# Pseudo-Masked Language Models for Unified Language Model Pre-Training

ICML-2020

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang,
Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, Hsiao-Wuen Hon

Microsoft

# Unified Pre-Training Framework

### Language Understanding

intent classification
entity recognition
question answering
…

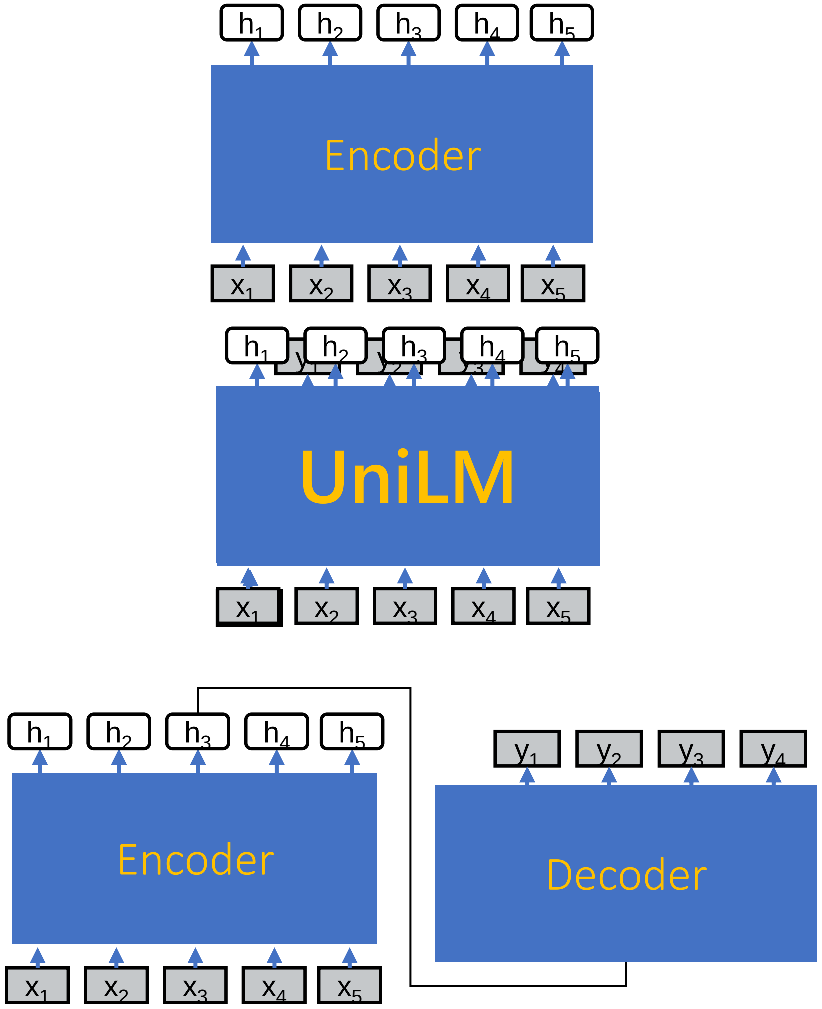
### Language Generation
(text generation)

story/news generation
…

### Language Generation
(sequence-to-sequence)

summary generation
question generation
response generation
machine translation
…

**Downstream Tasks**




### Bidirectional LM
All tokens can see each other.

BERT, RoBERTa

### Unidirectional (Left-to-Right) LM
A token can only see its left context.

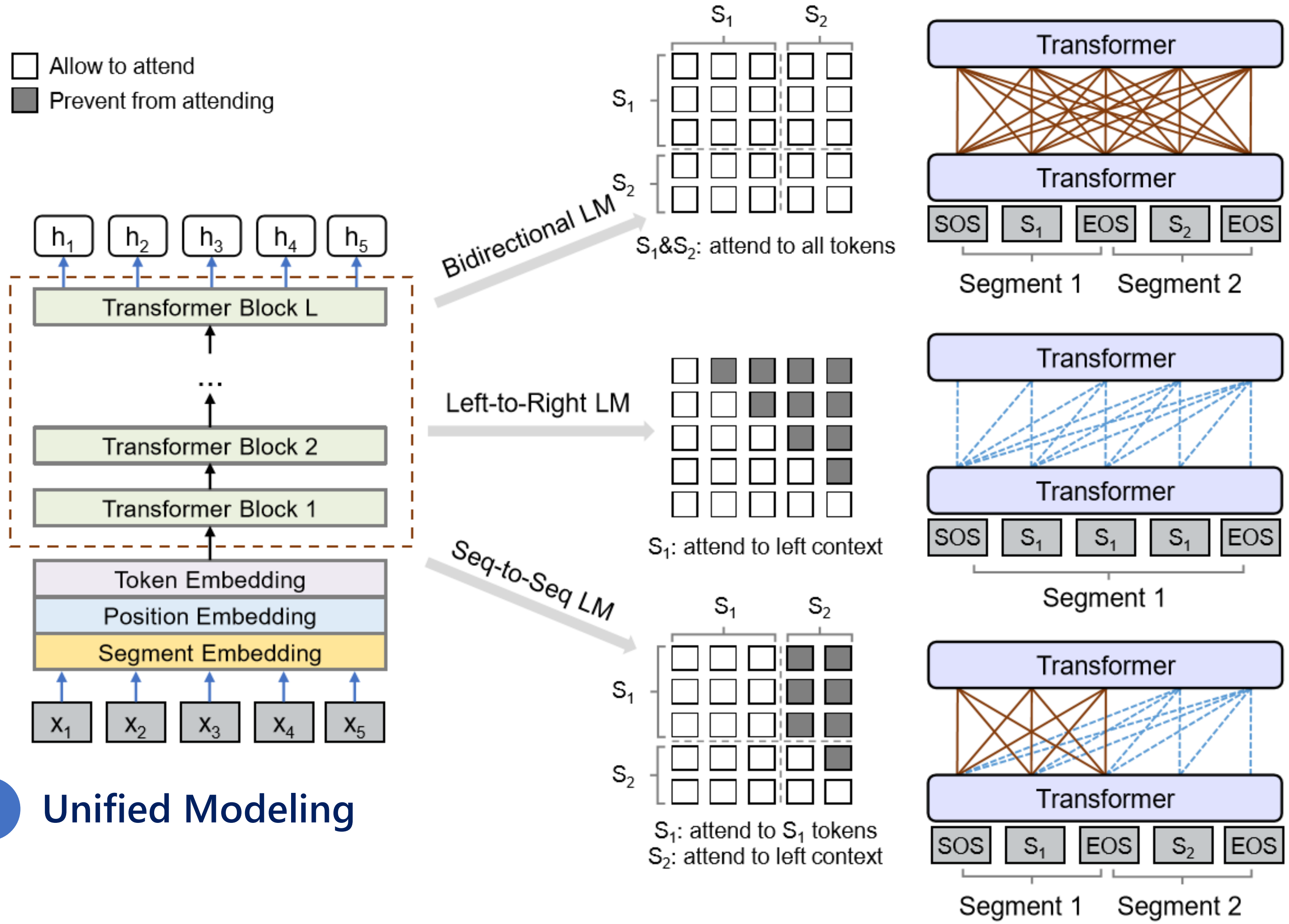
GPT

### Sequence-to-Sequence LM
1) The given input is bidirectionally encoded.
2) The output is unidirectionally decoded.

T5, BART

**Pre-Training Tasks**

# UniLM v1




1 **Unified Modeling**

2 **Multitask-Style Pre-Training**

Bidirectional Encoder

NLU: text classification, entity recognition, question answering, …

Unidirectional Decoder

NLG: synthetic text generation, …

Encoder-Decoder

NLG (sequence-to-sequence): text summarization, question generation, …

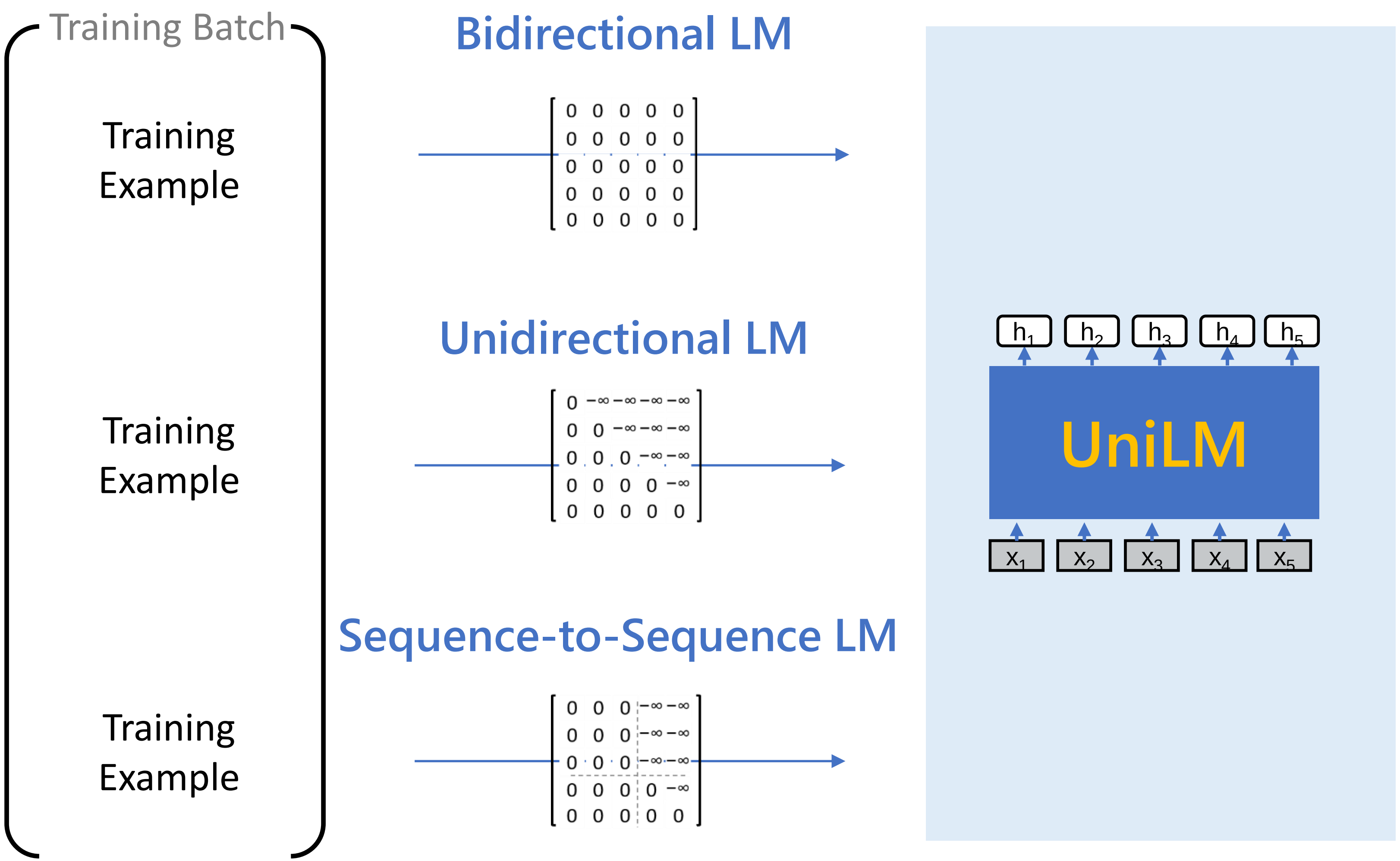
Unified Language Model Pre-training for Natural Language Understanding and Generation. NeurIPS 2019.

# Motivation of UniLM v2

(v1) One training example for each type of LM

- Three types of LMs
- Three forward passes with different self-attention masks

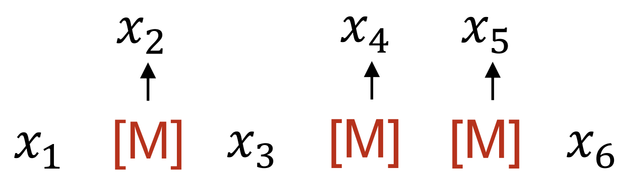
How to train multiple LMs in one forward pass?



# Pseudo-Masked Language Model

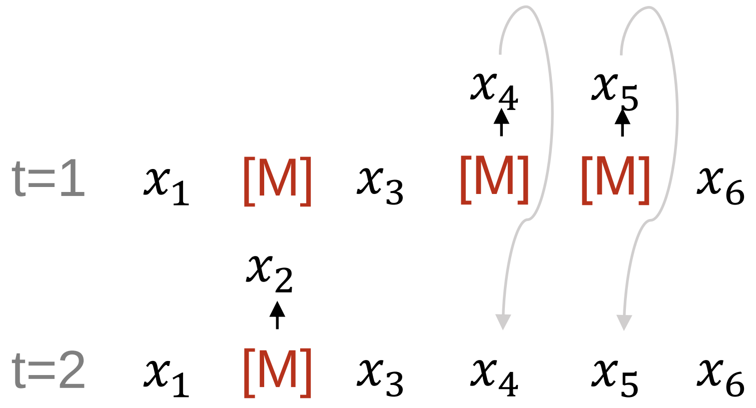
## Bidirectional LM Task (for NLU)

1. Bidirectionally encode context tokens
2. Predict the masked spans at the same time




## Sequence-to-Sequence LM Task (for NLG)

1. Bidirectionally encode context tokens
2. Predict the masked spans one by one (e.g., $x_4, x_5 \rightarrow x_2$)
    1. Predict $x_4, x_5$
    2. Encode $x_4, x_5$ (i.e., fill in what we have predicted)
    3. Predict $x_2$



# Pseudo-Masked Language Model

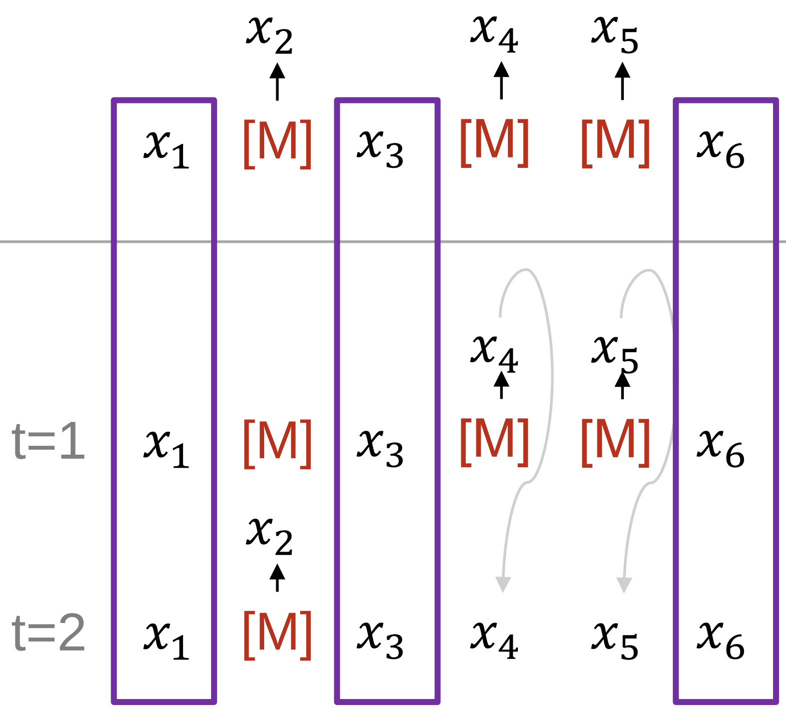
Observation 1: context encoding can be reused

## Bidirectional LM Task (for NLU)

1. Bidirectionally encode context tokens
2. Predict the masked spans at the same time

## Sequence-to-Sequence LM Task (for NLG)

1. Bidirectionally encode context tokens
2. Predict the masked spans one by one (e.g., $x_4, x_5 \rightarrow x_2$)
    1. Predict $x_4, x_5$
    2. Encode $x_4, x_5$ (i.e., fill in what we have predicted)
    3. Predict $x_2$

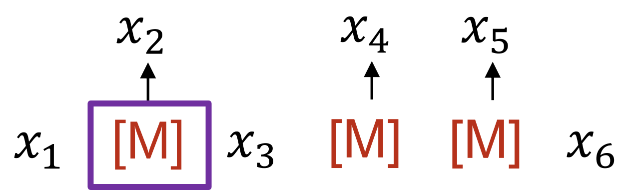


# Pseudo-Masked Language Model

Observation 1: context encoding can be reused
Observation 2: masked positions have three roles

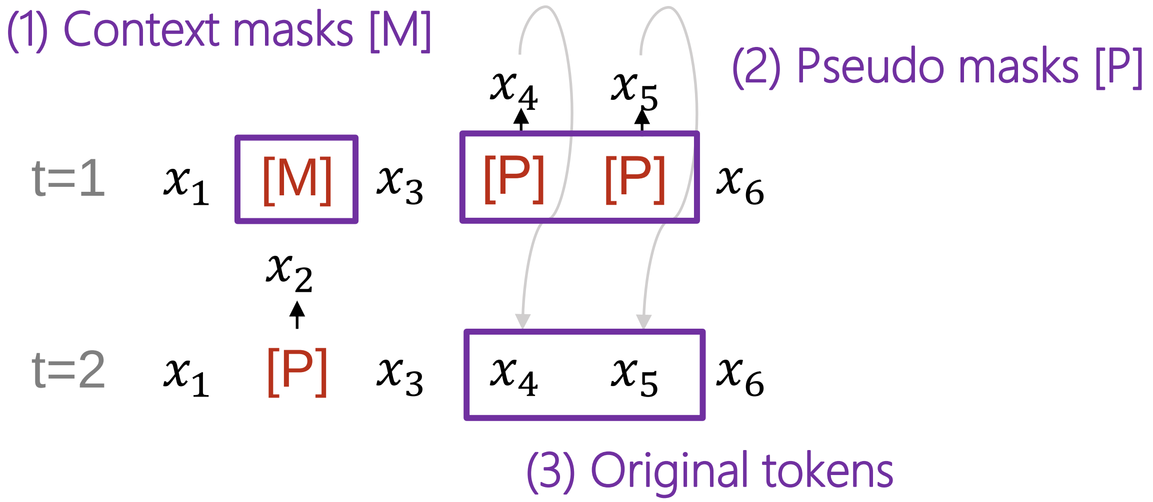
## Bidirectional LM Task (for NLU)

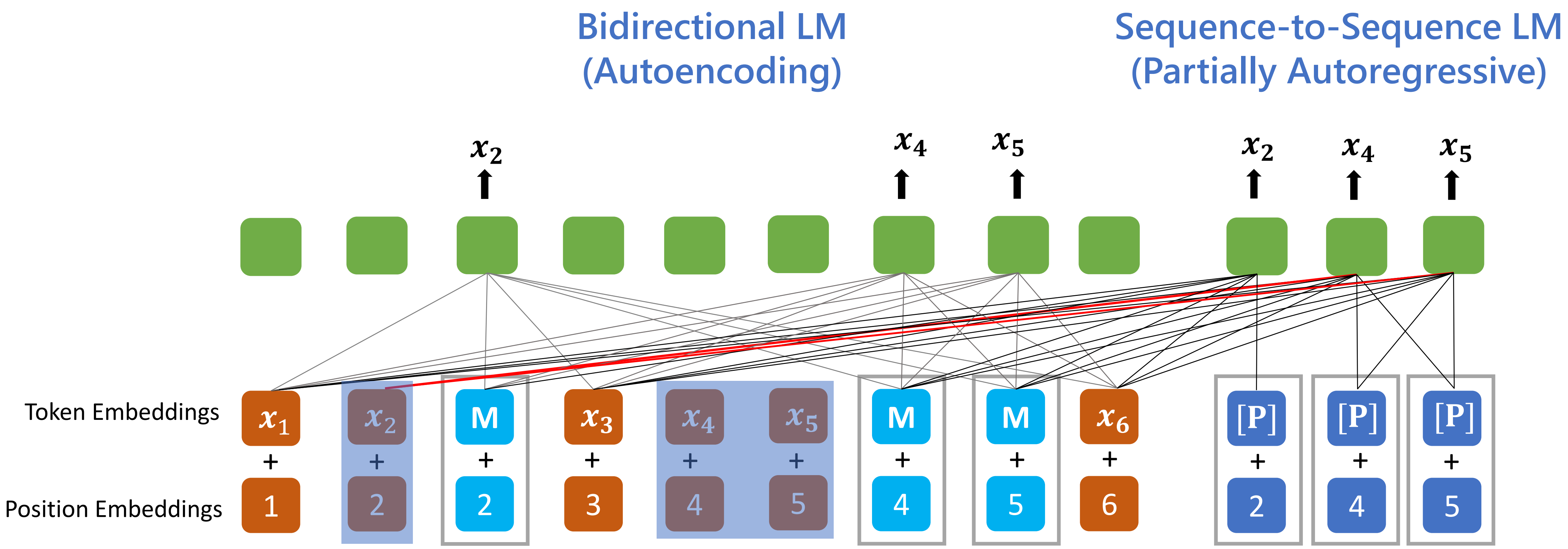
1. Bidirectionally encode context tokens
2. Predict the masked spans at the same time




## Sequence-to-Sequence LM Task (for NLG)

1. Bidirectionally encode context tokens
2. Predict the masked spans one by one (e.g., $x_4, x_5 \rightarrow x_2$)
    1. Predict $x_4, x_5$
    2. Encode $x_4, x_5$ (i.e., fill in what we have predicted)
    3. Predict $x_2$






(TL;DR) **UniLM v2**: **unified pre-training** of bi-directional LM (via autoencoding) and sequence-to-sequence LM (via partially autoregressive) with **Pseudo-Masked Language Model** for **language understanding and generation**

- Transformer/Self-attention treats tokens with the same position embeddings as the *same "token"* at that position
- Pseudo-masked LM can be used to efficiently realize different pre-training objectives, such as AE (autoencoding), AR (autoregressive), PAR (partially autoregressive) , AE + AR, and AE + PAR, among which AE + PAR performs the best

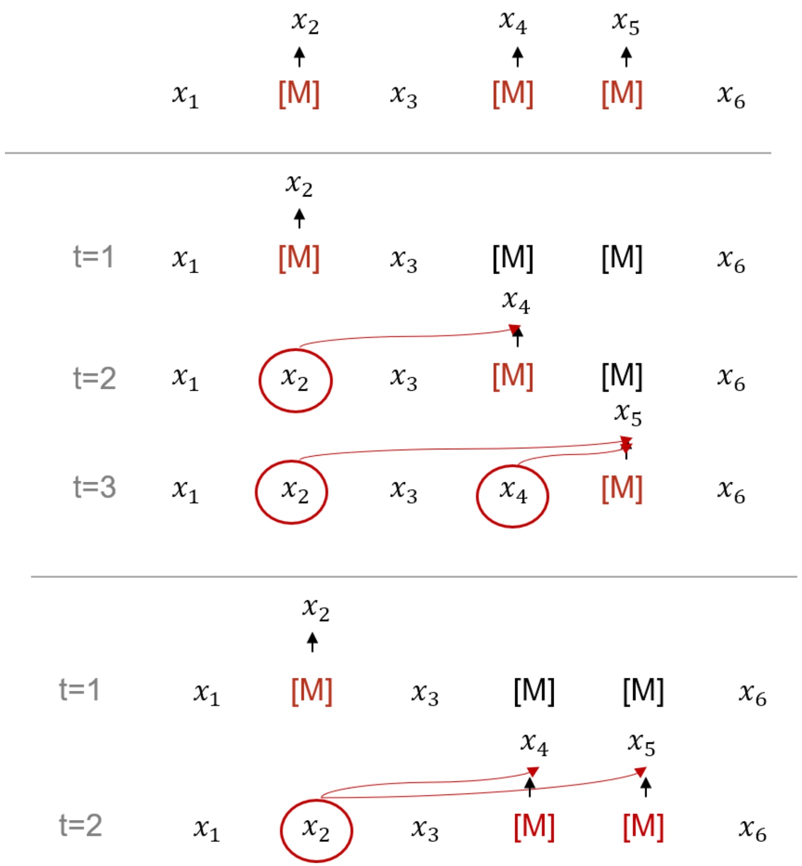
# Pre-Training Objectives

Autoencoding

Autoregressive

Partially Autoregressive

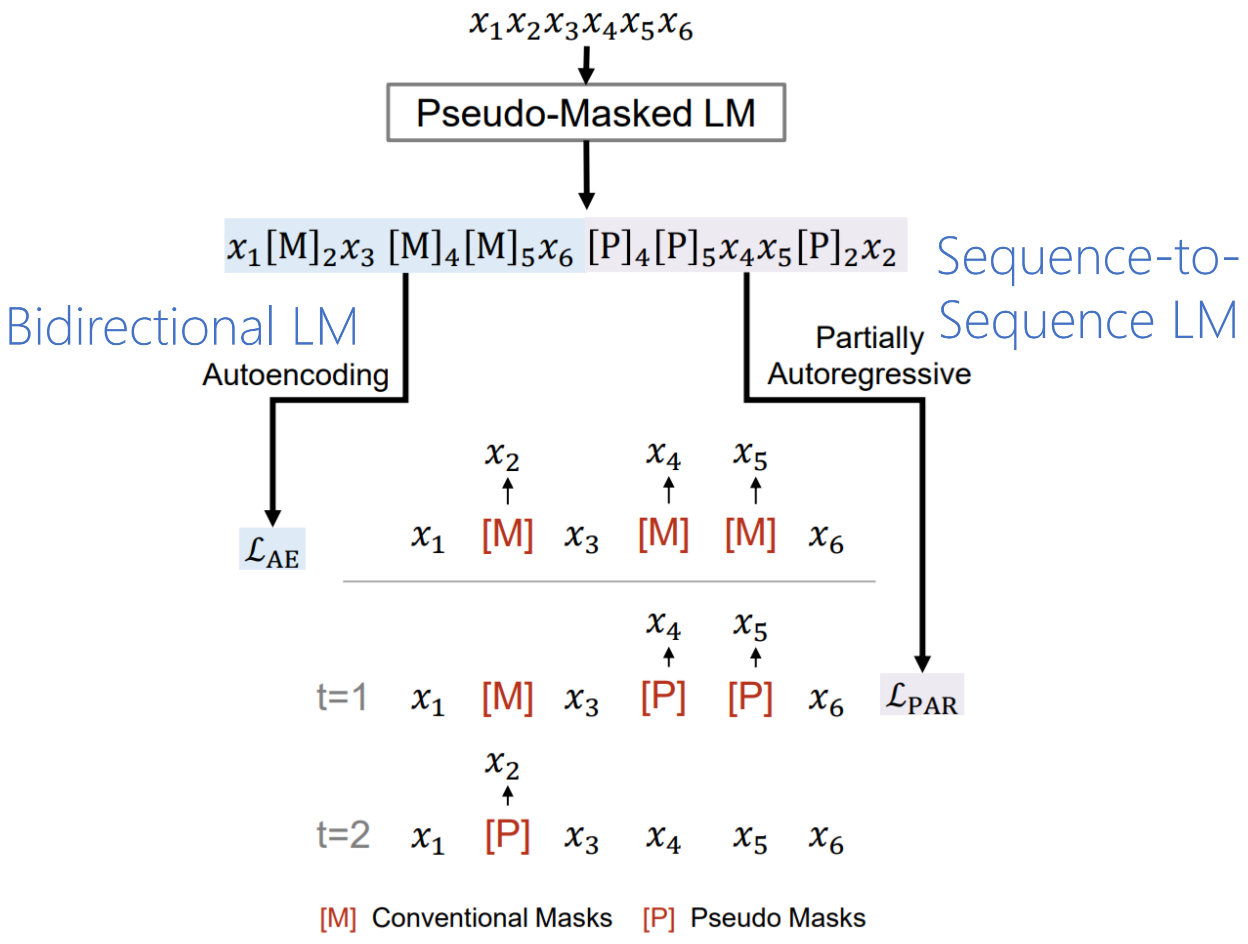
Encourage the pre-trained model to learn and use global context (long-distance dependency)




| | **Factorization Order** | **Probability of Masked Tokens** |
|---|---|---|
| Autoencoding (e.g., BERT, and our work) | – | $p(x_2 \mid x_{\setminus\{2,4,5\}})p(x_3 \mid x_{\setminus\{2,4,5\}})p(x_5 \mid x_{\setminus\{2,4,5\}})$ |
| Autoregressive (e.g., GPT, and XLNet) | $2 \rightarrow 4 \rightarrow 5$ | $p(x_2 \mid x_{\setminus\{2,4,5\}})p(x_4 \mid x_{\setminus\{4,5\}})p(x_5 \mid x_{\setminus\{5\}})$ |
| | $5 \rightarrow 4 \rightarrow 2$ | $p(x_5 \mid x_{\setminus\{2,4,5\}})p(x_4 \mid x_{\setminus\{2,4\}})p(x_2 \mid x_{\setminus\{2\}})$ |
| Partially Autoregressive (our work) | $2 \rightarrow 4, 5$ | $p(x_2 \mid x_{\setminus\{2,4,5\}})p(x_4 \mid x_{\setminus\{4,5\}})p(x_5 \mid x_{\setminus\{4,5\}})$ |
| | $4, 5 \rightarrow 2$ | $p(x_4 \mid x_{\setminus\{2,4,5\}})p(x_5 \mid x_{\setminus\{2,4,5\}})p(x_2 \mid x_{\setminus\{2\}})$ |

# Takeaway Message of UniLM v2

- Pseudo-masked language model efficiently realizes unified pre-training
- Two types of LM tasks within **one forward pass**
  - Bi-directional LM (for NLU)
  - Sequence-to-sequence LM (for NLG)
- Learn different word dependencies
  - Between context and mask predictions
  - Between mask predictions



# Benchmark Datasets

- Natural language understanding
  - Question answering (SQuAD)
  - GLUE: General Language Understanding Evaluation

Bidirectional encoding

- Natural language generation
  - Abstractive summarization
    - CNN / DailyMail
    - Gigaword
    - XSum
  - Question generation (SQuAD)

Sequence-to-sequence modeling

# UniLMv2-Base for NLU Tasks

| Model | SQuAD v1.1 | | SQuAD v2.0 | |
|---|---|---|---|---|
| | F1 | EM | F1 | EM |
| BERT | 88.5 | 80.8 | 76.3 | 73.7 |
| XLNet | - | - | - | 80.2 |
| RoBERTa | 91.5 | 84.6 | 83.7 | 80.5 |
| UniLMv2 | **93.1** | **87.1** | **86.1** | **83.3** |
| | +1.6 | +2.5 | +2.4 | +2.8 |

Results of **BASE-size** pre-trained models on the **SQuAD v1.1/v2.0** development sets. We report F1 scores and exact match (EM) scores. Results of UniLMv2 are averaged over five runs.

| Model | MNLI | SST-2 | MRPC | RTE | QNLI | QQP | STS | CoLA |
|---|---|---|---|---|---|---|---|---|
| | Acc | Acc | Acc | Acc | Acc | Acc | PCC | MCC |
| BERT | 84.5 | 93.2 | 87.3 | 68.6 | 91.7 | 91.3 | 89.5 | 58.9 |
| XLNet | 86.8 | 94.7 | 88.2 | 74.0 | 91.7 | 91.4 | 89.5 | 60.2 |
| RoBERTa | 87.6 | 94.8 | 90.2 | 78.7 | 92.8 | **91.9** | **91.2** | 63.6 |
| UniLMv2 | **88.5** | **95.1** | **91.8** | **81.3** | **93.5** | 91.7 | 91.0 | **65.2** |
| | +0.9 | +0.3 | +1.6 | +2.6 | +0.7 | -0.2 | -0.2 | +2.6 |

Results of **BASE-size** models on the development set of the **GLUE benchmark**. We report Matthews correlation coefficient (MCC) for CoLA, Pearson correlation coefficient (PCC) for STS, and accuracy (Acc) for the rest. Metrics of UniLMv2 are averaged over five runs for the tasks.

# UniLMv2-Base for NLG Tasks (Abstractive Summarization)

| **Model** | **#Param** | **#Corpus** | **CNN/DailyMail** RG-1/RG-2/RG-L | **XSum** RG-1/RG-2/RG-L |
|---|---|---|---|---|
| *Without pre-training* | | | | |
| PTRNET (See et al., 2017) | - | - | 39.53/17.28/36.38 | 28.10/8.02/21.72 |
| *Fine-tuning BASE-size pre-trained models* | | | | |
| $\text{MASS}_{\text{BASE}}$ (Song et al., 2019) | 123M | - | 42.12/19.50/39.01 | 39.75/17.24/31.95 |
| BERTSUMABS (Liu & Lapata, 2019) | 156M | 16GB | 41.72/19.39/38.76 | 38.76/16.33/31.15 |
| $\text{ERNIE-GEN}_{\text{BASE}}$ (Xiao et al., 2020) | 110M | 16GB | 42.30/19.92/39.68 | - |
| $\text{T5}_{\text{BASE}}$ (Raffel et al., 2019) | 220M | 750GB | 42.05/20.34/39.40 | - |
| UNILMV2 | 110M | 160GB | **43.87/20.99/40.95** | **44.51/21.53/36.62** |

Abstractive summarization results on CNN/DailyMail and XSum. The evaluation metric is the F1 version of ROUGE (RG) scores. We also present the number of parameters (#Param) and the corpus size (#Corpus) for the methods using pre-trained models.

# UniLMv2-Base for NLG Tasks (Question Generation)

| Model | #Param | Corpus | Official Split<br>BLEU-4 / MTR / RG-L | Reversed Split<br>BLEU-4 / MTR / RG-L |
|---|---|---|---|---|
| *Without pre-training* | | | | |
| (Du & Cardie, 2018) | - | - | 15.16 / 19.12 / - | - |
| (Zhao et al., 2018) | - | - | - | 16.38 / 20.25 / 44.48 |
| (Zhang & Bansal, 2019) | - | - | 18.37 / 22.65 / 46.68 | 20.76 / 24.20 / 48.91 |
| *Fine-tuning* BASE-*size pre-trained models* | | | | |
| ERNIE-GEN$_{\text{BASE}}$ (Xiao et al., 2020) | 110M | 16GB | 22.28 / 25.13 / 50.58 | 23.52 / 25.61 / 51.45 |
| UniLMv2 | 110M | 160GB | **24.70 / 26.33 / 52.13** | **26.30 / 27.09 / 53.19** |

MTR is short for METEOR, and RG for ROUGE. The official split is from (Du & Cardie, 2018), while the reversed split is the same as in (Zhao et al., 2018).

# Effect of Pre-Training Objectives

- AE: autoencoding
- AR: autoregressive (AR)
- PAR: partially autoregressive

| | Model | Objective | SQuAD v1.1 | | SQuAD v2.0 | | MNLI | | SST-2 |
|---|---|---|---|---|---|---|---|---|---|
| | | | F1 | EM | F1 | EM | m | mm | Acc |
| | $\text{BERT}_{\text{BASE}}$ | AE | 88.5 | 80.8 | 76.3 | 73.7 | 84.3 | 84.7 | 92.8 |
| | $\text{XLNet}_{\text{BASE}}$ | AR | - | - | 81.0 | 78.2 | 85.6 | 85.1 | **93.4** |
| | $\text{RoBERTa}_{\text{BASE}}$ | AE | 90.6 | - | 79.7 | - | 84.7 | - | 92.7 |
| | $\text{BART}_{\text{BASE}}$ | AR | 90.8 | - | - | - | 83.8 | - | - |
| [1] | $\text{UniLMv2}_{\text{BASE}}$ | AE+PAR | **92.0** | **85.6** | **83.6** | **80.9** | **86.1** | **86.1** | 93.2 |
| [2] | [1] – relative position bias | AE+PAR | 91.5 | 85.0 | 81.8 | 78.9 | 85.6 | 85.5 | 93.0 |
| [3] | [2] – blockwise factorization | AE+AR | 90.8 | 84.1 | 80.7 | 77.8 | 85.4 | 85.5 | 92.6 |
| [4] | [2] – PAR | AE | 91.0 | 84.2 | 81.3 | 78.4 | 84.9 | 85.0 | 92.4 |
| [5] | [2] – AE | PAR | 90.7 | 83.9 | 79.9 | 77.0 | 84.9 | 85.2 | 92.5 |
| [6] | [5] – blockwise factorization | AR | 89.9 | 82.9 | 79.3 | 76.1 | 84.8 | 85.0 | 92.3 |

Comparisons between the pre-training objectives. All models are pre-trained over Wikipedia and BookCorpus for one million steps with a batch size of 256. Results in the second block are average over five runs for each task. We report F1 and exact match (EM) scores for SQuAD, and accuracy (Acc) for MNLI and SST-2.

# Thanks!

https://github.com/microsoft/unilm