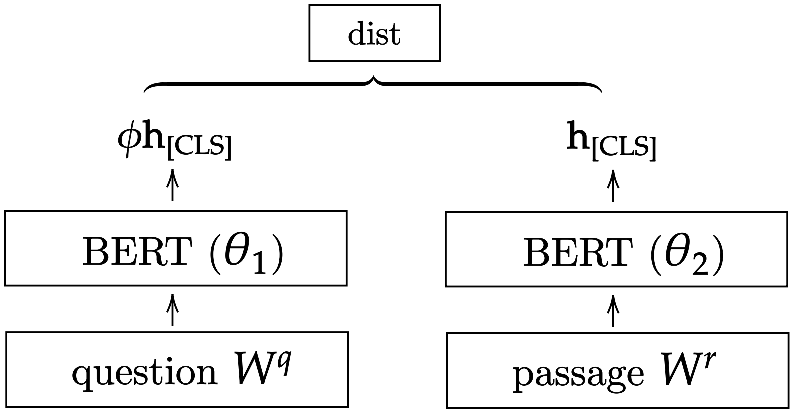
- Every database row as a sequence $W_j^{row}$
  Every database column as a sequence $W_j^{col}$
  Question as a sequence $W^q$
  Score $\mathrm{BERT}(W^q, W_j^{row})$ $\mathrm{BERT}(W^q, W_j^{col})$, find $\underset{i}{argmax}$, $\underset{j}{argmax}$

Ref: *CLTR: An End-to-End, Transformer-Based System for Cell Level Table Retrieval and Table Question Answering*

# Open Question Answering



- Dense passage retriver




- Contractive learning

  given $< W^q, W^{r+}, W_1^{r-}, W_2^{r-}, \cdots, W_m^{r-} >$

$$L = -\log \frac{e^{\mathrm{sim}(W^q, W^{r+})}}{e^{\mathrm{sim}(W^q, W^{r+})} + \sum_{i=1}^{M} e^{\mathrm{sim}(W^q, W_i^{r+})}}$$

Ref: *Dense Passage Retrieval for Open-Domain Question Answering*