

WangchanBERTa: Pretraining Transformer-based Thai Language Models

Lalita Lowphansirikul and Charin Polpanumas

Mon 5 April 2021

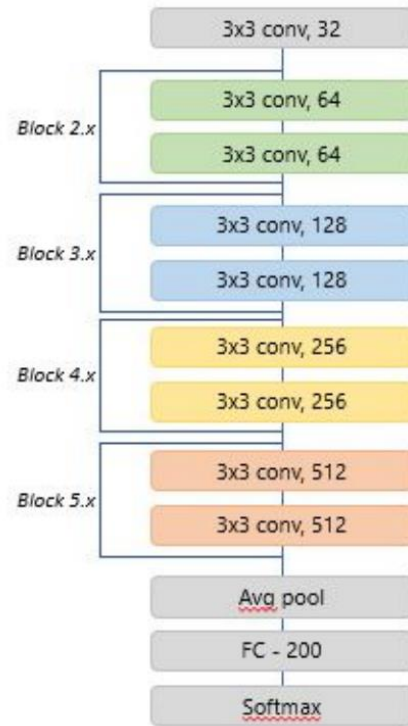
Why Pretrained Language Models ?

Pretrained images



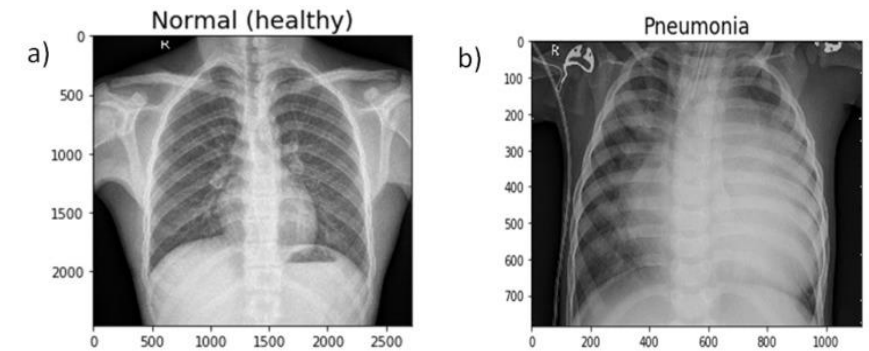
Sample Image from Tiny ImageNet (200 categories)

Figure from [Lei Sun]



ResNet

Images for target task



Chest Xray of (a) a healthy person and (b) a person suffering from pneumonia.

Figure from [Hashmi et al., 2020]

Why Pretrained Language Models

- Learn linguistic features from large text corpus to solve downstream NLP tasks
- Pretrain large model once and then finetune on downstream tasks (transfer learning)
- Reduce training time when finetuning on downstream tasks

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [[Devlin et al., 2018](#)]

- The authors proposed an approach to pretrain language model with masked language modeling (MLM) objective
- Use only stack of **Transformer encoders** (from “Attention is all you need”) papers
- Such pretrained language model is finetuned for downstream tasks (e.g. POS/NER tagging , sentiment analysis, Machine Reading Comprehension)

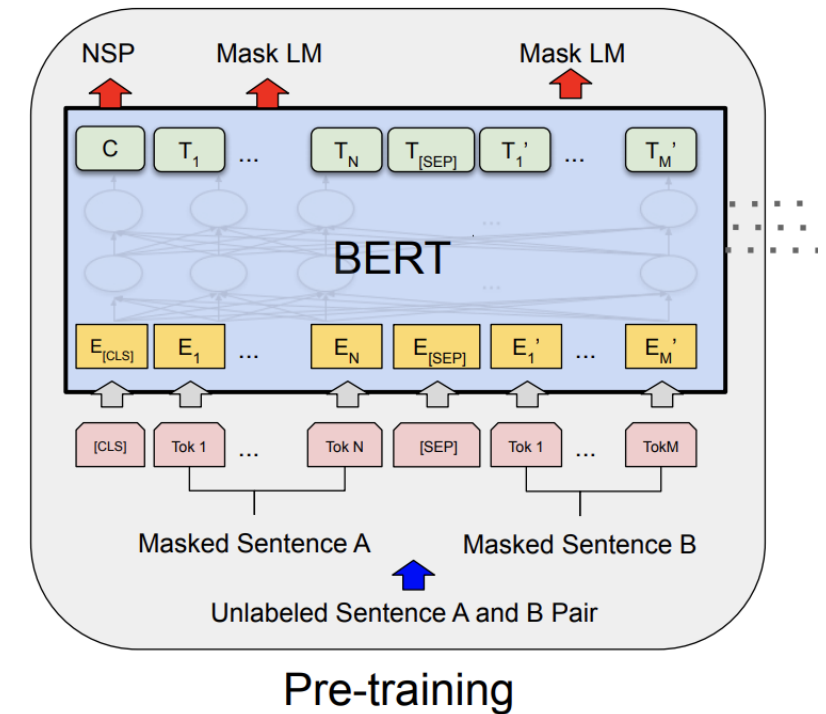


Figure from [Devlin et al., 2018](#)

Masked Language Modeling (MLM)

Masked sentence: "ที่บ้านฉันเลี้ยงหมา<MASK>ชิบะ"

Task: Predict the mask token by assign probability to each token in the vocabulary.

Masked Language Modeling (MLM)

Masked sentence: "ที่บ้านฉันเลี้ยงหมา<MASK>ชิบะ"

Task: Predict the mask token by assign probability to each token in the vocabulary.

Filled mask token #1 { token: "พันธุ์" , probability: 0.95 } : "ที่บ้านฉันเลี้ยงหมาพันธุ์ชิบะ"

Filled mask token #2 { token: "ชื่อ" , probability: 0.04 } : "ที่บ้านฉันเลี้ยงหมาชื่อชิบะ"

Filled mask token #3 { token: "ยี่ห้อ" , probability : 0.0065 } : "ที่บ้านฉันเลี้ยงหมายี่ห้อชิบะ"

Next Sentence Prediction (NSP)

Sentence A: "ที่บ้านฉันเลี้ยงหมาพันธุ์ชิบะ"

Sentence B: "มาได้เป็นเวลา 2 ปีแล้ว"

Task: Predict that sentence A is followed by sentence B from the original document

Sentence A:

Sentence B:

Prediction

"ที่บ้านฉันเลี้ยงหมาพันธุ์ชิบะ"

"มาได้เป็นเวลา 2 ปีแล้ว"

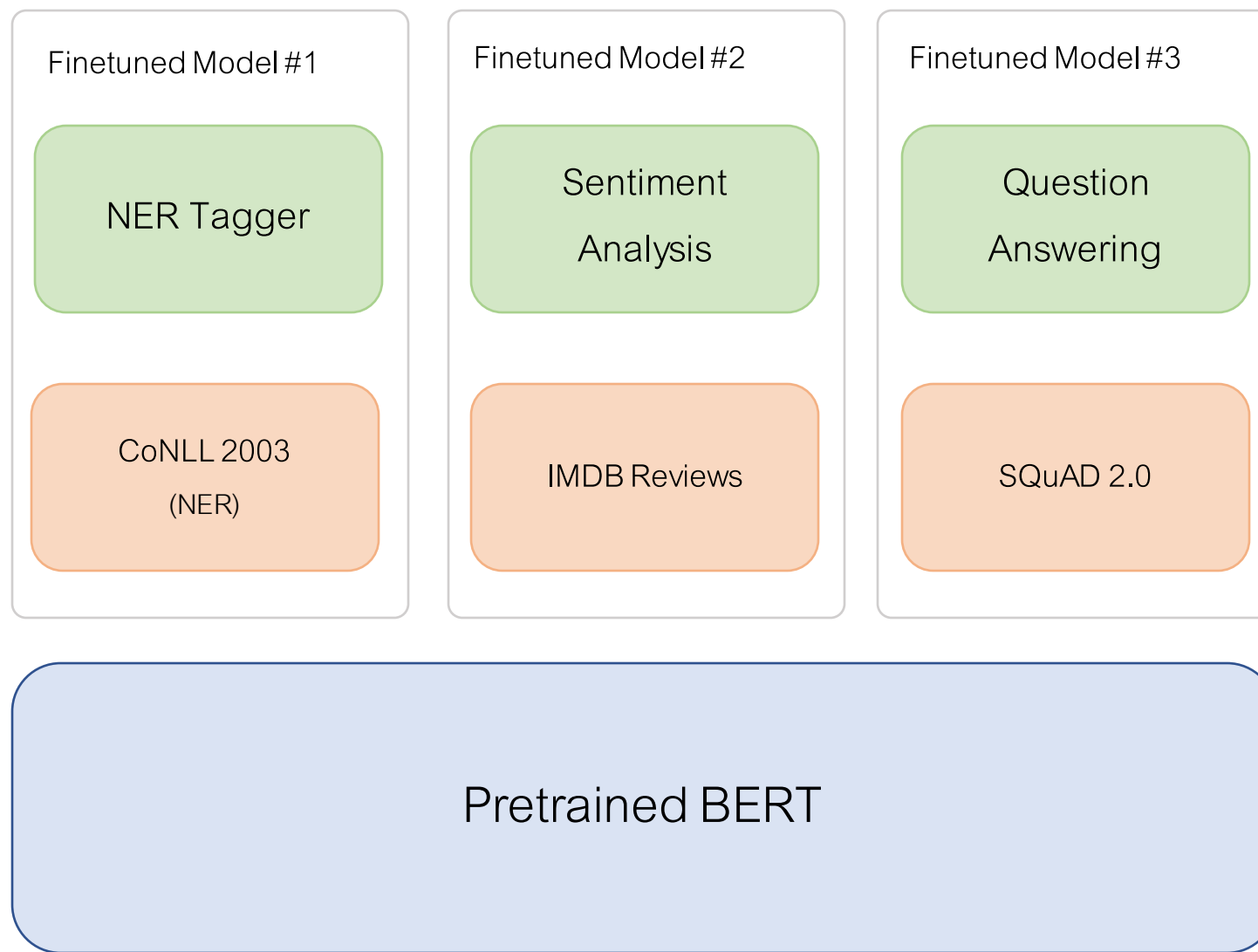


"ที่บ้านฉันเลี้ยงหมาพันธุ์ชิบะ"

"โดยนโยบายนี้ได้เริ่มใช้ตั้งแต่ปีค.ศ. 1902"



Fine-tuning BERT on Different Tasks



Evaluation results on SQuAD 1.1 and 2.0 ([Stanford QA Dataset](#))

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
BERT _{BASE} (Single)	80.8	88.5	-	-
BERT _{LARGE} (Single)	84.1	90.9	-	-
BERT _{LARGE} (Ensemble)	85.8	91.8	-	-
BERT _{LARGE} (Sgl.+TriviaQA)	84.2	91.1	85.1	91.8
BERT _{LARGE} (Ens.+TriviaQA)	86.2	92.2	87.4	93.2

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-	-	71.4	74.4
Ours				
BERT _{LARGE} (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

Evaluation results on GLUE Benchmark

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
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
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.⁸ BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

MNLI: Natural Language Inference (NLI)

QQP: Semantic equivalence

QNLI: Question Answering NLI

SST-2: Sentiment analysis

CoLA: Grammatical acceptability

STS-B: Semantic similarity

MRPC : Semantic equivalence

RTE: Textual entailment

RoBERTa: A Robustly Optimized BERT Pretraining Approach

[[Liu et al., 2019](#)]

- Opt out Next Sentence Prediction (NSP) objective, and expands maximum sequence length
- Authors pretrain RoBERTa with more data than original BERT (from 16GB to 160GB) and pretrain even longer (500K steps, w/ batch size of 8k sequences)
- RoBERTa shows significant improvement on several NLP tasks

Performance of RoBERTa on NLP tasks

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	94.6/89.4	90.2	96.4
BERT _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

MNLI: Natural Language
Inference (NLI)

SST-2: Sentiment analysis

SQuAD: Question Answering

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB \rightarrow 160GB of text) and pretrain for longer (100K \rightarrow 300K \rightarrow 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT_{LARGE}. Results for BERT_{LARGE} and XLNet_{LARGE} are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. Complete results on all GLUE tasks can be found in the Appendix.

Why Transformer-based Language Models

- Transformer-based models have had state-of-the-art downstream performance in almost all tasks ([GLUE](#), [SuperGLUE](#)), especially natural language inference, question answering and information retrieval.
- Our only existing model in Thai was [ThaiKeras BERT-th](#), which underperforms RNN-based models at least in text classification.

ผลการทดลอง Wongnai Reviews

	โมเดล	Public F1 (micro)	Private F1 (micro)	Weighted Test F1 (micro)
1	ULMFit Knight	61.11	62.58	62.14
2	ULMFit	59.31	60.32	60.02
3	fastText	51.45	51.09	51.20
4	LinearSVC	50.22	49.76	49.90
5	Kaggle Score	59.14	58.14	58.44
6	BERT-th	56.61	57.06	56.92
7	USE	42.69	41.03	41.53

Pretrained Language Models; Multilingual Models as Alternative

Name	Language	Data size	Architecture	# Params	Year
BERT	English	16GB		110M	2018
RoBERTa	English	160GB	RoBERTa base/large	125M / 355M	2019
RoBERTa-wwm-ext-lage	Chinese	5.4B tokens	RoBERTa	110M	2019
BOTO: Spanish BERT	Spanish	18GB	BERT BASE	110M	2020
CamemBERT	French	138GB	RoBERTa base/large	110M / 335M	2020
GBERT	German	163.4GB	BERT base/large	110M / 335M	2020
mBERT	Multilingual (104 languages)	-	BERT base	110M	2018
XLNet	Multilingual (100 languages)	2.5 TB	RoBERTa base/large	270M / 550M	2020

Limitations on ThaiKeras BERT-th, as Discussed with ThaiKeras

- **Hardware**; trained on 4x K80 GPUs
- **Data volume and diversity**; trained on Thai Wikipedia (about 500MB)
- **Sequence length**; limited to 128 tokens
- **Tokenization**; most subword tokenizers are domain-dependent
- **Space tokens** as important boundaries

Hardware; 4 K80s vs 8 V100s vs 1,024 v100s

Better hardware means you have more room for iterations. We can iterate with smaller datasets but that sometimes defeat the purpose of training a LARGE language model.

Model	# of GPUS	Dataset Size	Effective Batch Size	Steps	Days Spent
ThaiKeras BERT-th	4 K80s	0.5GB	32	1M	20
WangchanBERTa	8 V100s	78GB	4,092	500k	134
RoBERTa	1,024 V100s	160GB	8,000	500k	1
XLM-RoBERTa	500 V100s	2.5TB	8,192	1.5M	NA

Data Volume and Diversity

- Thai Wikipedia is over 100x smaller than Thai texts used to train XLM-RoBERTa (71.7GB) and over 300x smaller than texts used to train the original RoBERTa.
- Wikipedia also only include formal texts from encyclopedia.

Model	# of GPUS	Dataset Size	Effective Batch Size	Steps	Days Spent
ThaiKerasBERT-th	4 K80s	0.5GB	32	1M	20
WangchanBERTa	8 V100s	78GB	4,092	500k	134
RoBERTa	1,024 V100s	160GB	8,000	500k	1
XLM-RoBERTa	500 V100s	2.5TB	8,192	1.5M	NA

Short Sequence Length

Text; Sequence length=128; WangchanBERTa SentencePiece tokenizer

<s> hamtaro หรือ แสมทาโร่ แก๊งจิ๋วผจญภัย การ์ตูนญี่ปุ่นที่มีเหล่าแฮมสเตอร์เป็นตัวละครหลัก เป็นผลงานของอาจารย์ kawai ritsuko เดิมที แสมทาโร่ นั้นเป็นนิทานสำหรับเด็กมาก่อนถูกตีพิมพ์ที่ญี่ปุ่นในปี ค.ศ. 1997 เพราะกองบรรณาธิการของ นิตยสารการ์ตูนอยากได้การ์ตูนที่มีตัวเอกเป็นแฮมสเตอร์ และอาจารย์ก็กำลังเลี้ยงแฮมสเตอร์อยู่พอดีไม่แปลกใจเลย ทำไมอาจารย์ถึงวาดการ์ตูนและถึงเล่าถึงกิจวัตรประจำวันของเหล่าแฮมสเตอร์ได้ออกมาสมจริงและน่ารักสุดๆ หนังสือ นิทาน hamtaro ได้รับความนิยมมากจนกลายมาเป็นทีวี อนิเมะ ในหน้าร้อนของปี ค.ศ. 2000 เป็นที่นิยมทั้งเด็กทุกวัยไปจนถึงวัยผู้ใหญ่ด้วย

Model	# of GPUS	Dataset Size	Effective Batch Size	Sequence Length
ThaiKeras BERT-th	4 K80s	0.5GB	32	128
WangchanBERTa	8 V100s	78GB	4,092	416
RoBERTa	1,024 V100s	160GB	8,000	512
XLM-RoBERTa	500 V100s	2.5TB	8,192	512

Tokenization; Most Subword Tokenizers Are Domain Dependent

Even same SentencePiece tokenizers might get different results with different training set. Moreover, WordPiece tokenizer tokenizes too small subwords; we will see in later sections how this leads to a challenge in question answering task.

Text: ศิลปะไม่เป็นเจ้านายใครและไม่เป็นข้าใคร

WangchanBERTa(spm): ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้านาย', 'ใคร', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

WangchanBERTa-processed(spm): ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้านาย', 'ใคร', '<_>', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

XLNet(spm): ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้า', 'นาย', 'ใคร', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

MBERT(WordPiece): ['[CLS]', 'ศ', '##ิล', '##ป', '##ะ', '##ไ', '##ม', '##่', '##เป', '##็น', '##เ', '##จ', '##้า', '##นา', '##ย', '##ไ', '##ค', '##ร', '##และ', '##ไ', '##ม', '##่', '##เป', '##็น', '##ข', '##ี่', '##ั', '##ข', '##้า', '##ไ', '##ค', '##ร', '[SEP]']

Space Tokens as Important Boundaries

SentencePiece will create tokens where a space token is merged another non-space token.

Text: ศิลปะไม่เป็นเจ้านายใครและไม่เป็นข้าใคร

WangchanBERTa: ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้านาย', 'ใคร', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

WangchanBERTa-processed: ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้านาย', 'ใคร', '<_>', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

XLM-RoBERTa: ['<s>', ' ', 'ศิลปะ', 'ไม่เป็น', 'เจ้า', 'นาย', 'ใคร', 'และ', 'ไม่เป็น', 'ข้า', 'ใคร', '</s>']

mBERT: ['[CLS]', 'ศ', '##ิล', '##ป', '##ะ', '##ไ', '##ม', '##่', '##เป็', '##็น', '##เ', '##จ', '##ั', '##นา', '##ย', '##ไ', '##ค', '##ร', '##ล', '##ะ', '##ไ', '##ม', '##่', '##เป็', '##็น', '##ข', '##ี่', '##ั', '##ข', '##ั', '##ไ', '##ค', '[SEP]']

[Known Issue] There were still 650 —tokens out of 25,003 in WangchanBERTa despite replacing all spaces with space tokens (<_>)

due to space between beginning of sentence token (<s>) and the actual text. Example และ vs —และ are two separate tokens:

และ: 13

—และ: 222

WangchanBERTa: Transformer-based Thai Language Models






- Architecture: RoBERTa_{BASE} (110M params)
- Train on word-level, subword-level and syllable-level tokens
- Dataset are from various sources (Assorted Thai Texts)
(Total size of uncompressed text: 78GB)
- Finetune on
 - 4 text classification datasets
 - 2 NER/POS tagging datasets
 - 1 natural language inference dataset and
 - 1 question answering dataset
- Hardware: 8x NVIDIA V100 GPU (~4 months)

Road to WangchanBERTa

- Find large and diverse datasets
- Clean and prepare the datasets
- Choose a tokenizer
- Choose an architecture
- Training a really, really large language model

Publicly Available Datasets Are Few and Formal

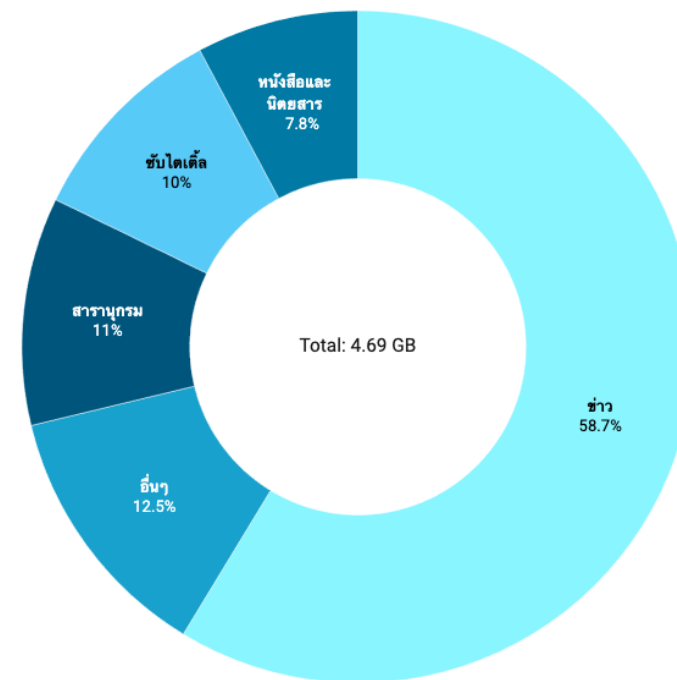
Our models will have difficulty dealing with:

- > บ้านเรามีแต่ศิลปินตลกๆ ของน้องนุ้ยมีไปหมดเลยอะอะอะอะ
- > ศิลปินี่แสนน่ารักเลย มือเล็กกว่าหน้าอีก
- > อัย่้องนุ้ยนนนนนนนนนนนนนนนน   
- > อู้ววววววว ดีกบ๊ใจอะไรเบอร์นี้ ขอบคุณม้ก ๆ เลยนะคะ เยิฟอ๊ะ ^^

4.69GB is also still very far away from 71.7GB of XLM-RoBERTa, not to mention 160GB of RoBERTa in English

ขนาดข้อมูลจากชุดข้อมูลเปิด (4.69 GB)

■ ชาว ■ อื่นๆ ■ สารานุกรม ■ ซับไตเติ้ล ■ หนังสือและนิตยสาร



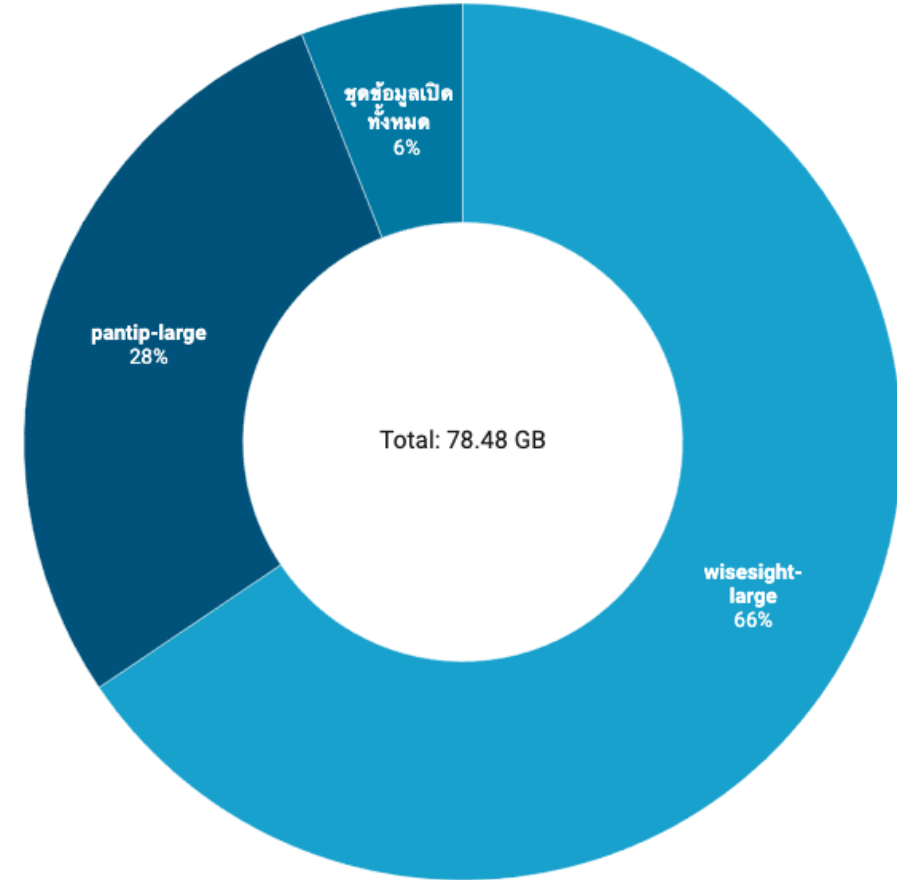
Get the data • Created with Datawrapper

Assorted Thai Texts

Wisesight, Chaos Theory and Pantip to the rescue

[Wisesight](#) contributed 51.44GB of data from Twitter, Facebook, Pantip, Instagram, and YouTube in 2019.

[Chaos Theory/Pantip](#) contributed 22.35GB of Pantip data from 2015-2019.



Why Not Use CommonCrawl like XLM-RoBERTa

Our experience with web crawled data suggests they need extensive, language-specific cleaning to be useful. [T5 paper](#) also suggested that training on [C4 \(745GB\)](#) has better results than training on Unfiltered C4 (6.1TB) despite 8x size difference.

Example of Thai sentences from CommonCrawl

Menu bar:

ค้นหาเว็บไซต์: ทุกประเภท กล้อง,อุปกรณ์ถ่ายภาพกีฬา,อุปกรณ์ท่องเที่ยวเกมส์ของเก่า,ของสะสมของที่ระลึก คอมพิวเตอร์,อุปกรณ์เครื่องเขียน,เครื่องใช้สำนักงานเครื่องใช้ไฟฟ้า,อิเล็กทรอนิกส์เครื่องดนตรีเครื่องประดับ ,อัญมณี

Menu bar:

หน้าหลัก > ผลิตภัณฑ์บำรุงผิวหน้า > กลุ่มผลิตภัณฑ์บำรุงผิวหน้า > ผลิตภัณฑ์บำรุงผิวหน้า > การ์นิเย่ โลท์ คอมพลีท >ฟิล ออฟ มาส์ก

Incorrect
format:

ปีนี้ถือว่าเป็นปีทองของนักเรียนไทยที่สามารถสร้างชื่อเสียงในการแข่งขันโอลิมปิกวิชาการ bpeeM neeH theuuR waaF bpenM bpeeM thaawngM khaawngR nakH riianM thaiM theeF saaR maatF saangF cheuuF siiangR naiM gaanM khaengL khanR o:hM limM bpikL wiH chaaM gaanM example sentence "This is a very special year for that students who made a name for themselves at the Academic Olympics competition."

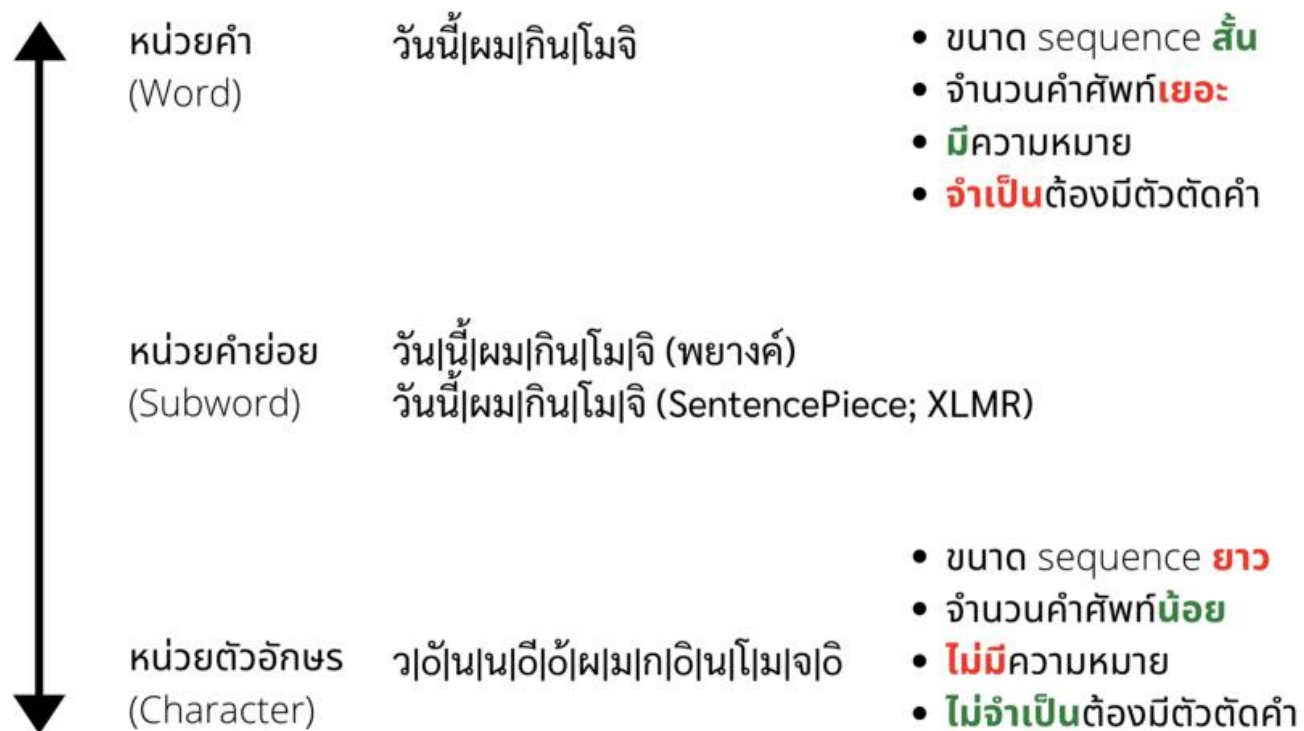
Clean and prepare the datasets

thai2transformers.preprocess.process_transformers

- Replace extra spaces, newlines, html tags with just space
- Remove html tags
- Remove empty brackets e.g. (), {}, []
- Normalize repeated characters e.g. อัยน์นนน→อัยน์
- Normalized repeated (newmm) words e.g. เต็มไปหมดเลยอะอะอะอะอะเป็น→เต็มไปหมดเลยอะ
- Break long texts into 300-(newmm)word sequences with CRFCut
- Deduplication; 295GB down to 78GB

Choose A Tokenizer

Subword-level tokenization seems to be the way to go.




Choose A Tokenizer

But which subword tokenizers?

- Dictionary-based
 - Syllable; possibly domain independent; which syllable tokenizer?
 - Word; tokens have meaning; large vocabs
- BPE/Byte-level BPE (GPT) iteratively merges most frequent tokens;
requires pre-tokenization
- WordPiece (BERT) iteratively merges most likely tokens;
requires pre-tokenization
- SentencePiece ([unigram](#)) (RoBERTa) iteratively eliminates tokens that least affect negative log-likelihood of the training data; does NOT require pre-tokenization

Choose A Tokenizer

Space tokens () serve as important boundary token for tasks such as sentence breaking and other token classification tasks.

Example: sentence breaking task

Text: ศิลปะไม่เป็นเจ้านายใครและไม่เป็นข้าใคร

WangchanBERTa:

[('ศิลปะ', 'O'), ('ไม่เป็น', 'O'), ('เจ้านาย', 'O'), ('ใคร', 'O'), ('และ', 'E'), ('ไม่เป็น', 'O'), ('ข้า', 'O'), ('ใคร', 'O')]

WangchanBERTa with space tokens:

[('ศิลปะ', 'O'), ('ไม่เป็น', 'O'), ('เจ้านาย', 'O'), ('ใคร', 'O'), ('<_>', 'E'), ('และ', 'O'), ('ไม่เป็น', 'O'), ('ข้า', 'O'), ('ใคร', 'O')]

Choose A Tokenizer

Eyeballing for vocab size of SentencePiece

5k วิชา|ที่|อาจารย์|อรรถ|พล|สอน|คือ| |ศาสตร์|ที่|นำ|ทฤษฎี|ทาง|ภาษา|ศาสตร์|มา|รวม|กับ|เทคโนโลยี|ต่างๆ| |เป็น|
การศึกษา|ที่ใช้|การ|ผสม|ผสาน|ระหว่าง|วิทยา|การ|คอมพิว|เตอร์|และ|ทฤษฎี|ทาง|ภาษา|ศาสตร์

25k วิชา|ที่|อาจารย์|อรรถ|พล|สอน|คือ| |ศาสตร์|ที่|นำ|ทฤษฎี|ทาง|ภาษา|ศาสตร์|มา|รวม|กับ|เทคโนโลยี|ต่างๆ| |เป็น|
การศึกษา|ที่ใช้|การ|ผสม|ผสาน|ระหว่าง|วิทยา|การ|คอมพิว|เตอร์|และ|ทฤษฎี|ทาง|ภาษา|ศาสตร์

32k วิชา|ที่|อาจารย์|อรรถ|พล|สอน|คือ| |ศาสตร์|ที่|นำ|ทฤษฎี|ทาง|ภาษา|ศาสตร์|มา|รวม|กับ|เทคโนโลยี|ต่างๆ| |เป็น|
การศึกษา|ที่ใช้|การ|ผสม|ผสาน|ระหว่าง|วิทยา|การ|คอมพิว|เตอร์|และ|ทฤษฎี|ทาง|ภาษา|ศาสตร์

Our Tokenizers for WangchanBERTa

Model	Dataset	Trained Sentences	Vocab Size (Excluding Special Tokens)
SentencePiece	Assorted Thai Texts	15M	25,000
SentencePiece	Thai Wikipedia	945k	24,000
newmm	-	-	97,982
PyThaiNLP syllable	-	-	59,235
SEFR (deepcut)	BEST	149k (lines)	92,177

Note that we lowercase for Assorted Thai Texts as we expect lower/uppercasing does not matter much for English words in Thai contexts.

Choose An Architecture

To NSP or not to NSP, that is the question

BERT-like models have 2 self-supervision tasks for pretraining: **masked language model (MLM)** and **next sentence prediction (NSP)**.

We have not found a definite gold standard for sentence boundaries and, as shown by the [CRF-based sentence segmenter CRFCut](#), sentence boundaries seem to be very domain dependent.

Our baseline model CRFCut achieves the following performance ⁹.

Training set	Validation set	Non-boundary token			Sentence boundary token			space-correct
		Precision	Recall	F1 score	Precision	Recall	F1 score	
TED	TED	0.99	0.99	0.99	0.74	0.70	0.72	0.82
TED	Orchid	0.95	0.99	0.97	0.73	0.24	0.36	0.73
TED	Product Review	0.98	0.99	0.98	0.86	0.70	0.77	0.78
Orchid	TED	0.98	0.98	0.98	0.56	0.59	0.58	0.71
Orchid	Orchid	0.98	0.99	0.99	0.85	0.71	0.77	0.87
Orchid	Product Review	0.97	0.99	0.98	0.77	0.63	0.69	0.70
Product Review	TED	0.99	0.95	0.97	0.42	0.85	0.56	0.56
Product Review	Orchid	0.97	0.96	0.96	0.48	0.59	0.53	0.67
Product Review	Product Review	1	1	1	0.98	0.96	0.97	0.97
TED + Orchid + Product Review	TED	0.99	0.98	0.99	0.66	0.77	0.71	0.78
TED + Orchid + Product Review	Orchid	0.98	0.98	0.98	0.73	0.66	0.69	0.82
TED + Orchid + Product Review	Product Review	1	1	1	0.98	0.95	0.96	0.96

Table 1: The precision, recall and F1 score for non-boundary and sentence boundary token of CRF-based sentence segmentor models trained and validated on different datasets. space-correct is accuracy of predicting if spaces are sentence boundaries or not.

Choose An Architecture

[RoBERTa](#) works well without relying on pre-tokenization nor next sentence prediction

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	94.6/89.4	90.2	96.4
BERT _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB \rightarrow 160GB of text) and pretrain for longer (100K \rightarrow 300K \rightarrow 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT_{LARGE}. Results for BERT_{LARGE} and XLNet_{LARGE} are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. Complete results on all GLUE tasks can be found in the Appendix.

It might require much larger datasets, which we have in Assorted Thai Texts.

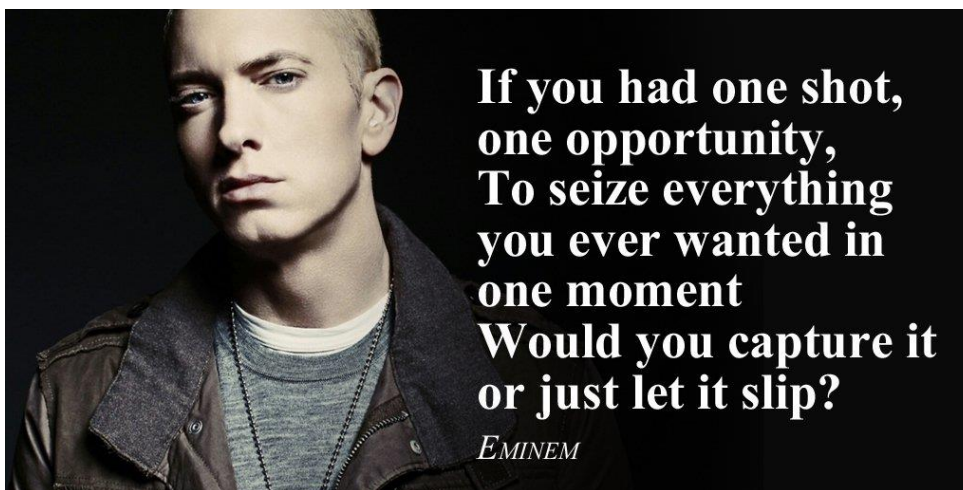
Training A Really, Really Large Language Model

How people think it goes



How it really went

488.5K steps in 134 days 2 hours



Model Inference

Masked Token Prediction

Input text: “ข้าวหน้าเนื้อ หรือเรียกเป็นภาษา<mask>ว่ากิวด้ง (Gyudon)”

Prediction: “ข้าวหน้าเนื้อ หรือเรียกเป็นภาษาญี่ปุ่นว่ากิวด้ง (Gyudon)”
(Top-4)
(word = ญี่ปุ่น , probability = 83.58)

“ข้าวหน้าเนื้อ หรือเรียกเป็นภาษาท้องถิ่นว่ากิวด้ง (Gyudon)”
(word = ท้องถิ่น , probability = 3.97)

“ข้าวหน้าเนื้อ หรือเรียกเป็นภาษาได้ว่ากิวด้ง (Gyudon)”
(word = ได้ , probability = 1.14)

“ข้าวหน้าเนื้อ หรือเรียกเป็นภาษากลางว่ากิวด้ง (Gyudon)”
(word = กลาง , probability = 1.12)

Training A Really, Really Large Language Model

- Training Instability
- No Train-on-Smaller-Dataset-First, No Retries
- Distributed Data Parallel (DDP) vs Data Parallel (DP)
- Memory Mapping for Large Datasets

Training Instability

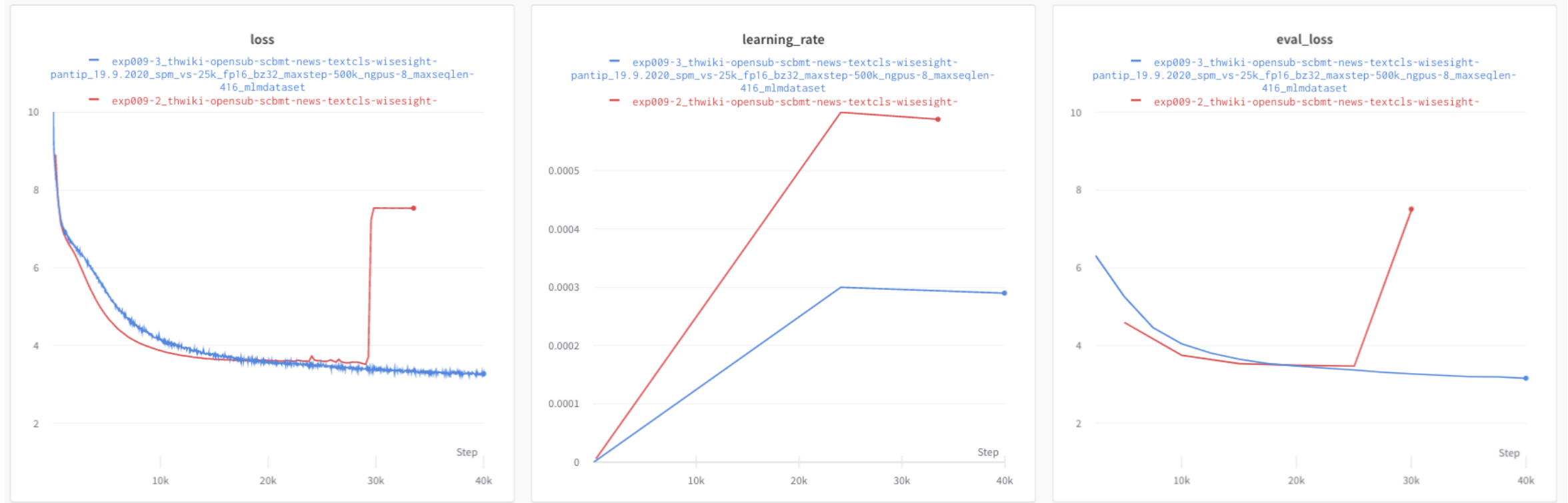
“We primarily follow the original BERT optimization hyperparameters, given in Section 2, except for the **peak learning rate** and **number of warmup steps**, which are **tuned separately** for each **setting**.

We additionally found **training to be very sensitive to the Adam epsilon term**, and in some cases we obtained better performance or improved stability after tuning it. Similarly, we found setting $\beta_2 = 0.98$ to improve stability when training with large batch sizes.”

Excerpt from the [RoBERTa paper](#)

Training Losses of Different Peak Learning Rates

At least two weeks spent to learn this lesson



Red: Peak learning rate = $6e-04$

Blue: Peak learning rate = $3e-04$

No Train-on-Smaller-Dataset-First, No Retries

We also trained on Thai Wikipedia to try out several tokenizers (words, syllables, SentencePiece subwords).

Hyperparameters	RoBERTa _{BASE} (Wikipedia-only Dataset)	RoBERTa _{BASE} (Assorted Thai Texts Dataset)
Number of Layers	12	12
Hidden size	768	768
FFN hidden size	3,072	3,072
Attention heads	12	12
Dropout	0.1	0.1
Attention dropout	0.1	0.1
Max sequence length	512	416
Effective batch size	8,192	4,092
Warmup steps	1,250	24,000
Peak learning rate	7e-4	3e-4
Learning rate decay	Linear	Linear
Max steps	31,250	500,000
Weight decay	0.01	0.01
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.98	0.999
FP16 training	True	True

Table 2: Hyperparameters of RoBERTa_{BASE} used when pretrain on *Assorted Thai Texts dataset* and *Wikipedia-only dataset*.

When training with Wikipedia-only Dataset, we did not find the stability issue despite learning rate of 7e-4, possibly due to larger effective batch size and/or smaller/less diverse training set.

This makes it difficult to anticipate and debug the models in "staging environment", thus forcing us to "test in production".

	Architecture	Dataset	Tokenizer
wangchanberta-base-wiki-spm	RoBERTa-base	Wikipedia-only	SentencePiece
wangchanberta-base-wiki-newmm	RoBERTa-base	Wikipedia-only	word (newmm)
wangchanberta-base-wiki-syllable	RoBERTa-base	Wikipedia-only	syllable (newmm)
wangchanberta-base-wiki-sefr	RoBERTa-base	Wikipedia-only	SEFR
wangchanberta-base-att-spm-uncased	RoBERTa-base	Assorted Thai Texts	SentencePiece

Table 3: WangchanBERTa model names

Distributed Data Parallel (DDP) vs Data Parallel (DP)

- In LM pretraining phase, we pretrain on 1x DGX1 (8 GPUs) with DataParallel (DP; single-process, multi-thread) strategy.
- However, the training time could be reduced if using DistributedDataParallel (DDP; multi-process).

Reference:

- PyTorch Tutorial: [Comparison between DataParallel and DistributedDataParallel](#)
- Lambda Lab : [A Gentle Introduction to Multi GPU and Multi Node Distributed Training](#)

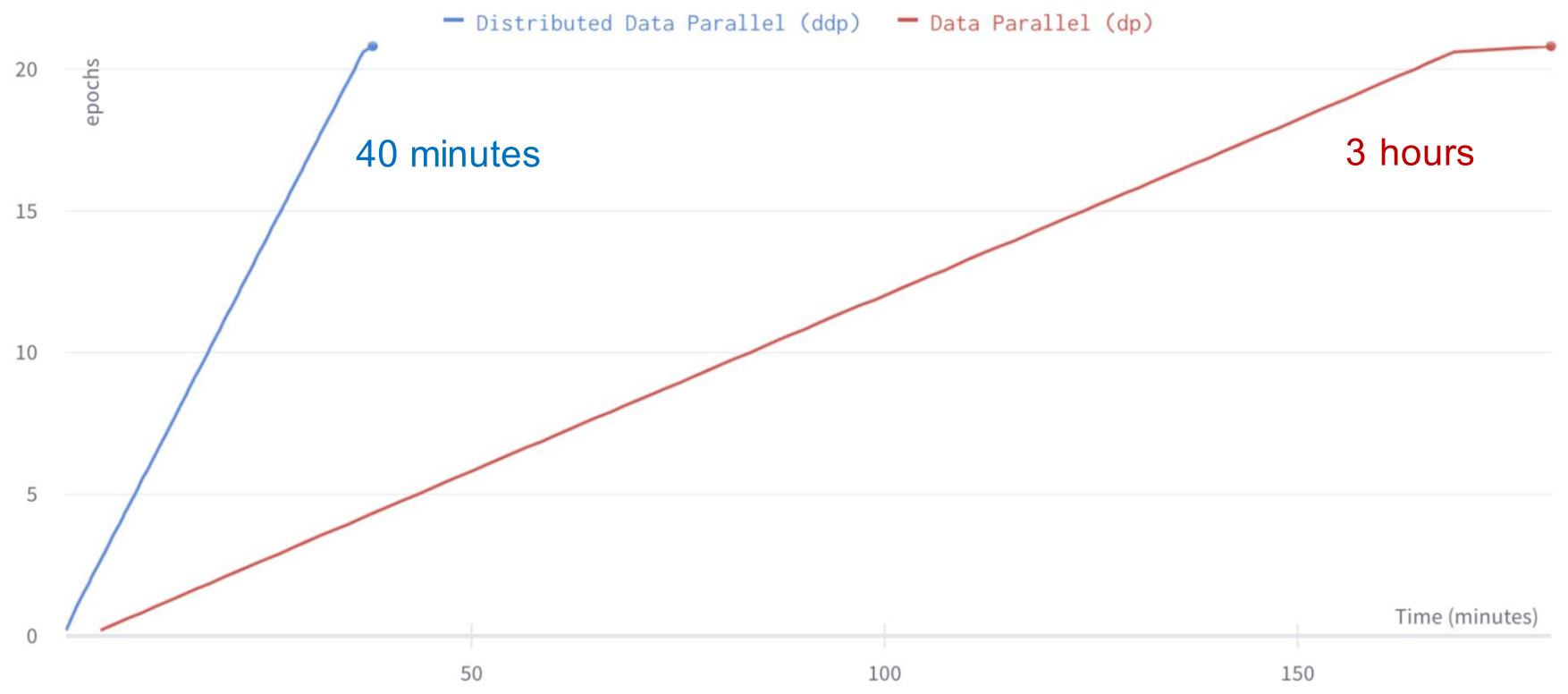
Distributed Data Parallel (DDP) vs Data Parallel (DP)

Training RoBERTa_{BASE} with DDP (blue) and DP (red) in 1 node of DGX1 (8x V100) for 500 steps (effective batch size is $32 * 16 * 8 = 4,096$).

Valid Loss

dp: 0.74660

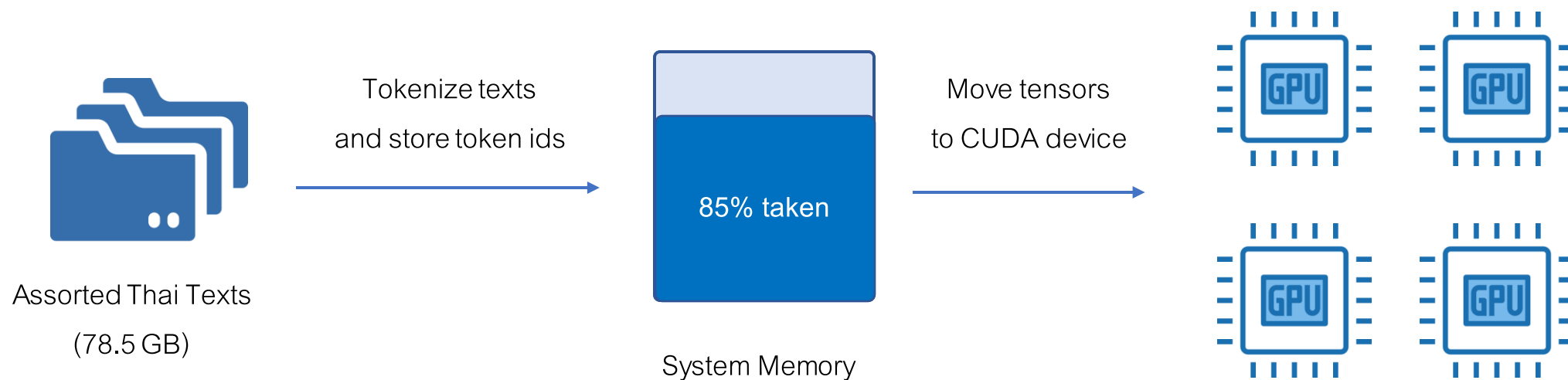
ddp: 0.72864



With DDP,
language model
pretraining got
4.5x speedup.

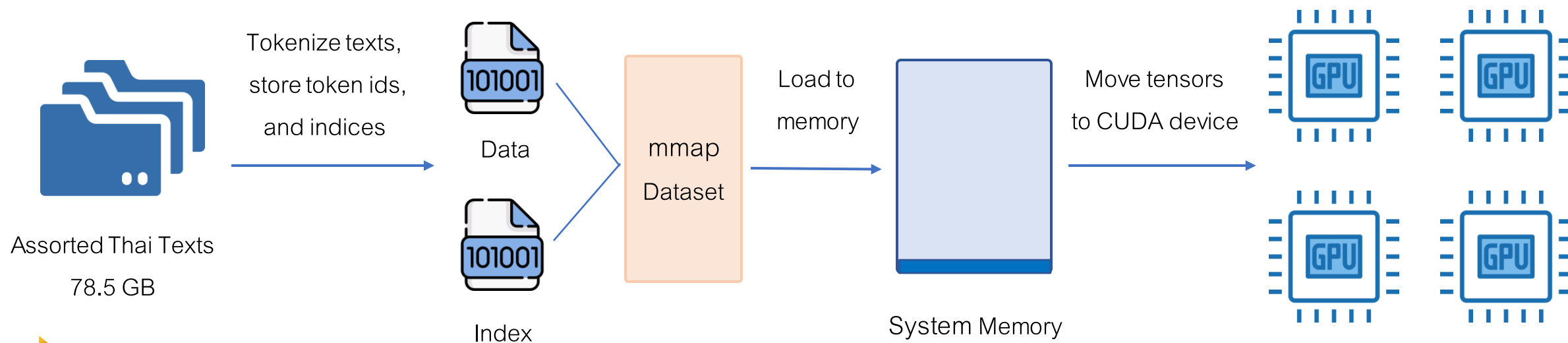
Memory Mapping for Large Datasets

- Huggingface's transformers (3.4.0) provides [LineByLineTextDataset](#). This module read the text file, tokenize all sentences, convert tokens into token ids, and store them into memory.
- However, this approach is not applicable for large datasets ([DGX1](#) has 512GB of system memory)



Memory Mapping for Large Datasets

- Our team implement a new Dataset module based on memory mapping [\[ref\]](#).
- With memory mapping, data loader can load data from the binary file only 1 chunk (or 1 minibatch) per model forward/backward.



Finetuning for Downstream Tasks

- Sequence classification (sentiment analysis, review classification, etc.)
- Token classification (POS and NER)
- Natural language inference (XNLI) **exclusive!**
- Question answering (iapp_wiki_qa_squad and thaiqa) **exclusive!**

HuggingFace's Datasets

- **Datasets** is a library with two main features; one-line public dataset loading, data preprocessing
- All the datasets used in our experiments is available at [HuggingFace Dataset Hub](#)

■ wongnai_reviews

Wongnai's review dataset contains restaurant reviews and ratings, mainly in Thai language. The reviews are in 5 classes ranging from 1 to 5 stars.

■ wisesight_sentiment

Wisesight Sentiment Corpus: Social media messages in Thai language with sentiment category (positive, neutral, negative, question) * Released to public domain under Creative Commons Zero v1.0 Universal license. * Category (Labels): {"pos": 0...

■ generated_reviews_enth

`generated_reviews_enth` Generated product reviews dataset for machine translation quality prediction, part of [scb-mt-en-th-2020](https://arxiv.org/pdf/2007.03541.pdf) `generated_reviews_enth` is created as part of [scb-mt-en-th-2020]...

■ prachathai67k

`prachathai-67k` : News Article Corpus and Multi-label Text Classification from Prachathai.com The prachathai-67k dataset was scraped from the news site Prachathai. We filtered out those articles with less than 500 characters of body te...

■ thainer

ThaiNER (v1.3) is a 6,456-sentence named entity recognition dataset created from expanding the 2,258-sentence [unnamed dataset](http://pioneer.chula.ac.th/~awirote/Data-Nutcha.zip) by [Tirasaroj and Aroonmanakun (2012)]...

■ lst20

LST20 Corpus is a dataset for Thai language processing developed by National Electronics and Computer Technology Center (NECTEC), Thailand. It offers five layers of linguistic annotation: word boundaries, POS tagging, named entities,...

■ iapp_wiki_qa_squad

`iapp_wiki_qa_squad` is an extractive question answering dataset from Thai Wikipedia articles. It is adapted from [the original iapp-wiki-qa-dataset](https://github.com/iapp-technology/iapp-wiki-qa-dataset) to [SQuAD]...

■ thaiqa_squad

`thaiqa_squad` is an open-domain, extractive question answering dataset (4,000 questions in `train` and 74 questions in `dev`) in [SQuAD](https://rajpurkar.github.io/SQuAD-explorer/) format, originally created by [NECTEC]...

Sequence Classification Task

Input: “เคยบ้าเฒ่าเคกับแม่ กินอาทิตย์ละ3-4 วันติด โศตรหนักและโคตรเปลืองงง”

Label: neg

Classes: {"pos": 0, "neu": 1, "neg": 2, "q": 3}

(from [Wisesight Sentiment Corpus](#)'s test set)



Preprocessed input: “เคยบ้าเฒ่าเคกับแม่ กินอาทิตย์ละ3-4 วันติด โศตรหนักและโคตรเปลือง”



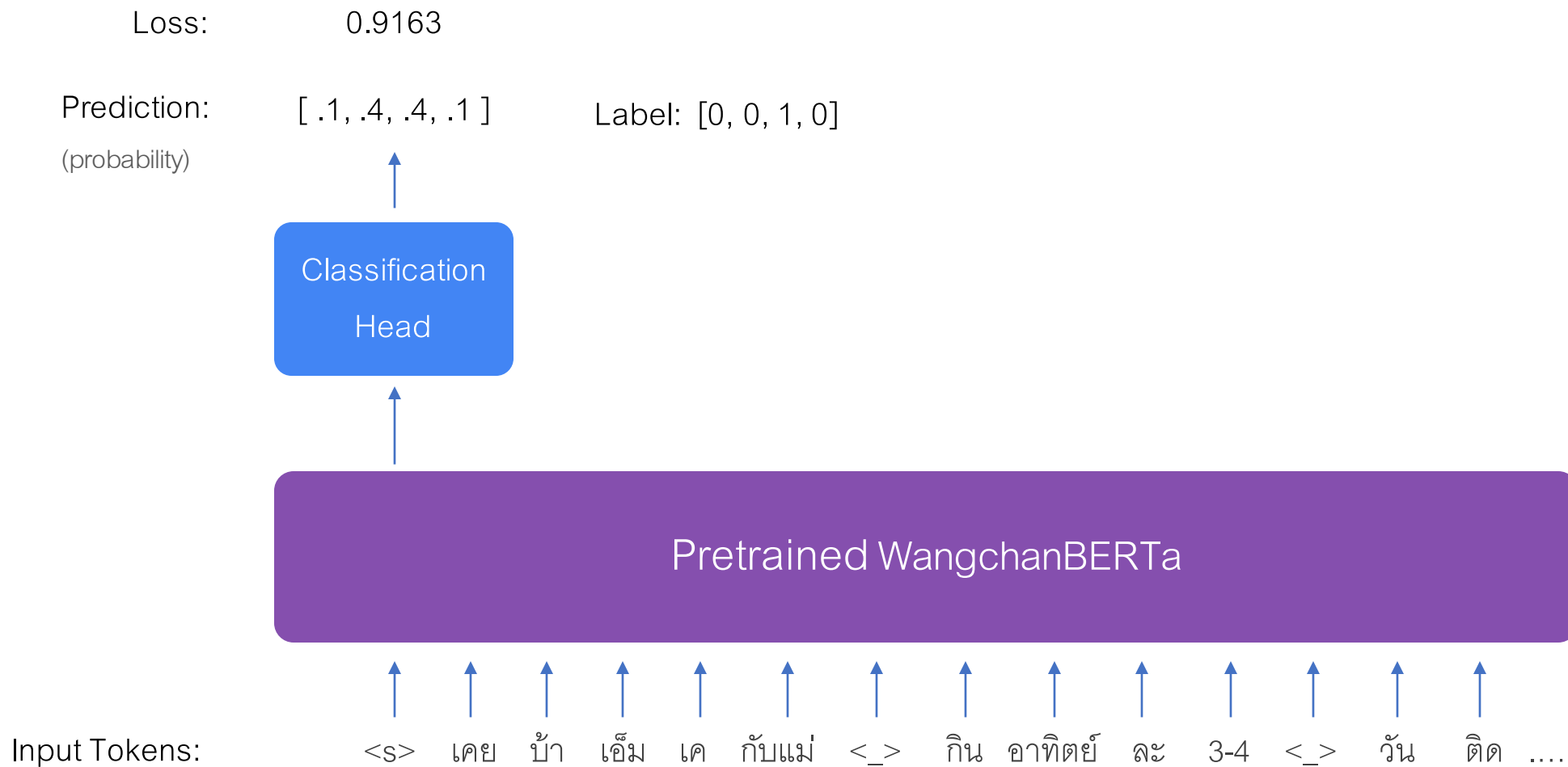
Input tokens: [“<s>”, “เคย”, “บ้า”, “เฒ่า”, “เค”, “กับแม่”, “<_>”, “กิน”, “อาทิตย์”, “ละ”, “3-4”, “<_>”, “วัน” “ติด”, “<_>”, “โศตร”, “หนัก”, “และ”, “โคตร”, “เปลือง”, “</s>”]

Input Ids: [5, 6193, 1442, 1837, 952, 7697, 11990, 1908, 153, 5068, 10, 125, 266, 10, 1982, 794, 13, 1982, 10147, 6, ...]

(padded to max sequence length)

Label Id: 2

Sequence Classification Task



Model Inference

Sequence

Classification

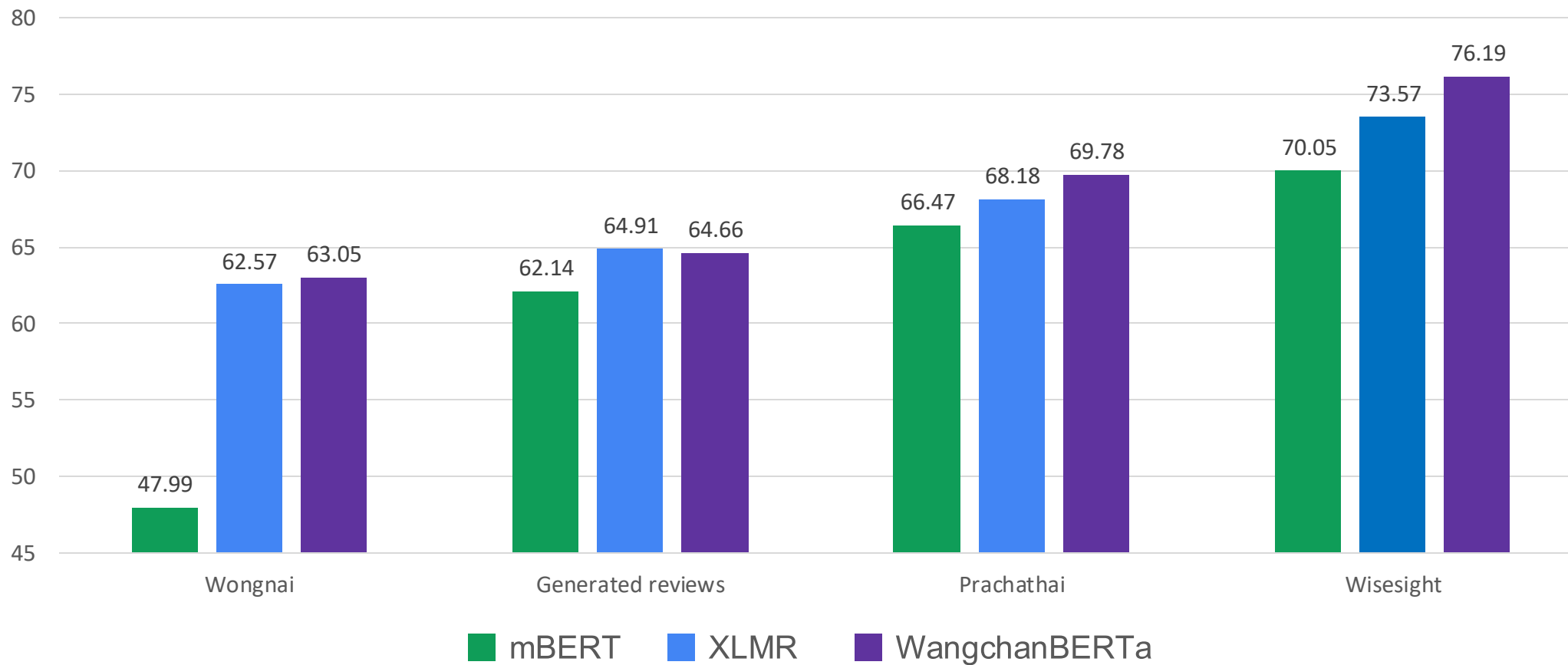
Input text: "ฟอร์ด บุกตลาด อีวี ในอินเดีย #prachachat #ตลาดรถยนต์"

Prediction: label: **Neutral** , score: 98.92

Input text: "สั่งไป 1 เมนู ไอศกรีมชาเขียว รสมันออกไปไม้ๆมากกว่าชาเขียว
แล้วก็หวานไป โดยรวมแล้วเฉยมากก ดีแคมีน้ำเปล่าบริการฟรี"

Prediction: label: **Negative** , score: 92.85

Sequence Classification Benchmarks



Sequence Classification Benchmarks

Model	Wisesight	Wongnai	Generated Reviews (Rating)	Prachathai
<i>Existing multilingual models</i>				
XLMR [Conneau et al.,2019]	73.57	62.57	64.91	68.18
mBERT [Devlin et al., 2018]	70.05	47.99	62.14	66.47
<i>Our baseline models</i>				
Navie Bays SVM	72.03	58.38	59.68	66.77
ULMFit (thai2fit)	70.95	61.79	64.33	66.21
<i>Our pretrained RoBERTa_{BASE} model</i>				
WangchanBERTa	76.19	63.05	64.66	69.78

Notes on Sequence Classification

- Uplifts most likely are not worth productionizing over NBSVM strong baselines.
- XLM-RoBERTa outperforms specifically in translated texts (generated_reviews_enth), pointing towards benefits of multilingual models in translated contexts.
- wiselight_sentiment has the highest uplifts due to social media being majority of our training data.

Token Classification Task

Human legible Inputs: “โรงเรียนสวนกุหลาบเป็นโรงเรียนที่ดี แต่ไม่มีสวนกุหลาบ”



(Words, Labels): [('โรงเรียน', 'B-ORG'), ('สวนกุหลาบ', 'I-ORG'), ('เป็น', 'O'), ('โรงเรียน', 'O'), ('ที่', 'O'), ('ดี', 'O'), (' ', 'O'), ('แต่', 'O'), ('ไม่', 'O'), ('มี', 'O'), ('สวนกุหลาบ', 'O')]

(Subwords, Labels): [('โรงเรียน', 'B-ORG'), ('สวน', 'I-ORG'), ('กุหลาบ', 'I-ORG'), ('เป็น', 'O'), ('โรงเรียน', 'O'), ...]

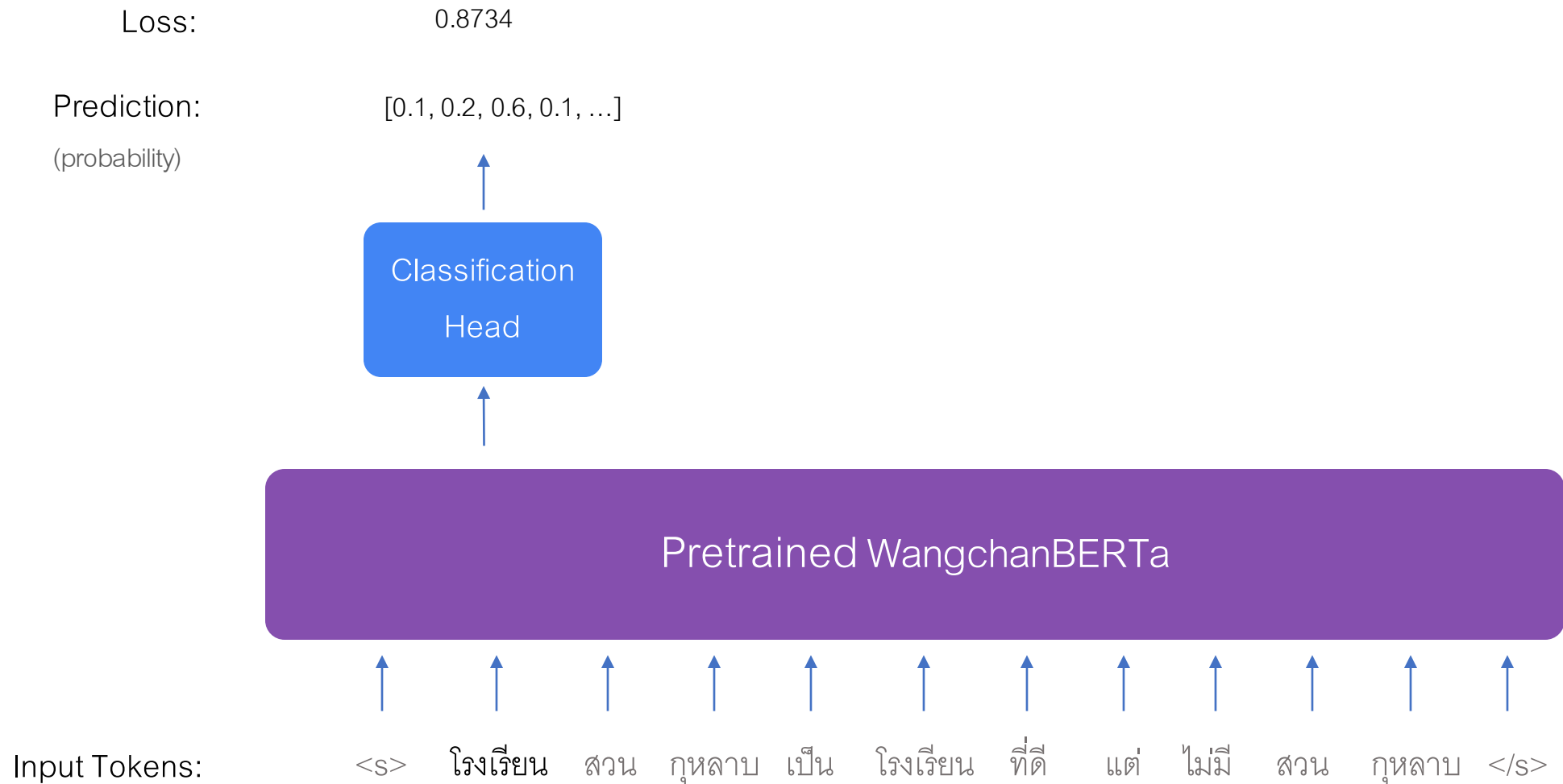
Classes: There are many tagging conventions for NER such as IOB (ThaiNER) and IOBE (LST20)



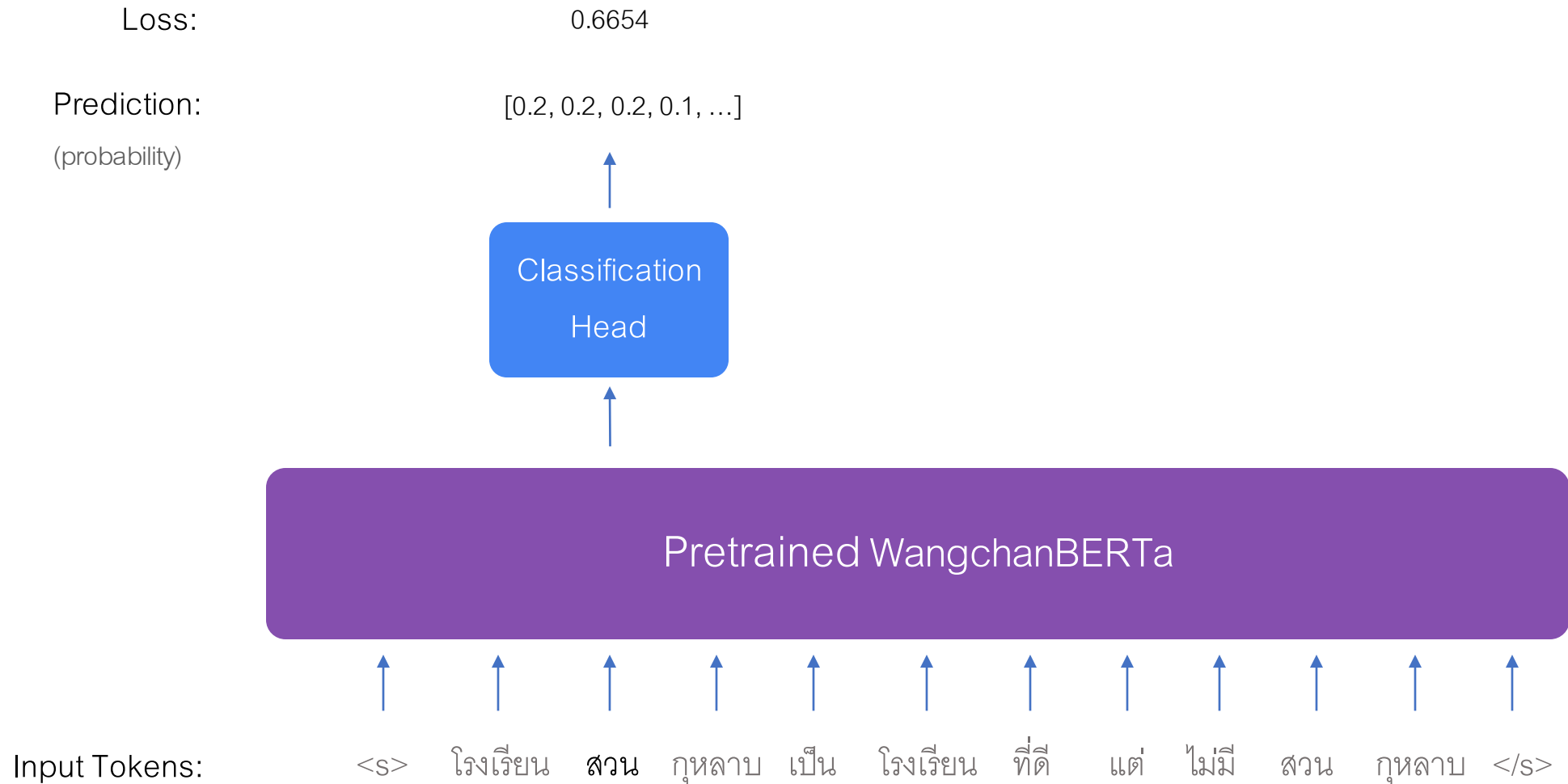
Input Ids: [5, 10, 377, 24542, 17, 377, 693, 10, 5320, 24542, 6, ...] #padded to max sequence length

Label Ids: [1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...] #predict <pad> as 'O'; do not calculate for metrics

Token Classification Task



Token Classification Task



Evaluating Token Classification with seqeval and More

Differing entity breaking aka $[('โรงเรียน', 'B-X')]$ vs $[('โรงเรียน', 'B-X'), ('เรียน', 'I-X')]$ will result in differing results, if we measure at tag level.

One solution is to measure performance at **entity level**:

- $(('โรงเรียน', 'B-X')) \rightarrow ('โรงเรียน', 'X')$
- $[('โรงเรียน', 'B-X'), ('เรียน', 'I-X')] \rightarrow ('โรงเรียน', 'X')$

Inferring Token Classifier in the Wild

- We need to pre-tokenize using the same tag-level tokenizer as our training set.
- Label propagation from tags to smaller subwords could lead to predicting two separate entities when there is only one:

e.g. [('โรงเรียน', 'B-ORG')] → [('โรง', 'B-ORG'), ('เรียน', 'B-PER')] → [('โรง', 'ORG'), ('เรียน', 'PER')]

Word Tags Subword Tags Entities

- [Custom NER pipeline](#) needed to apply post-processing.

```
[{'entity_group': 'ORGANIZATION', 'score': 0.782967746257782, 'word': ''},  
{ 'entity_group': 'ORGANIZATION',  
  'score': 0.9278752207756042,  
  'word': 'โรงเรียน'},  
{ 'entity_group': 'ORGANIZATION',  
  'score': 0.9350618720054626,  
  'word': 'สวนกุหลาบ'},  
{ 'entity_group': 'O',  
  'score': 0.8276164361408779,  
  'word': 'เป็นโรงเรียนที่ดี<_> แต่ไม่มีสวนกุหลาบ'}]
```



```
group_entities:  
[{'entity_group': 'LOCATION', 'word': 'โรงเรียนสวนกุหลาบ'},  
{ 'entity_group': 'O', 'word': 'เป็นโรงเรียนที่ดี แต่ไม่มีสวนกุหลาบ'}]
```

Inferring Token Classifier in the Wild

Input sentence: “เป็นรถไฟที่เชื่อมระหว่าง Keisei Ueno Station และสนามบินนาริตะโดยใช้ระยะเวลาเพียง 41 นาที”

Word-level entities:

```
[{'entity': 'O', 'word': 'เป็น'}, {'entity': 'O', 'word': 'รถไฟ'},  
{ 'entity': 'O', 'word': 'ที่'}, {'entity': 'O', 'word': 'เชื่อม'},  
{ 'entity': 'O', 'word': 'ระหว่าง'}, {'entity': 'O', 'word': ' '},  
{ 'entity': 'B_LOC', 'word': 'keisei'}, {'entity': 'I_LOC', 'word': ' '},  
{ 'entity': 'I_LOC', 'word': 'ueno'}, {'entity': 'I_LOC', 'word': ' '},  
{ 'entity': 'E_LOC', 'word': 'station'}, {'entity': 'O', 'word': ' '},  
{ 'entity': 'O', 'word': 'และ'}, {'entity': 'B_LOC', 'word': 'สนามบิน'},  
{ 'entity': 'E_LOC', 'word': 'นาริตะ'}, {'entity': 'O', 'word': 'โดย'},  
{ 'entity': 'O', 'word': 'ใช้'}, {'entity': 'O', 'word': 'ระยะเวลา'},  
{ 'entity': 'O', 'word': 'เพียง'}, {'entity': 'O', 'word': ' '},  
{ 'entity': 'B_MEAN', 'word': '41'}, {'entity': 'I_MEAN', 'word': ' '},  
{ 'entity': 'E_MEAN', 'word': 'นาที'}]
```

Chunk-level entities:

```
[{'entity_group': 'O', 'word': 'เป็นรถไฟที่เชื่อมระหว่าง '},  
{ 'entity_group': 'LOC', 'word': 'keisei ueno station'},  
{ 'entity_group': 'O', 'word': ' และ'},  
{ 'entity_group': 'LOC', 'word': 'สนามบินนาริตะ'},  
{ 'entity_group': 'O', 'word': 'โดยใช้ระยะเวลาเพียง '},  
{ 'entity_group': 'MEAN', 'word': '41 นาที'}]
```

Model Inference

Token

Classification

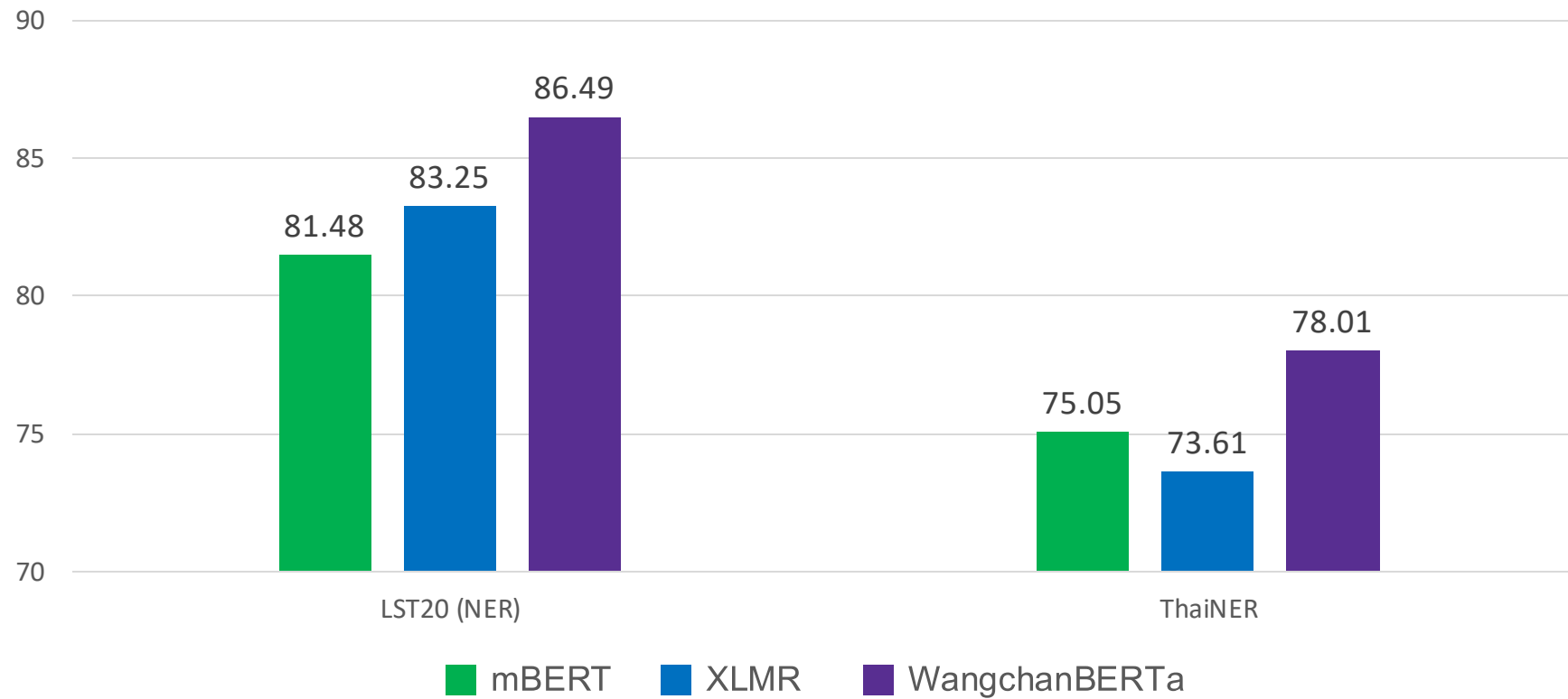
Input text: “เจนนี่เพิ่งย้ายมาใช้เอไอเอสได้ 2 เดือน”

Prediction: “เจนนี่”: PERSON , “เอไอเอส”: ORGANIZATION , “2 เดือน”: MEASUREMENT

Input text: “รฟม.คาดเดือนมี.ค.ได้ผู้ชนะประมูลรถไฟฟ้า”

Prediction: “รฟม.”: ORGANIZATION , “เดือนมี.ค.”: DATETIME

Token Classification Benchmarks



Token Classification Benchmarks

Model	ThaiNER (NER)	LST20 (POS)	LST20 (NER)
<i>Existing multilingual models</i>			
XLMR [Conneau et al.,2019]	83.25	96.57	73.61
mBERT [Devlin et al., 2018]	81.48	96.44	75.05
<i>Our baseline models</i>			
Conditional Random Fields (CRF)	78.98	96.28	75.94
<i>Our pretrained RoBERTa_{BASE} model</i>			
WangchanBERTa	86.49	96.62	78.01

Notes on Token Classification

- Better uplifts than sequence classification but still might not worth productionizing.
- POS is most likely too easy to use transformers for.
- Mapping subwords to entity labels is a problem for both training inputs and inference in the wild.

Natural Language Inference Task

Input: {'premise': 'สนุกสำหรับผู้ใหญ่และเด็กค่ะ', 'hypothesis': 'สนุกสำหรับเด็กเท่านั้น'}

Label: **contradiction**

Classes Available: {"entailment": 0, "neutral": 1, "contradiction": 2}



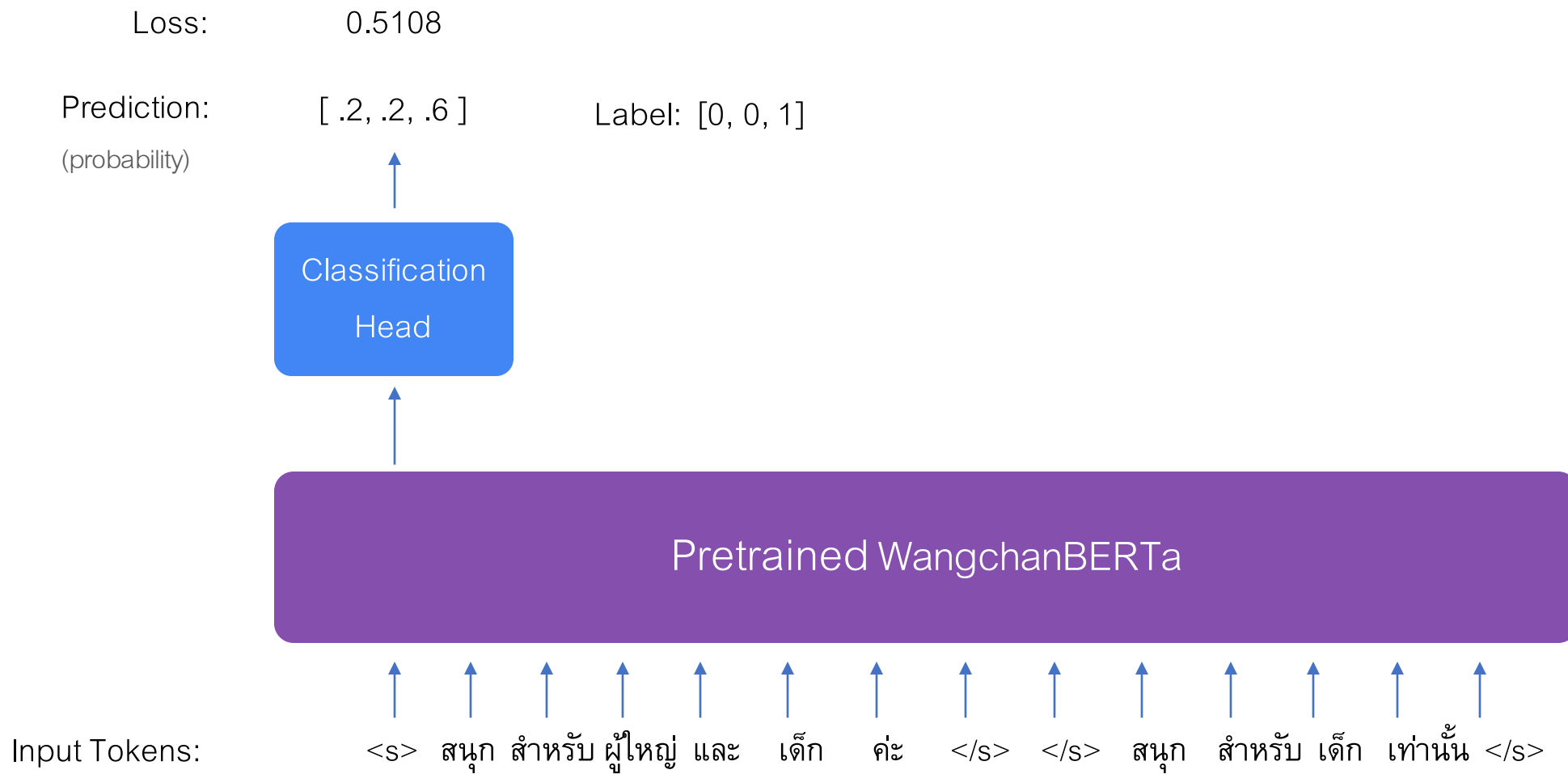
Processed Input: <s> สนุก สำหรับ ผู้ใหญ่ และ เด็ก ค่ะ </s> </s> สนุก สำหรับ เด็ก เท่านั้น </s>

Input Ids: [5, 10, 1307, 1584, 10, 1231, 222, 6345, 10, 70, 6, 6, 10, 1307, 1584, 6345, 10, 268, 6, ...]

(padded to max sequence length)

Label ID: 2

Sequence Classification Task

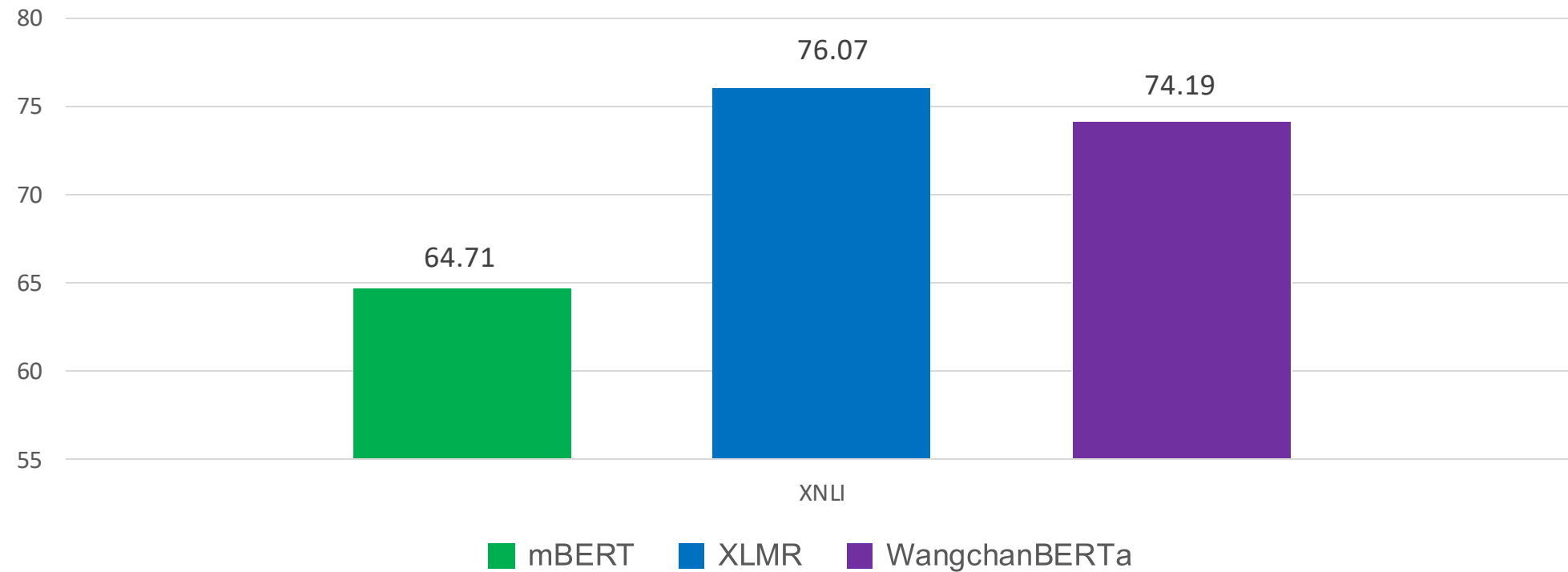


Benchmark on Cross-Lingual NLI Corpus ([XNLI](#))

Model	Accuracy
<i>Existing multilingual models</i>	
XLMR [Conneau et al.,2019]	76.07
mBERT [Devlin et al., 2018]	64.71
<i>Our pretrained RoBERTa_{BASE} model</i>	
WangchanBERTa	74.19

*Remark: We finetune from the pretrained models on Thai translated sentences of the XNLI training set and evaluate on the test set (human annotation)

Benchmark on Cross-Lingual NLI Corpus ([XNLI](#))



*Remark: We finetune from the pretrained models on Thai translated sentences of the XNLI training set and evaluate on the test set (human annotation)

Notes on Natural Language Inference

- XNLI (train set) is a machine-translated dataset; we hypothesize XLM-RoBERTa has the advantage.

Set	Premise	Hypothesis	Label
train	One of our number will carry out your instructions minutely .	A member of my team will execute your orders with immense precision .	Entailment
	หนึ่งในหมายเลขของเรา จะ ดำเนินการ ให้ คำแนะนำ ของ คุณ minutely	สมาชิก ใน ทีม ของ ฉัน จะ ทำ ตามคำสั่ง ของ คุณ ด้วย ความแม่นยำ อัน ยิ่งใหญ่	
dev	I didn't know what I was going for or anything, so was to report to a designated place in Washington.	I was not quite certain what I was going to do so I went to Washington where I was assigned to report.	Entailment
	ฉันไม่รู้ว่าจะไปเพื่ออะไรหรือเพื่อสิ่งใด ดังนั้นแค่รายงานเกี่ยวกับ สถานที่ที่ระบุในวอชิงตัน	ฉันไม่ค่อยมั่นใจว่าฉันจะทำอะไร ดังนั้นฉันจึงไปวอชิงตันที่ซึ่งฉันได้รับ มอบหมายให้ทำงาน	

Notes on Natural Language Inference

Nonetheless, finetuning on NLI helps with feature extraction and zero-shot classification.

Input: “นักวิจัยนาโนเทคโนโลยีจากสถาบันวิทยสิริเมธี ผู้คิดค้นแบตเตอรี่จากวัสดุกราฟีน คว่ำรางวัลนักวิทยาศาสตร์ดีเด่น ประจำปี 2562”

```
{ 'labels': ['เทคโนโลยี', 'การเมือง', 'ศิลปะ-บันเทิง', 'เศรษฐกิจ-ธุรกิจ'],  
  'scores': [0.7019814252853394,  
             0.12360921502113342,  
             0.10269065946340561,  
             0.07171864807605743],  
  'sequence': 'นักวิจัยนาโนเทคโนโลยี<_>จากสถาบันวิทยสิริเมธี<_>ผู้คิดค้นแบตเตอรี่จากวัสดุกราฟีน<_>คว่ำรางวัลนักวิทยาศาสตร์ดีเด่น<_>ประจำปี<_>2562' }
```

Input: “SCB 10X ร่วมลงทุนใน BlockFi Startup ด้าน Digital Asset”

```
{ 'labels': ['เศรษฐกิจ-ธุรกิจ', 'เทคโนโลยี', 'การเมือง', 'ศิลปะ-บันเทิง'],  
  'scores': [0.3542027771472931,  
             0.30350786447525024,  
             0.1763230562210083,  
             0.16596631705760956],  
  'sequence': 'scb<_>10x<_>ร่วมลงทุนใน<_>blockfi<_>startup<_>ด้าน<_>digital<_>asset' }
```

Question Answering Task – Human Legible Input

Objective: find **start and end token/character positions** for answer to 'question' in 'context'

Input:

```
{'question_id': '11Wa2yX4q3jaPbhSQBmb_000',
```

```
  'question': 'ไซราคุเซียเป็นชื่อเรียกของอะไรในสมัยกรีกโบราณ',
```

```
  'answers': {'answer_end': [42], 'answer_start': [38], 'text': ['เรือ']},
```

```
  'context': 'ไซราคุเซีย (อังกฤษ: Syracuse) คือชื่อเรือในสมัยกรีกโบราณ ซึ่งบางครั้งกล่าวกันว่าเป็นเรือโดยสารที่ใหญ่  
ที่สุดในโลกยุคโบราณผู้ออกแบบคือ อาร์คิมิดีส สร้างขึ้นราวปีที่240 ก่อนคริสตกาล โดยอาร์เซียสแห่งโครินธ์ ตาม  
คำสั่งของพระเจ้าเฮียโรที่2 แห่งซีราคิวส์ นักประวัติศาสตร์ชื่อ มอสซิออน แห่งฟาเซลิส กล่าวว่า ไซราคุเซีย สามารถ  
บรรทุกสินค้าได้ถึง 1,600 - 1,800 ตัน บรรทุกทหารได้ 200 คนรวมทั้งเครื่องยิงหินเคยออกเดินเรือเพียงครั้งเดียวไปยัง  
เมืองเบิร์ธในอเล็กซานเดรีย หลังจากนั้นก็ถูกมอบให้แก่พระเจ้าทอเลมีที่3 ยูร์เจเทส แห่งอียิปต์ และเปลี่ยนชื่อไปเป็น  
อเล็กซานดริส มีการอธิบายเกี่ยวกับเรือนี้รวมถึงบทความชุดสมบูรณ์ของอาเธนาอุส (นักเขียนกรีกช่วงปลาย  
ศตวรรษที่ 2 ผู้อ้างอิงคำบรรยายเรือ ไซราคุเซีย จากมอสซิออน ซึ่งเป็นนักเขียนยุคก่อนหน้านี้ที่ผลงานหายสาบสูญไป  
แล้ว) ปรากฏอยู่ในงานเขียนของคัสสันเรื่อง Ships and Seamanship in the Ancient World.'}
```

Question Answering Task – Processed Input

Processed Input; in case concatenated question-contexts are too long,
we break each question into several examples (max length = 128, doc stride = 50):

1. <s> ไชราคุเซียเป็นชื่อเรียกของอะไรในสมัยกรีกโบราณ</s></s> ไชราคุเซีย (อังกฤษ: syracusia) คือชื่อเรือในสมัยกรีกโบราณซึ่งบางครั้งกล่าวกันว่าเป็นเรือโดยสารที่ใหญ่ที่สุดในโลกยุคโบราณ ผู้ออกแบบคือ อาร์คิมิดีส สร้างขึ้นราวปีที่ 240 ก่อนคริสตกาล โดยอาร์เซียสแห่งโครินธ์ ตามคำสั่งของพระเจ้าเฮียโรที่ 2 แห่งซีราคิวส์ นักประวัติศาสตร์ชื่อ มอสซิออน แห่งฟาเซลิส กล่าวว่า ไชราคุเซีย สามารถบรรทุกสินค้าได้ถึง 1,600 - 1,800 </s>
2. <s> ไชราคุเซียเป็นชื่อเรียกของอะไร ในสมัยกรีกโบราณ</s></s> ตามคำสั่งของพระเจ้าเฮียโรที่ 2 แห่งซีราคิวส์ นักประวัติศาสตร์ชื่อ มอสซิออน แห่งฟาเซลิส กล่าวว่า ไชราคุเซีย สามารถบรรทุกสินค้าได้ถึง 1,600 - 1,800 ตัน บรรทุกทหารได้ 200 คนรวมทั้งเครื่องยิงหิน เคยออกเดินเรือเพียงครั้งเดียวไปยังเมืองเปิร์ธ ในอเล็กซานเดรีย หลังจากนั้นก็ถูกมอบให้แก่พระเจ้าทอเลมีที่ 3 ยูร์เจเทส แห่งอียิปต์ และเปลี่ยนชื่อไปเป็น อเล็กซานดริส มีการอธิบายเกี่ยวกับเรือนี้ รวมถึงบทความ</s>

Question Answering Task – Modeling Objective

Input Ids: [

[5, 10, 1901, 104, 9, 2135, 4365, 17, 18516, 16, 118, 10, 3110, 9246, 2235, 6, 6, 10, 1901, 104, ...],

[5, 10, 1901, 104, 9, 2135, 4365, 17, 18516, 16, 118, 10, 3110, 9246, 2235, 6, 6, 567, 1672, 6193, ...],

...] #one question can be broken into several examples; for examples without a possible answer, model will predict BOS token

Label Ids: {'start_token_position': 20, 'end_token_position': 25]

Predictions: [[0.1, 0.01, 0.8, ...], [0.01, 0.2, 0.06, 0.1, ...]]

#Softmax predictions over all tokens for start and end token position

Loss: Cross-entropy loss

(Question-context, (Start Token Position, End Token Position)) >> Pretrained Model

>> Classification Head >> (Predictions, Loss)

Question Answering Task – Evaluation

- We select `n_best_size` start and end positions according to their logits for each example.
- For those start and end position pairs, we select only those that
 - end positions come after start position
 - has text length no more than `max_answer_length`
- We select the pair with highest summed logits to be the answer for that example.
- Lastly, we map examples back to their original questions; for questions with several answers, we choose the answer with highest summed logits.

Model Inference

Question Answering

Input context:

” ชวงชวง (Chuàng Chuàng) เป็นชื่อของแพนด้ายักษ์เพศผู้ ที่สาธารณรัฐประชาชนจีน ให้ประเทศไทยยืมจัดแสดงที่ สวนสัตว์เชียงใหม่ ในฐานะทูตสันถวไมตรีไทย-จีน โดยจัดแสดงคู่กับ หลินฮุ่ย แพนด้ายักษ์เพศเมีย ตั้งแต่เดือนตุลาคม พ.ศ. 2546 มีชื่อไทยว่า "เทวัญ" และมีชื่อล้านนาว่า "คำอ้าย" ชวงชวง เกิดเมื่อวันที่ 6 สิงหาคม พ.ศ. 2543 ที่ศูนย์วิจัยและอนุรักษ์แพนด้ายักษ์เขตอนุรักษ์ภูหลวง เมืองเชินตู มณฑลเสฉวน สาธารณรัฐประชาชนจีน”

Input question: ”ประเทศไทยรับชวงชวงมาจัดแสดงตั้งแต่ปีอะไร”

Prediction: sequence: “พ.ศ. 2546” , score: 90.75
(Top-3)

sequence: “2546” , score: 8.25

sequence: “พ.ศ. 2543” , score: 1.96

Benchmark on `iapp_wiki_qa_squad`

- We combine training sets of `iapp_wiki_qa_squad` and `thaiqa`, then [validate and test](#) on `iapp_wiki_qa_squad`; we removed all questions containing contexts similar to validation and test sets.
- We use SQuAD v2's [evaluate.py](#) but change normalization function to `process_transformers` and tokenization from split at spaces to `newmm`

model	test-set_exact_match	test-set_f1
mBERT	39.378	52.89
XLMR	52.639	76.002
WangchanBERTa	47.497	71.898

Notes on Question Answering Task

- Being able to handle longer sequences clearly has an advantage in question answering tasks; mBERT performed poorly because its subwords are too small, leading to 2x number of training examples, most of which have no answer (if we evaluate with squad v2 standards).
- Error analysis to be performed to find out why monolingual model WangchanBERTa underperforms multilingual model XLMR; a possible explanation could be Thai Wikipedia has a lot of translated texts.



Summary

- Provide building blocks for the Thai NLP community
- Encourage the development of applications integrated with Thai NLP
- Increase the capability of Thai NLP

Hands-on workshop: WangchanBERTa



Pretrain

Learn from large amount
of monolingual text corpus



Finetune

Tune pretrained model on
downstream tasks



Evaluate

Test the performance of
our finetuned models



Deploy

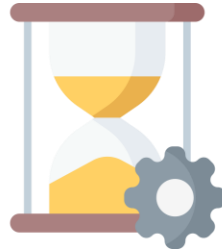
Let's employ
the new capability

Hands-on workshop: WangchanBERTa



Pretrain

Learn from large amount
of monolingual text corpus



Finetune

Tune pretrained model on
downstream tasks



Evaluate

Test the performance of
our finetuned models



Deploy

Let's employ
the new capability

Hands-on workshop: WangchanBERTa

1. Train two models for sentiment analysis on [Wisesight Sentiment](#) dataset
 - Train Naïve Bays SVM based on TFIDF features (Baseline model)
 - Finetune WangchanBERTa from pretrained weights
2. Model inference
3. Evaluate on the test set of *wisesight_sentiment*

Hands-on workshop: WangchanBERTa model finetuning

- [Hands-on workshop: WangchanBERTa model finetuning](#)
- [Question Answering Workshop](#); **Bonus points**: find out which types of questions WangchanBERTa underperforms XLMR

Useful Links

Medium blog: bit.ly/39i2JzR

Technical report: arxiv.org/abs/2101.09635

Getting started notebook: [Colab Notebook](#)

Huggingface Model Hub: huggingface.co/airesearch

GitHub repository (thai2transformers): github.com/vistec-AI/thai2transformers



Q&A