

# Dive into Deep Learning for NLP

## 4. Contextual Representations

Haibin Lin

[gluon-nlp.mxnet.io](http://gluon-nlp.mxnet.io)

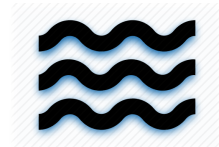
13:15-14:15	Natural Language Processing and Deep Learning Basics
14:15-14:25	Break
14:25-15:15	Context-free Representations with Word Embeddings
15:15-15:55	Machine Translation and Sequence Generation
15:55-16:35	Contextual Representations with BERT
16:35-16:45	Break
16:45-17:15	Model Deployment with TVM

# Context Matters: Retail Bank or River Bank?

1. I jog along the **bank** of Duwamish River every day.
2. I went to the **bank** to open a savings account.

# Context Matters: Retail Bank or River Bank?

1. I jog along the **bank** of Duwamish River every day.

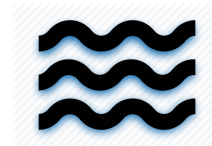


2. I went to the **bank** to open a savings account.



# Context Matters: Retail Bank or River Bank?

1. I jog along the **bank** of Duwamish River every day.



2. I went to the **bank** to open a savings account.



With word embedding, the vector representing “**bank**” is the **same** in both sentences

Can we have representations  
that depend on the **context**?

# Representations

- Context-free representation
  - CBOW/Skip-gram
  - FastText
- Contextual representation
  - ELMo: Embedding from Language Model
  - **BERT: Bidirectional Embedding Representation from Transformers**

# BERT

Bidirectional Embedding from  
Transformers





# General Language Understanding Evaluation (GLUE Benchmark)

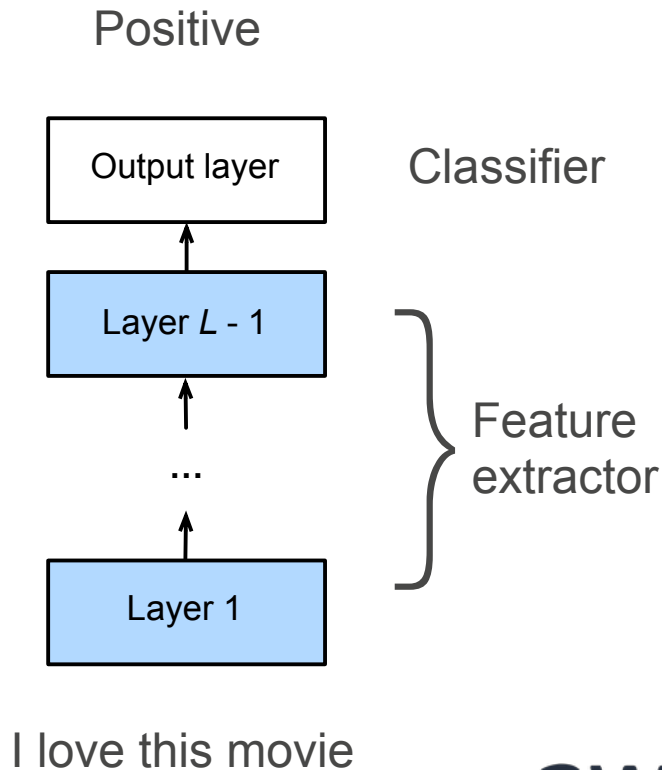
Including datasets for:

- acceptability
- sentiment
- paraphrase
- sentence similarity
- natural language inference

Model	Avg Score
CBOW	58.6
BERT	80.5

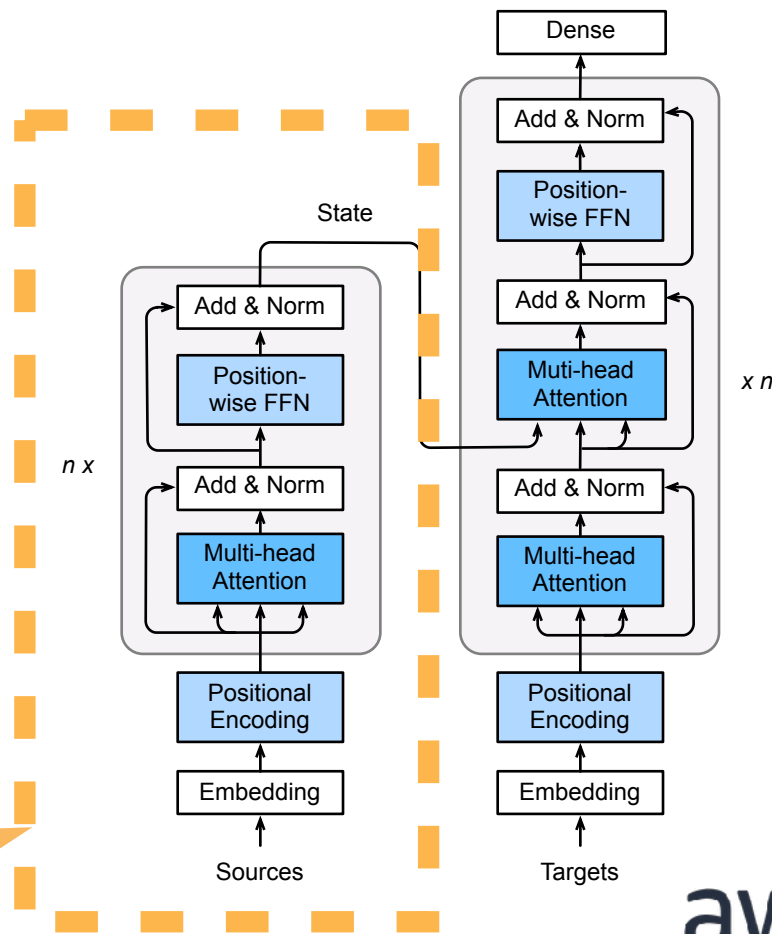
# BERT

1. Pre-training: learn contextual representation on large scale corpus
2. Fine-tuning: add a simple output layer on BERT and fine-tune with the task at hand



# BERT Architecture

- A (big) Transformer encoder
- BERT Base
  - # blocks = 12
  - # parameters = 110M
- BERT Large
  - # blocks = 24
  - # parameter = 340M



BERT

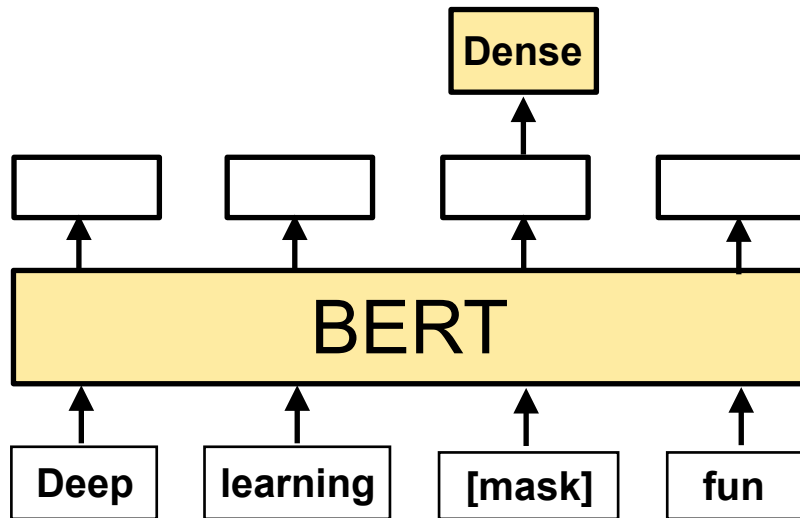
# BERT Pre-training

- Pre-training tasks:
  - masked language modeling
  - next sentence prediction
- Dataset: Wikipedia and BooksCorpus (>3B words)

# Pre-training Task 1: Masked Language Model

Original sentence:  
Deep learning is fun.

Masked sentence:  
Deep learning [mask] fun.



$$loss = -\log p(is \mid deep, learning, [mask], fun)$$

# Pre-training Task 2: Next Sentence Prediction

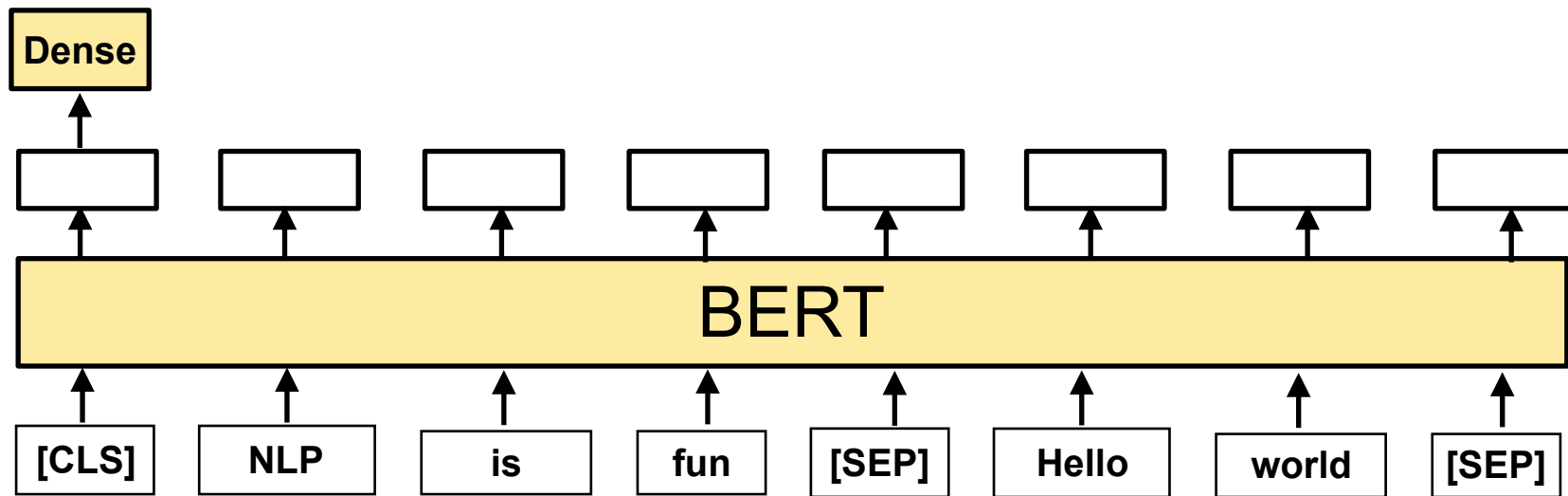
- Each example is a pair of sentences

**is\_next\_sentence:** NLP is fun. GluonNLP is awesome.

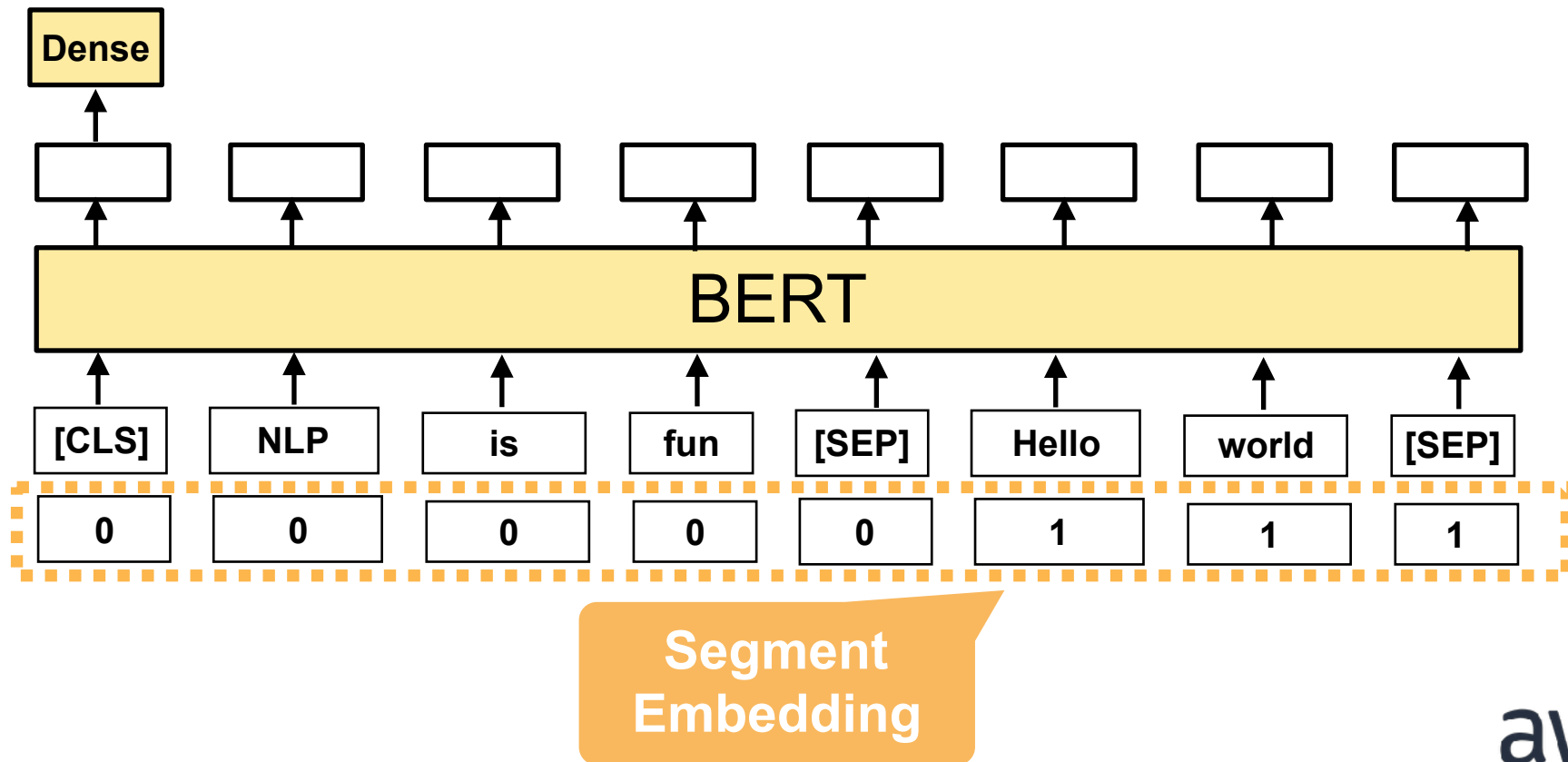
**not\_next\_sentence:** NLP is fun. Hello world.

- Sentence level binary classification

# Pre-training Task 2: Next Sentence Prediction



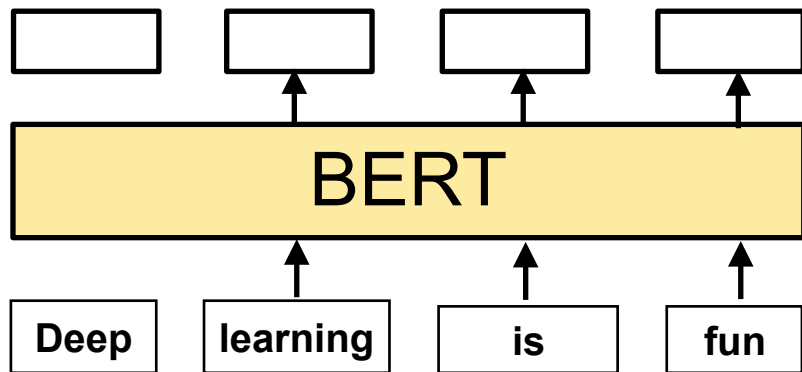
# Pre-training Task 2: Next Sentence Prediction





# BERT Fine-tuning

- BERT returns a (contextual) feature vector for each token
- Different fine-tuning tasks use a different set of vectors



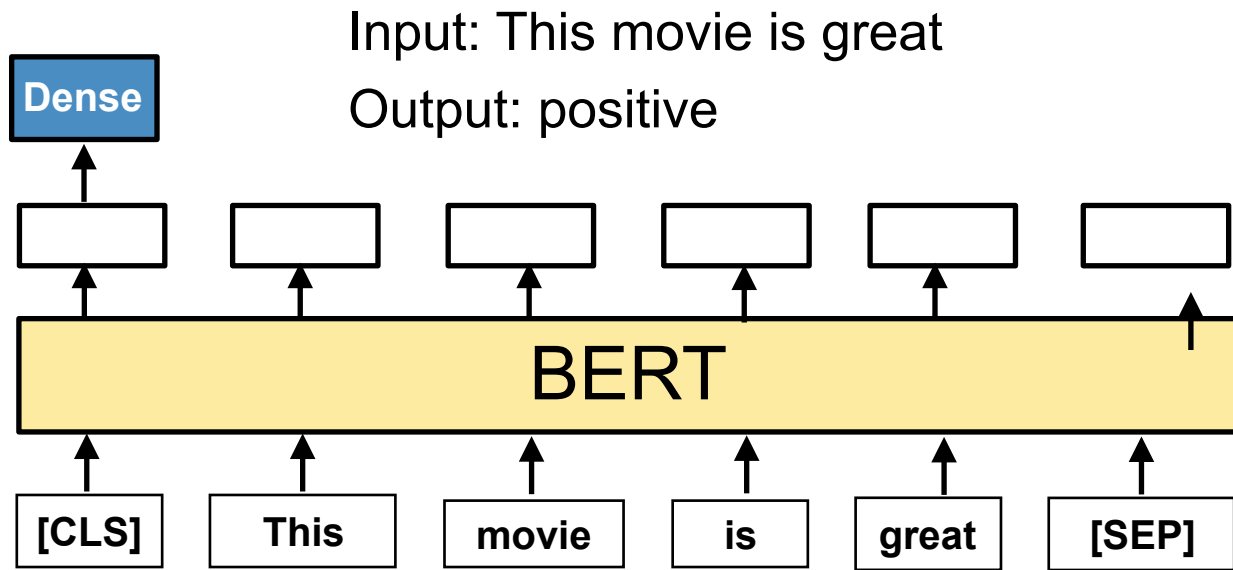
# Fine-tuning: Sentence Classification

Input: This movie is great

Output: positive

# Fine-tuning: Sentence Classification

Feed the [CLS] token vector into a dense output layer.



# Fine-tuning: Sentence Pair Classification

Input\_0: The processor was announced in San Jose at the Forum.

Input\_1: The processor was unveiled at the Forum in San Jose.

Output: is\_paraphrase

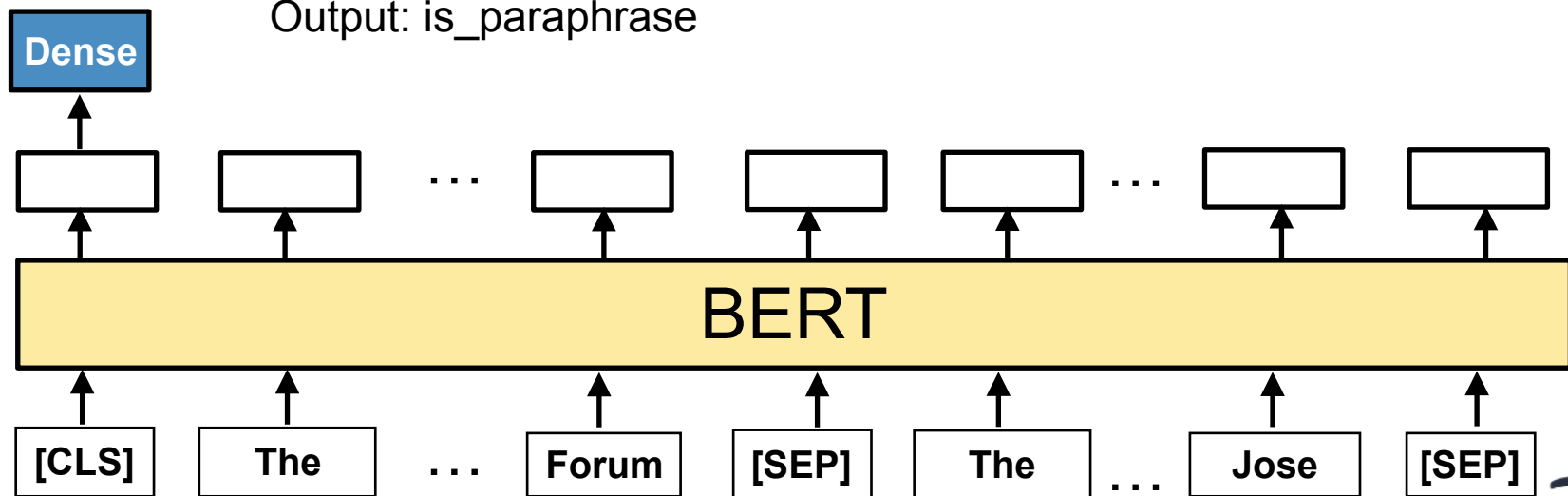
# Fine-tuning: Sentence Pair Classification

- Feed the [CLS] token vector into a dense output layer.

Input\_0: The processor was announced in San Jose at the Forum.

Input\_1: The processor was unveiled at the Forum in San Jose.

Output: is\_paraphrase



# Fine-tuning: Named Entity Recognition

Input: Jim bought 3000 shares of Amazon in 2006.

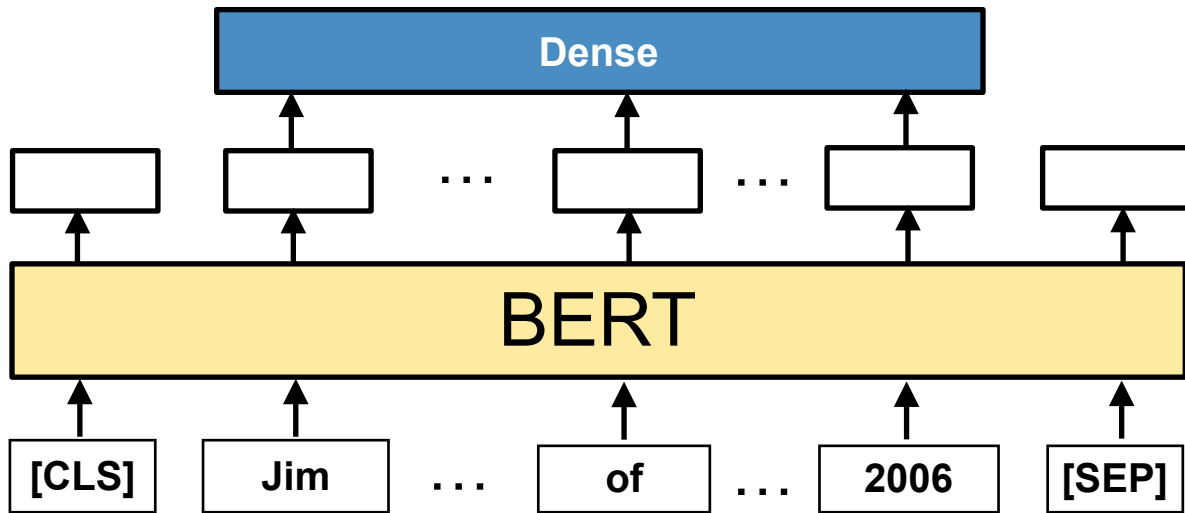
Output: [person] [organization] [time]

# Fine-tuning: Named Entity Recognition

- Feed each non-special token vector into a dense output layer

Input: Jim bought 3000 shares of Amazon in 2006.

Output: [person] [organization] [time]



# Fine-tuning: Question Answering

Given a question and a description text, find the answer, which is a text segment in the description

Input\_0: AMLC 2019 is held in Seattle

Input\_1: Where is AMLC held

Output: Seattle

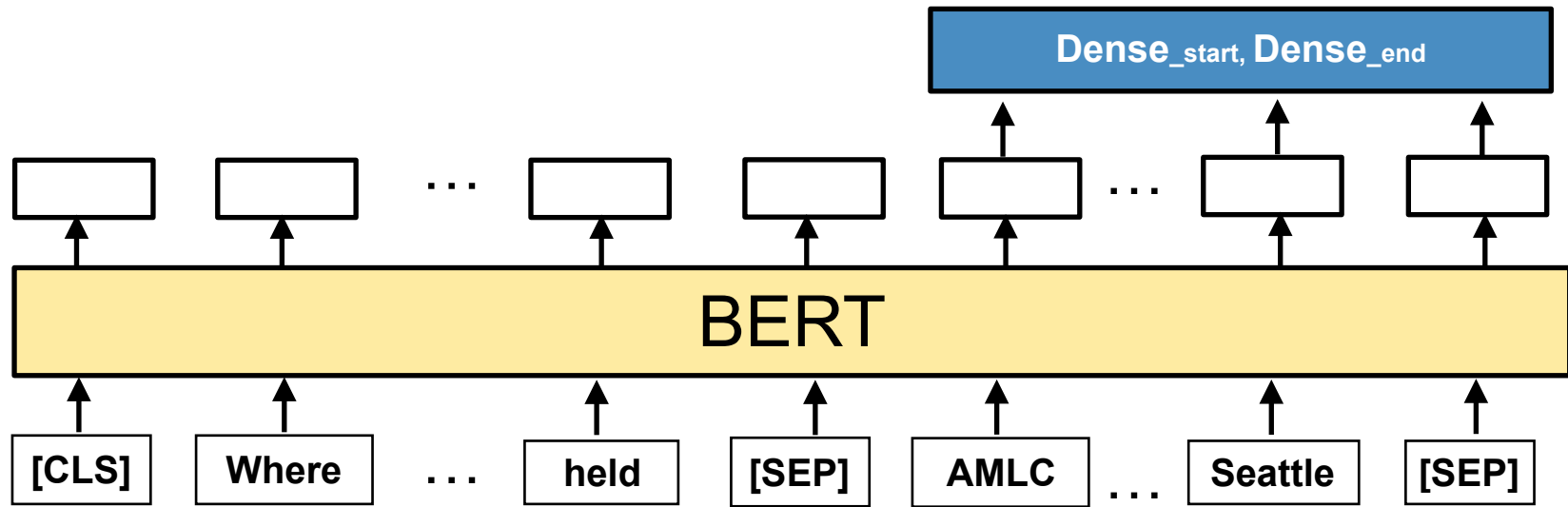


# Fine-tuning: Question Answering

Input\_0: AMLC 2019 is held in Seattle

Input\_1: Where is AMLC held

Output: Seattle



# BERT in GluonNLP

```
from gluonnlp import model  
  
model.get_model(  
    "bert_12_768_12",  
    dataset_name="wiki_cn_cased"  
)
```

[w.amazon.com?BERT](https://www.amazon.com/BERT)

	bert_12_768_12	bert_24_1024_16
book_corpus_wiki_en_uncased	✓	✓
book_corpus_wiki_en_cased	✓	✓
openwebtext_book_corpus_wiki_en_uncased	✓	x
wiki_multilingual_uncased	✓	x
wiki_multilingual_cased	✓	x
wiki_cn_cased	✓	x
scibert_scivocab_uncased	✓	x
scibert_scivocab_cased	✓	x
scibert_basevocab_uncased	✓	x
scibert_basevocab_cased	✓	x
biobert_v1.0_pmc_cased		
biobert_v1.0_pubmed_cased		
biobert_v1.0_pubmed_pmc_cased		
biobert_v1.1_pubmed_cased		
clinicalbert_uncased	✓	x
ernie_baidu_cn_uncased	✓	x

Available in  
GluonNLP

# BERT in GluonNLP

Source	Google		GluonNLP
Num layers	12	24	12
Dataset size (GB)	18	18	56
SST-2	93.5	94.9	<b>95.3</b>
RTE	66.4	70.1	<b>73.6</b>
QQP	71.2	72.1	<b>72.3</b>
SQuAD	88.5	90.9	<b>91.0</b>
STS-B	85.8	86.5	<b>87.5</b>
MNLI	83.4	<b>85.9</b>	84.9

# BERT inference with GluonNLP

## float32 inference

- BERT Base sentence classifier on Yahoo answers dataset
- with 4 cores on c5.12xlarge (out of 48 vCPUs)

Package	max_length	latency (ms)	accuracy
mxnet-mkl=1.4.1	256	178.04	74.6
latest mxnet	256	75.39	74.6

# BERT inference with GluonNLP

## float32 inference

- BERT Base sentence classifier on Yahoo answers dataset
- with 4 cores on c5.12xlarge (out of 48 vCPUs)

Package	max_length	latency (ms)	accuracy
mxnet-mkl=1.4.1	256	178.04	74.6
latest mxnet	256	75.39	74.6

## int8 inference (coming soon)

- 1.7x latency reduction, 2.2x model size reduction
- <1% accuracy drop

# Demo: BERT for Question Answering

04\_contextual\_representation/  
question\_answering.ipynb