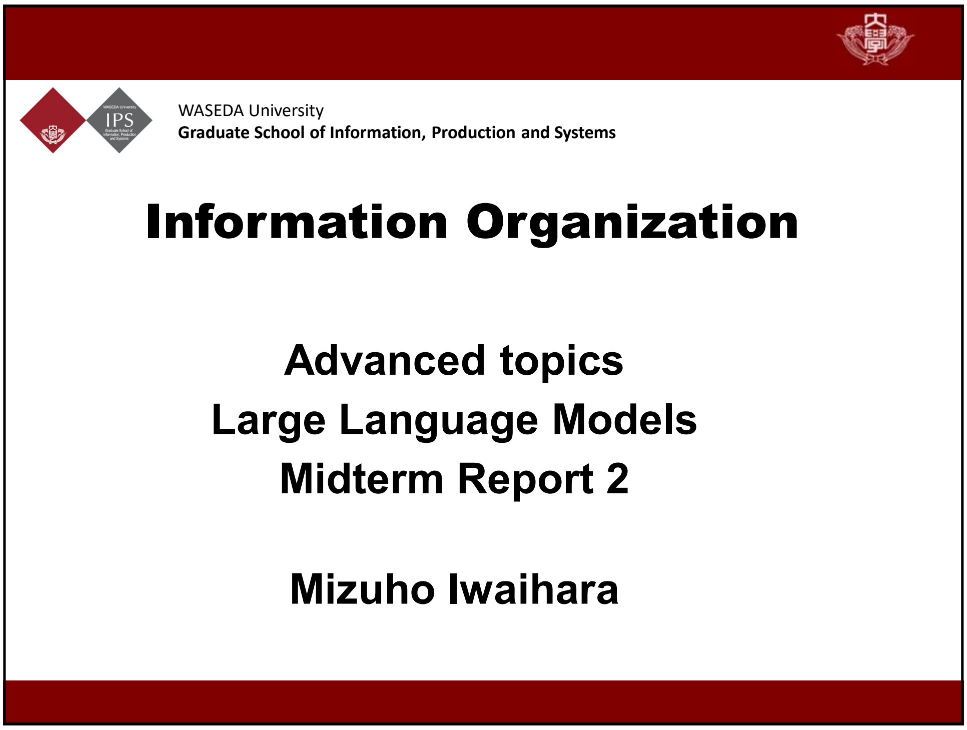
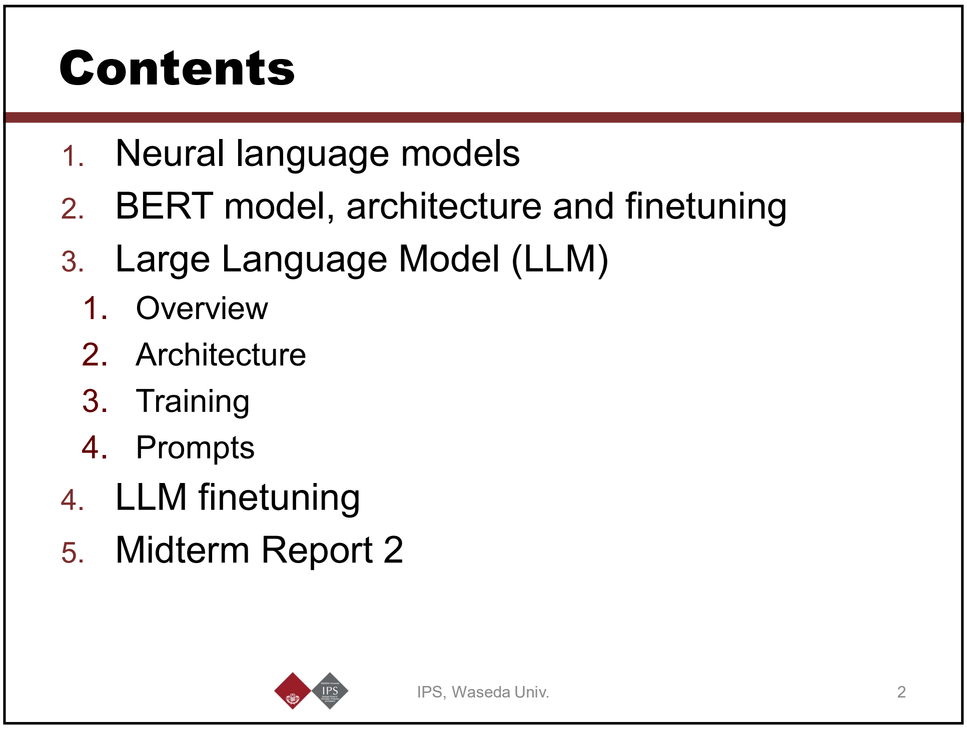


WASEDA University
Graduate School of Information, Production and Systems

Information Organization

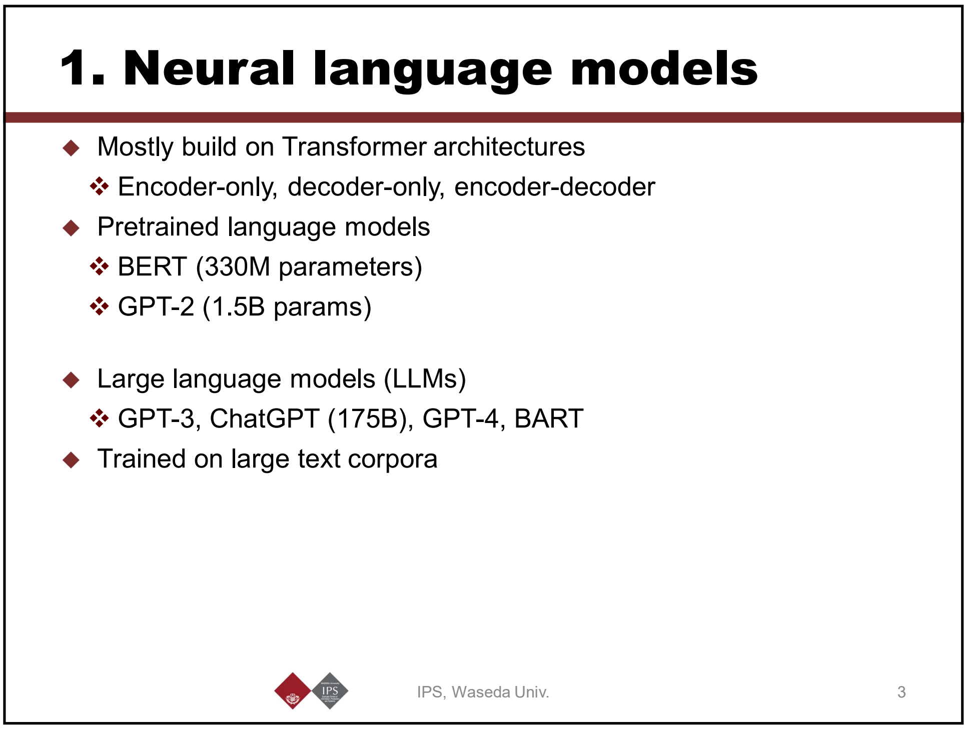
Advanced topics
Large Language Models
Midterm Report 2

Mizuho Iwaihara



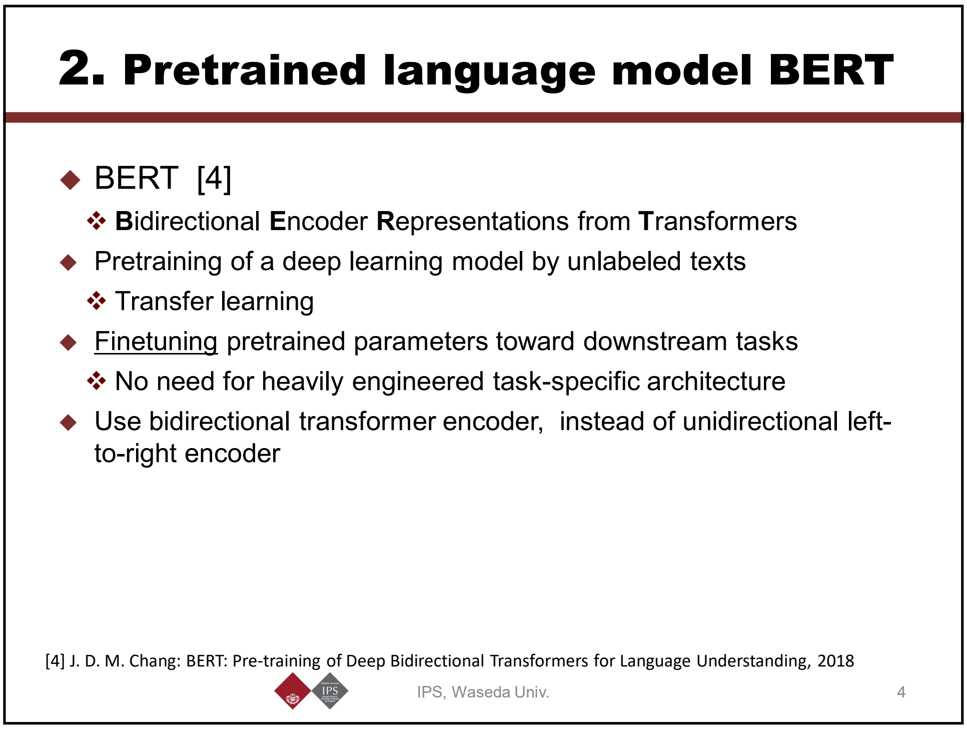
Contents

1. Neural language models
2. BERT model, architecture and finetuning
3. Large Language Model (LLM)
   1. Overview
   2. Architecture
   3. Training
   4. Prompts
4. LLM finetuning
5. Midterm Report 2



1. Neural language models

- Mostly build on Transformer architectures
  - Encoder-only, decoder-only, encoder-decoder
- Pretrained language models
  - BERT (330M parameters)
  - GPT-2 (1.5B params)
- Large language models (LLMs)
  - GPT-3, ChatGPT (175B), GPT-4, BART
- Trained on large text corpora



2. Pretrained language model BERT

- BERT [4]
  - **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- Pretraining of a deep learning model by unlabeled texts
  - Transfer learning
- Finetuning pretrained parameters toward downstream tasks
  - No need for heavily engineered task-specific architecture
- Use bidirectional transformer encoder, instead of unidirectional left-to-right encoder

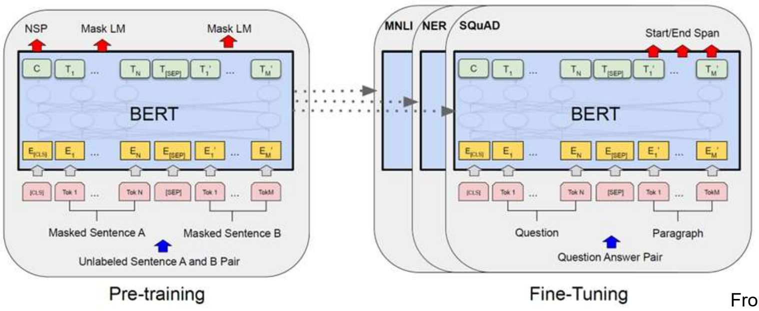
[4] J. D. M. Chang: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

Pretrained language model

- BERT [4] Pretraining objectives:
  - 1) Masked language model (MLM)
    - Randomly masks some of the tokens from the input, and the objective is to predict the original token based on its context
  - 2) Next sentence prediction (NSP)
  - No need for labeled (annotated) texts
  - Large unlabeled corpora are used for pretraining.

[4] J. D. M. Chang: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018

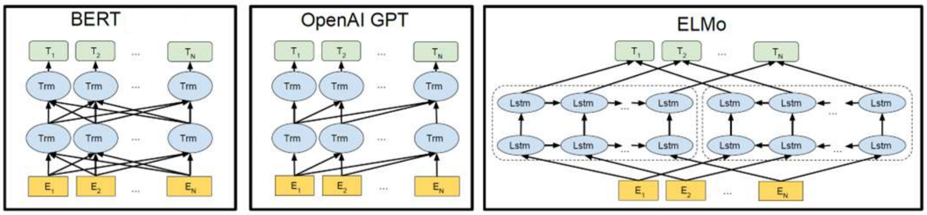
Pretraining and finetuning



From [4]

- Except output layers, the same architectures for pre-training and finetuning.
- The same pretrained model parameters are used to initialize models for different down-stream tasks.
- All parameters are finetuned.
- Relatively small amount of task-specific training samples are used for finetuning.

Language model architectures



From [4]. Tm: Transformer.

- BERT: bidirectional Transformer
- Right and left context in all layers
- Finetuning by MMN and NSP

- OpenAI GPT: Left-to-right Transformer
- Left context in all layers
- Finetuning by next word prediction

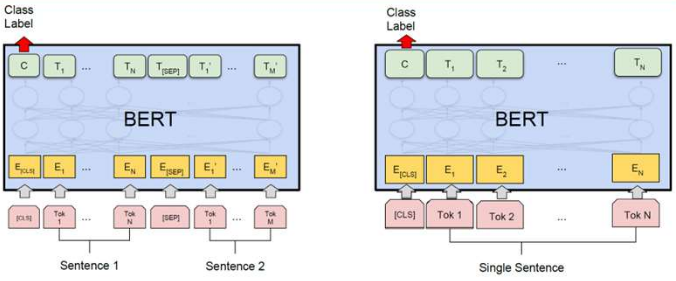
- ELMo:
- Concatenation of right-to-left and left-to-right LSTMs
- Feature-based training

BERT model size and input

- $\text{BERT}_{\text{BASE}}$ (L=12, H=768, A=12, Total Parameters= 110M)
- $\text{BERT}_{\text{LARGE}}$ (L=24, H=1024, A=16, Total Parameters=340M).
  - L: the number of layers (i.e., Transformer blocks)
  - H: the hidden size
  - A: the number of self-attention heads

- BERT training corpus
  - BooksCorpus (800M words)
  - English Wikipedia (2,500M words)
    - Documents, not sentence, are used to learn contexts.

Single-sentence and sentence pair tasks



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

From [4]

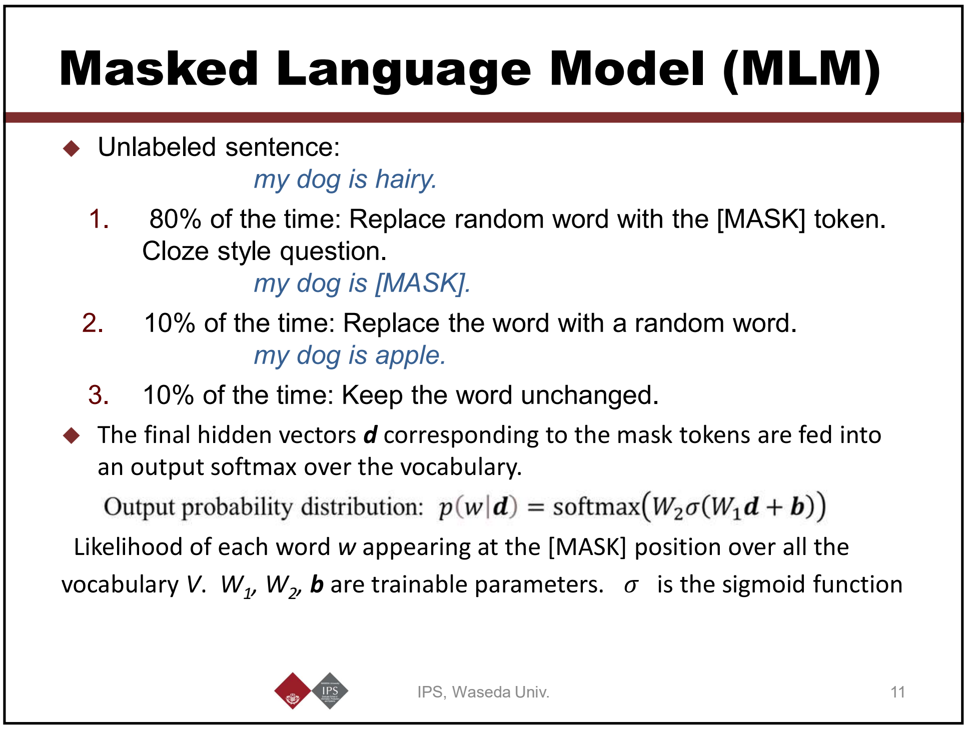
- BERT achieved state-of-the-art records (best) on NLP tasks in 2018.
  - GLUE tasks
    - Sentence-pair classification: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG
    - Single sentence classification tasks: SST-2, CoLA

BERT Input representation

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

From [4]

- Input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.
- Token sequence are WordPeice embeddings (30,000 token vocabularies)
- [CLS] is a special symbol added in front of every input example
  - Final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.
- [SEP] is a special separator token (e.g. separating questions/answers).



Masked Language Model (MLM)

- Unlabeled sentence:

  my dog is hairy.

1. 80% of the time: Replace random word with the [MASK] token. Cloze style question.

   my dog is [MASK].

2. 10% of the time: Replace the word with a random word.

   my dog is apple.

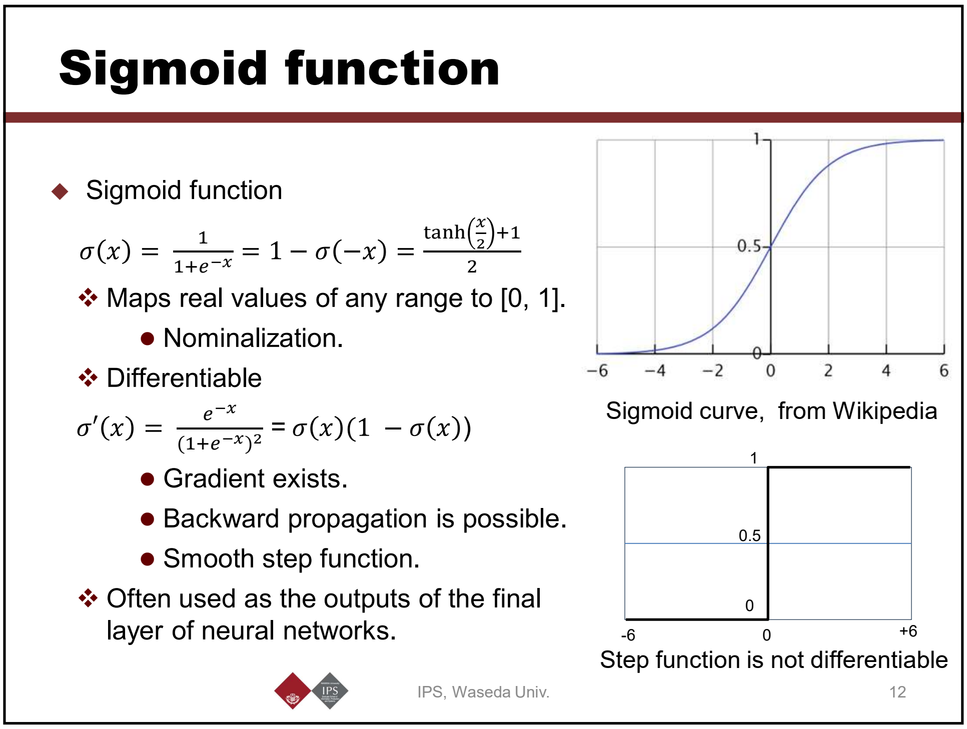
3. 10% of the time: Keep the word unchanged.

- The final hidden vectors ***d*** corresponding to the mask tokens are fed into an output softmax over the vocabulary.

Output probability distribution: $p(w|\boldsymbol{d}) = \mathrm{softmax}\left(W_2\sigma(W_1\boldsymbol{d} + \boldsymbol{b})\right)$

Likelihood of each word *w* appearing at the [MASK] position over all the vocabulary *V*. W_1, W_2, ***b*** are trainable parameters. σ is the sigmoid function

IPS, Waseda Univ. 11



Sigmoid function

- Sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}} = 1 - \sigma(-x) = \frac{\tanh\left(\frac{x}{2}\right)+1}{2}$$

- Maps real values of any range to [0, 1].
  - Nominalization.
- Differentiable

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \sigma(x)(1 - \sigma(x))$$

  - Gradient exists.
  - Backward propagation is possible.
  - Smooth step function.
- Often used as the outputs of the final layer of neural networks.

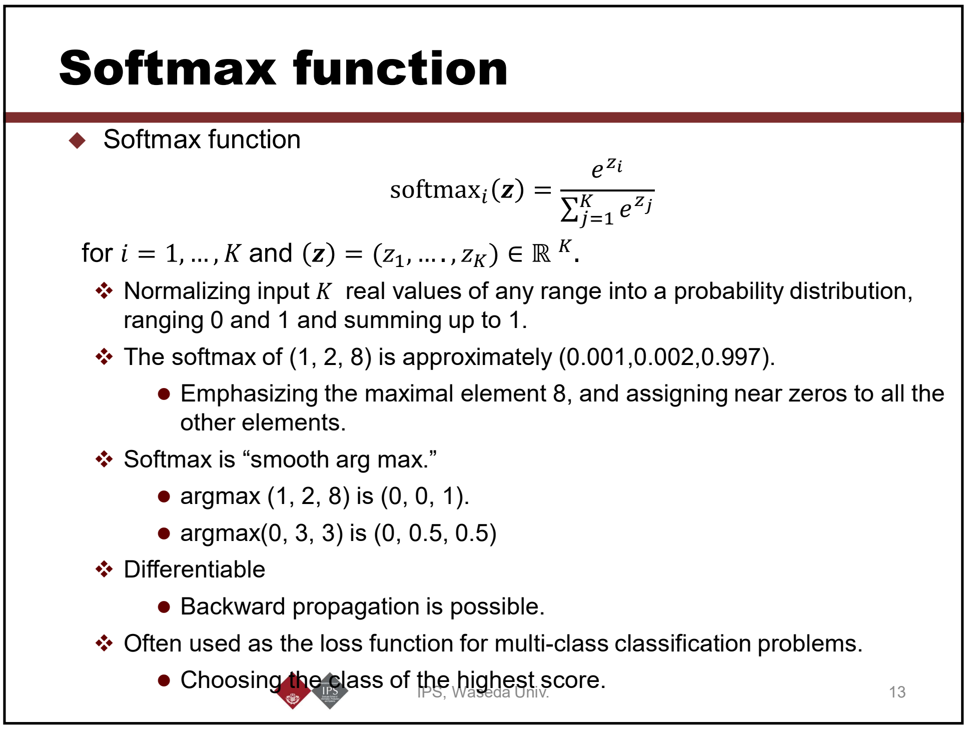
1
0.5
0
−6
−4
−2
0
2
4
6

Sigmoid curve, from Wikipedia

1
0.5
0
-6
0
+6

Step function is not differentiable

IPS, Waseda Univ. 12



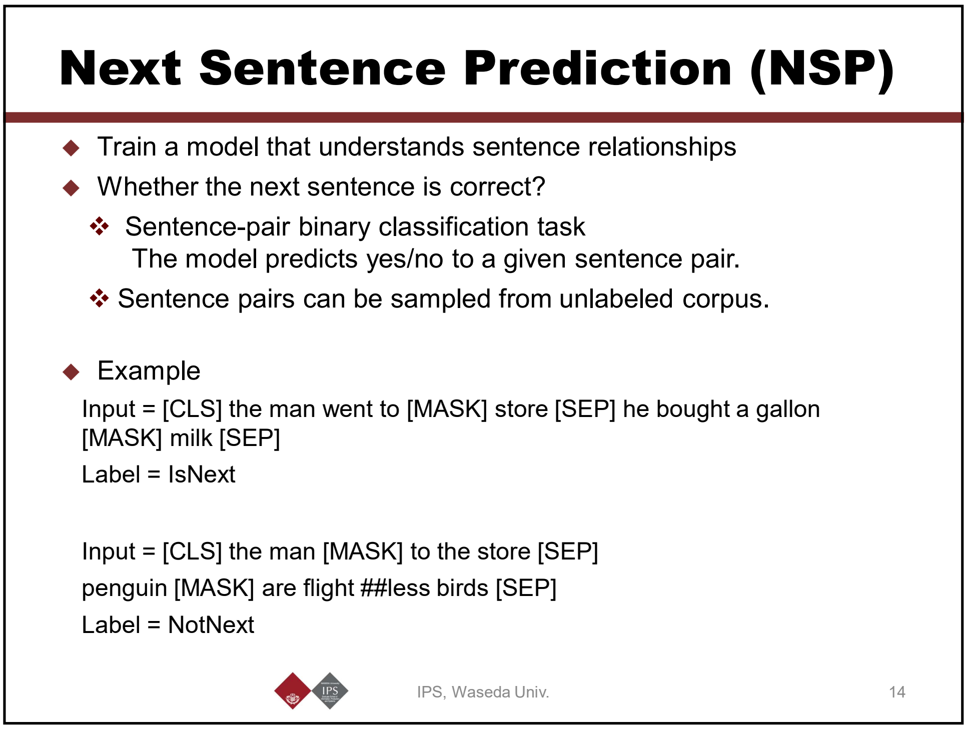
Softmax function

- Softmax function

$$\mathrm{softmax}_i(\boldsymbol{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

for $i = 1, \dots, K$ and $(\boldsymbol{z}) = (z_1, \dots., z_K) \in \mathbb{R}^K$.

- Normalizing input K real values of any range into a probability distribution, ranging 0 and 1 and summing up to 1.
- The softmax of (1, 2, 8) is approximately (0.001,0.002,0.997).
  - Emphasizing the maximal element 8, and assigning near zeros to all the other elements.
- Softmax is “smooth arg max.”
  - argmax (1, 2, 8) is (0, 0, 1).
  - argmax(0, 3, 3) is (0, 0.5, 0.5)
- Differentiable
  - Backward propagation is possible.
- Often used as the loss function for multi-class classification problems.
  - Choosing the class of the highest score.



Next Sentence Prediction (NSP)

- Train a model that understands sentence relationships
- Whether the next sentence is correct?
  - Sentence-pair binary classification task
    The model predicts yes/no to a given sentence pair.
  - Sentence pairs can be sampled from unlabeled corpus.

- Example

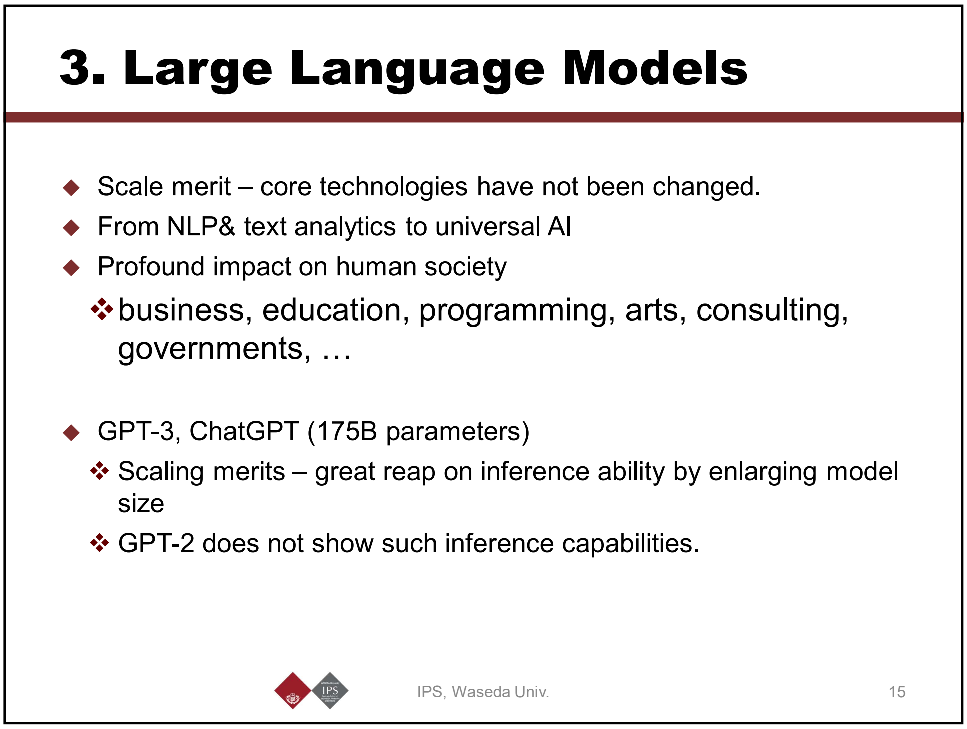
Input = [CLS] the man went to [MASK] store [SEP] he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

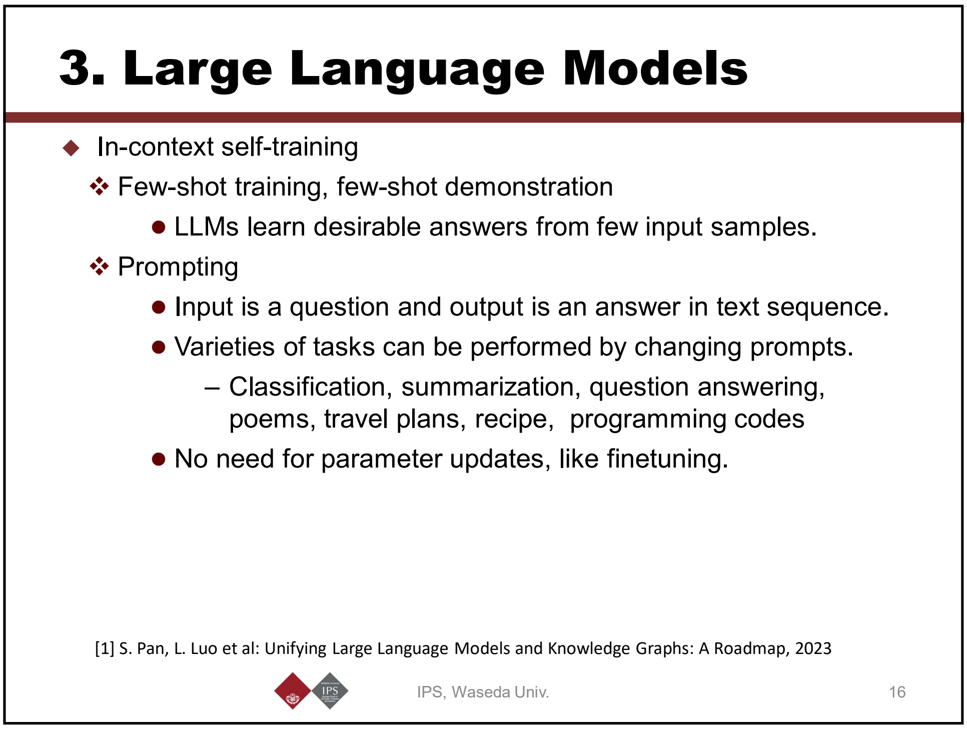
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext



3. Large Language Models

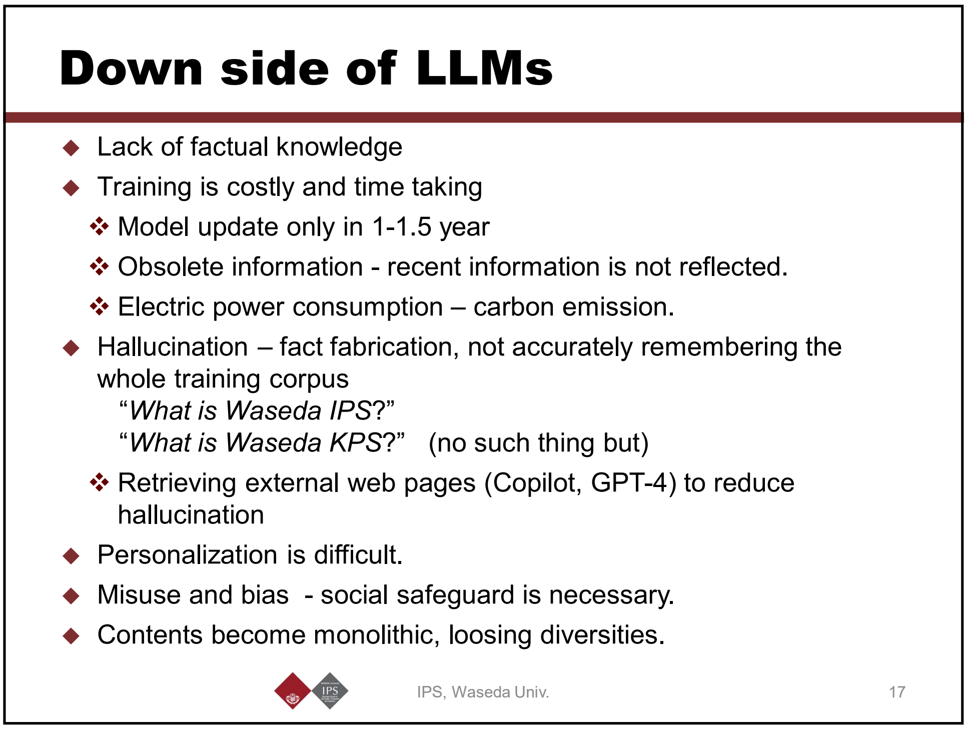
- Scale merit – core technologies have not been changed.
- From NLP& text analytics to universal AI
- Profound impact on human society
  - business, education, programming, arts, consulting, governments, …
- GPT-3, ChatGPT (175B parameters)
  - Scaling merits – great reap on inference ability by enlarging model size
  - GPT-2 does not show such inference capabilities.



3. Large Language Models

- In-context self-training
  - Few-shot training, few-shot demonstration
    - LLMs learn desirable answers from few input samples.
  - Prompting
    - Input is a question and output is an answer in text sequence.
    - Varieties of tasks can be performed by changing prompts.
      - Classification, summarization, question answering, poems, travel plans, recipe, programming codes
    - No need for parameter updates, like finetuning.

[1] S. Pan, L. Luo et al: Unifying Large Language Models and Knowledge Graphs: A Roadmap, 2023

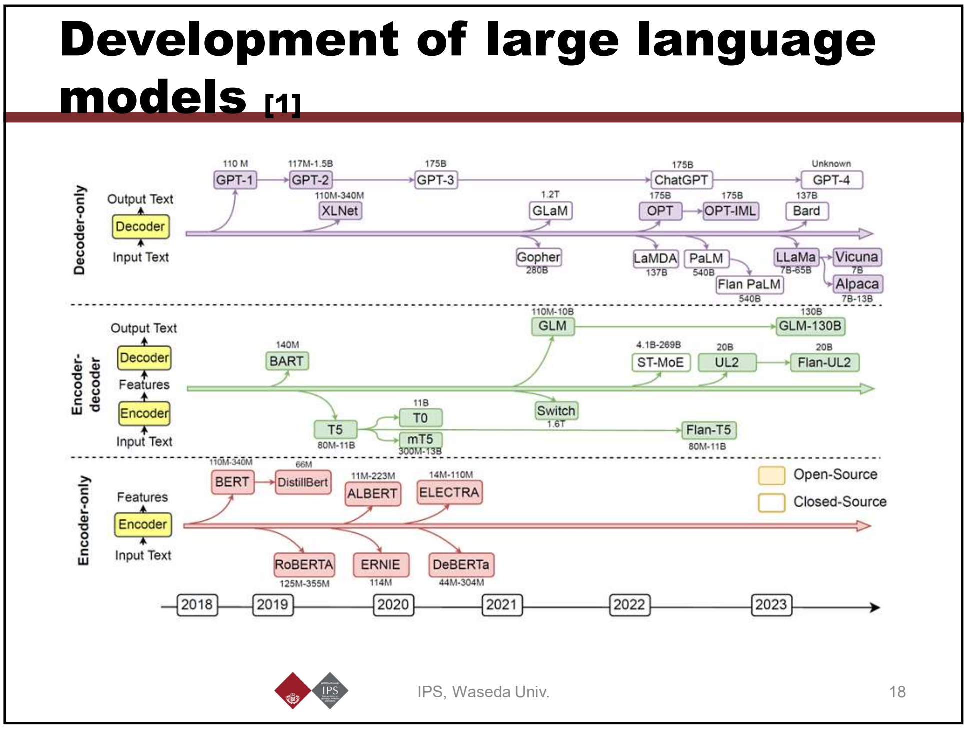


Down side of LLMs

- Lack of factual knowledge
- Training is costly and time taking
  - Model update only in 1-1.5 year
  - Obsolete information - recent information is not reflected.
  - Electric power consumption – carbon emission.
- Hallucination – fact fabrication, not accurately remembering the whole training corpus

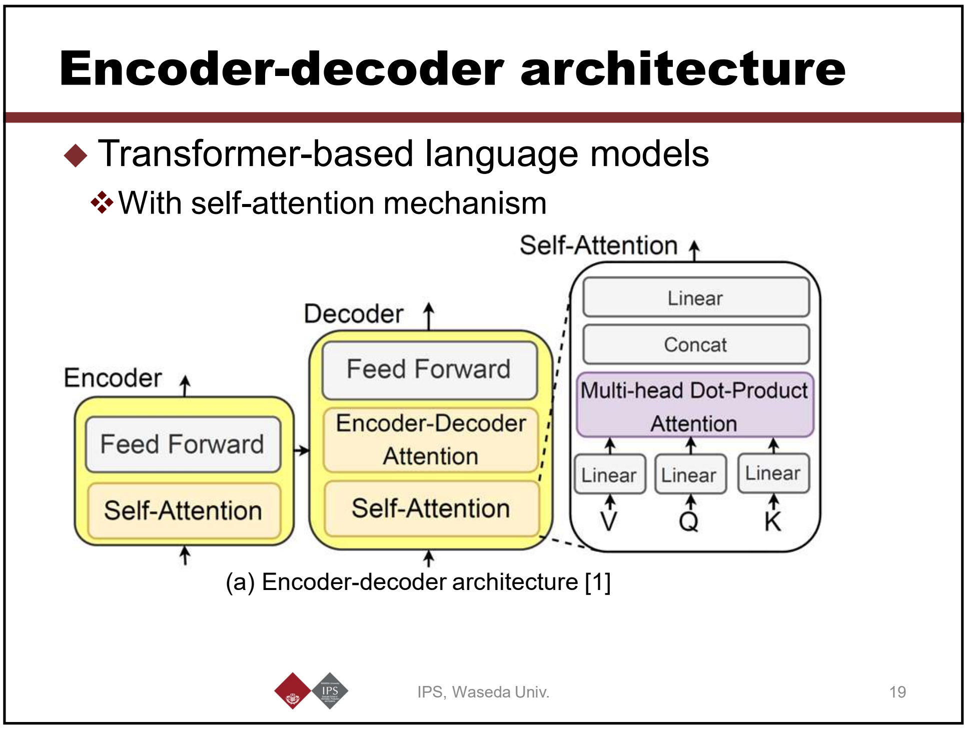
  "*What is Waseda IPS*?"
  "*What is Waseda KPS*?"   (no such thing but)
  - Retrieving external web pages (Copilot, GPT-4) to reduce hallucination
- Personalization is difficult.
- Misuse and bias - social safeguard is necessary.
- Contents become monolithic, loosing diversities.

IPS, Waseda Univ. 17



Development of large language models [1]

IPS, Waseda Univ. 18

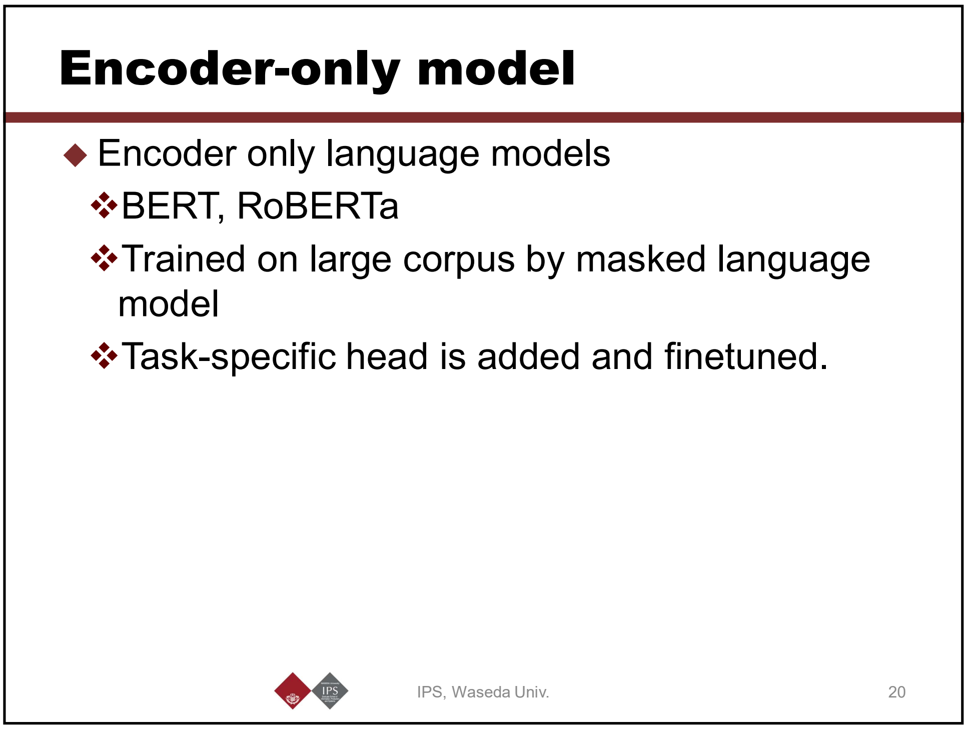


Encoder-decoder architecture

- Transformer-based language models
  - With self-attention mechanism

(a) Encoder-decoder architecture [1]

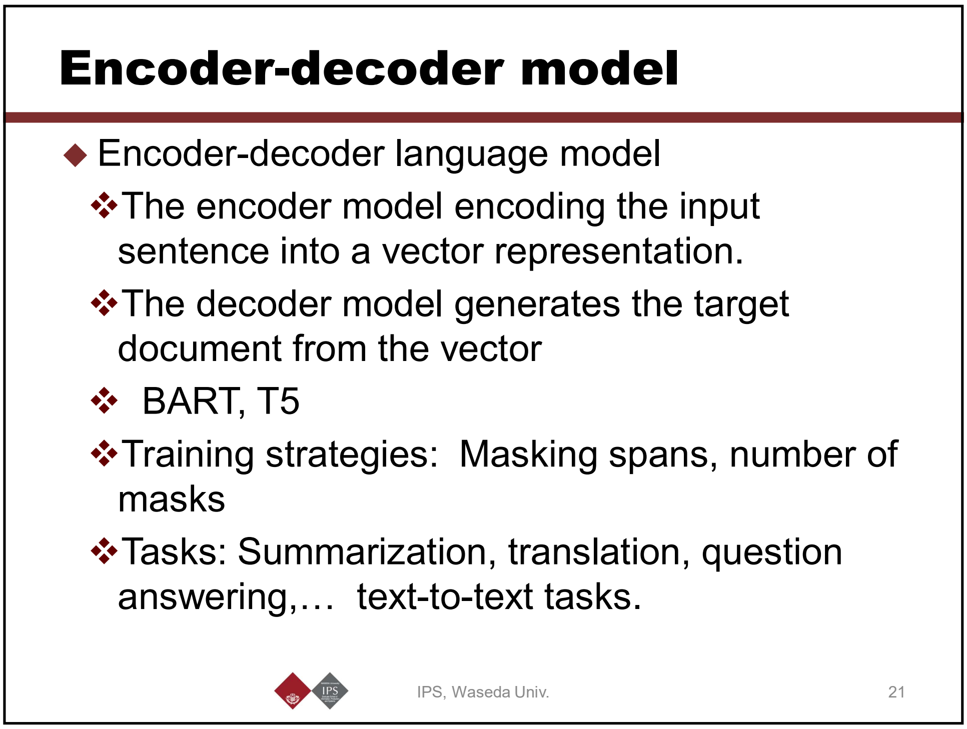
IPS, Waseda Univ. 19



Encoder-only model

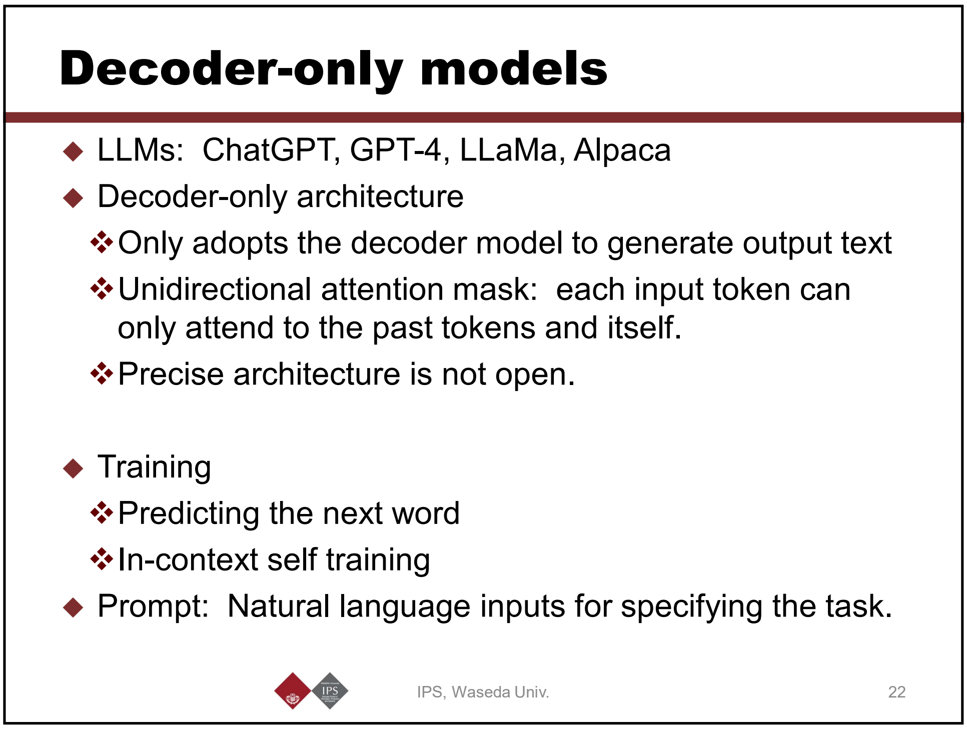
- Encoder only language models
  - BERT, RoBERTa
  - Trained on large corpus by masked language model
  - Task-specific head is added and finetuned.

IPS, Waseda Univ. 20



Encoder-decoder model

- Encoder-decoder language model
  - The encoder model encoding the input sentence into a vector representation.
  - The decoder model generates the target document from the vector
  - BART, T5
  - Training strategies: Masking spans, number of masks
  - Tasks: Summarization, translation, question answering,… text-to-text tasks.



Decoder-only models

- LLMs: ChatGPT, GPT-4, LLaMa, Alpaca
- Decoder-only architecture
  - Only adopts the decoder model to generate output text
  - Unidirectional attention mask: each input token can only attend to the past tokens and itself.
  - Precise architecture is not open.

- Training
  - Predicting the next word
  - In-context self training
- Prompt: Natural language inputs for specifying the task.

Prompt

- Instruction: Specifies a specific task.
- Context: Context of the task, or few-shot examples. Conversation with the user.
- Input text: On which the task is executed.

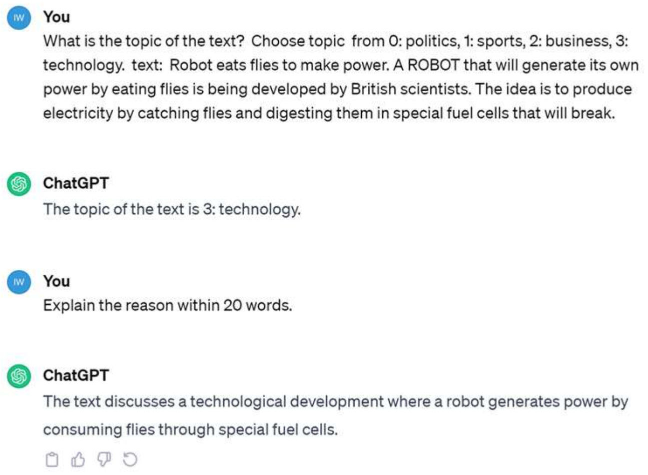
- Few-shot demonstration
- Prompts can utilize LLMs without finetuning.
  - Finetuning LLMs is impractical, due to the parameter size.
- Prompt engineering: Finding effective prompts, manually or automatically.

Prompt example on ChatGPT

Prompt

Answer

Explanation



You

What is the topic of the text? Choose topic from 0: politics, 1: sports, 2: business, 3: technology. text: Robot eats flies to make power. A ROBOT that will generate its own power by eating flies is being developed by British scientists. The idea is to produce electricity by catching flies and digesting them in special fuel cells that will break.

ChatGPT

The topic of the text is 3: technology.

You

Explain the reason within 20 words.

ChatGPT

The text discusses a technological development where a robot generates power by consuming flies through special fuel cells.

Parameter-efficient Finetuning of LLMs

- Full finetuning of LLMs, like GPT 175B, is infeasible, because all parameters need to be retrained.
- Parameter-efficient tuning
  - Freezing most of parameters and update only small portions of task specific parameters, like 1%-0.01%
  - Replacing task-specific parameters
    - For adopting tasks
    - For customization, personalization, recommendation, etc
- Freezing several layers.
- Prompt tuning
- LoRA: Low-Rank Adaptation

LoRA [3]

- $P_\Phi(y|x)$ is a pre-trained autoregressive language model parameterized by Φ.
- $Z = \{(x_i, y_i)\}_{i=1,\ldots,N}$ is a training dataset of a downstream task, where x_i, y_i are token sequences.
  - For summarization, x_i is a document, and y_i is a summary.
- Full finetuning: pretrained weights Φ_0 are updated to $\Phi_0 + \Delta\Phi$ by the gradient to maximize the conditional language modeling objective:

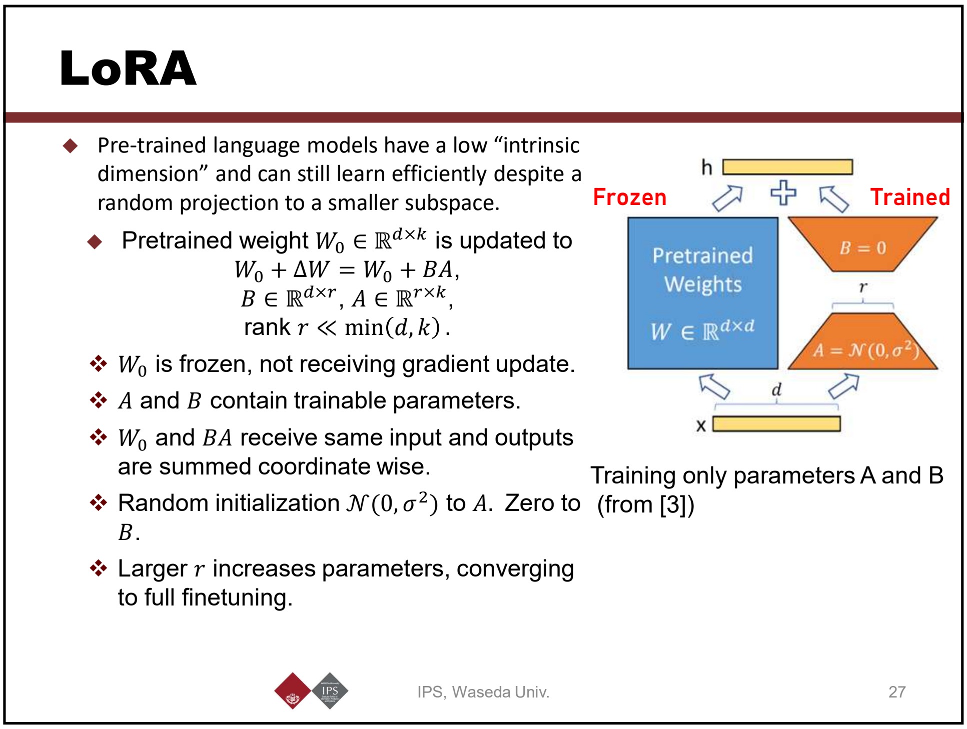
$$\max_{\Theta} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log(p_{\Phi_0+\Delta\Phi(\Theta)}(y_t|x, y_{<t})),$$

$y_{<t}$: tokens before y_t

The model size $|\Phi_0| \approx 135B$ for GPT-3. Model update $\Delta\Phi$ has the same dimension with the model Φ_0 for full finetuning.

LoRA: Encoding $\Delta\Phi(\Theta)$ with much smaller-sized parameters Θ, $|\Theta| \ll |\Phi_0|$, by low rank representation of $\Delta\Phi(\Theta)$.

[3] Edward J Hu, Yelong Shen, et al., LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

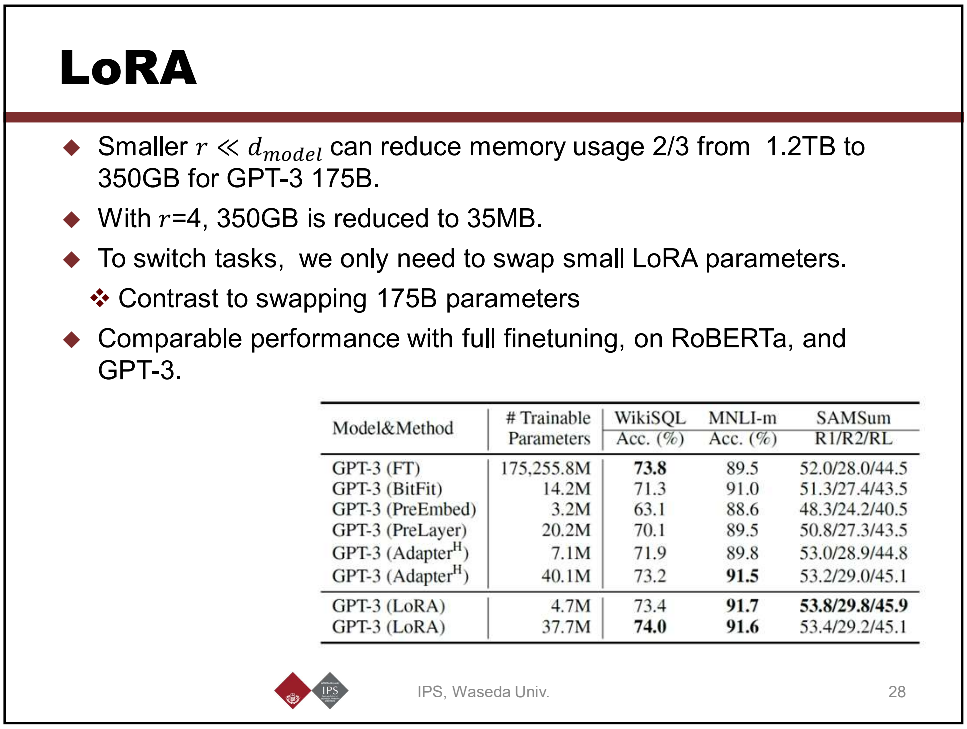


LoRA

- Pre-trained language models have a low “intrinsic dimension” and can still learn efficiently despite a random projection to a smaller subspace.
  - Pretrained weight $W_0 \in \mathbb{R}^{d\times k}$ is updated to
  $$W_0 + \Delta W = W_0 + BA,$$
  $$B \in \mathbb{R}^{d\times r},\ A \in \mathbb{R}^{r\times k},$$
  rank $r \ll \min(d,k)$.
  - W_0 is frozen, not receiving gradient update.
  - A and B contain trainable parameters.
  - W_0 and BA receive same input and outputs are summed coordinate wise.
  - Random initialization $\mathcal{N}(0,\sigma^2)$ to A. Zero to B.
  - Larger r increases parameters, converging to full finetuning.

Training only parameters A and B (from [3])

IPS, Waseda Univ. 27

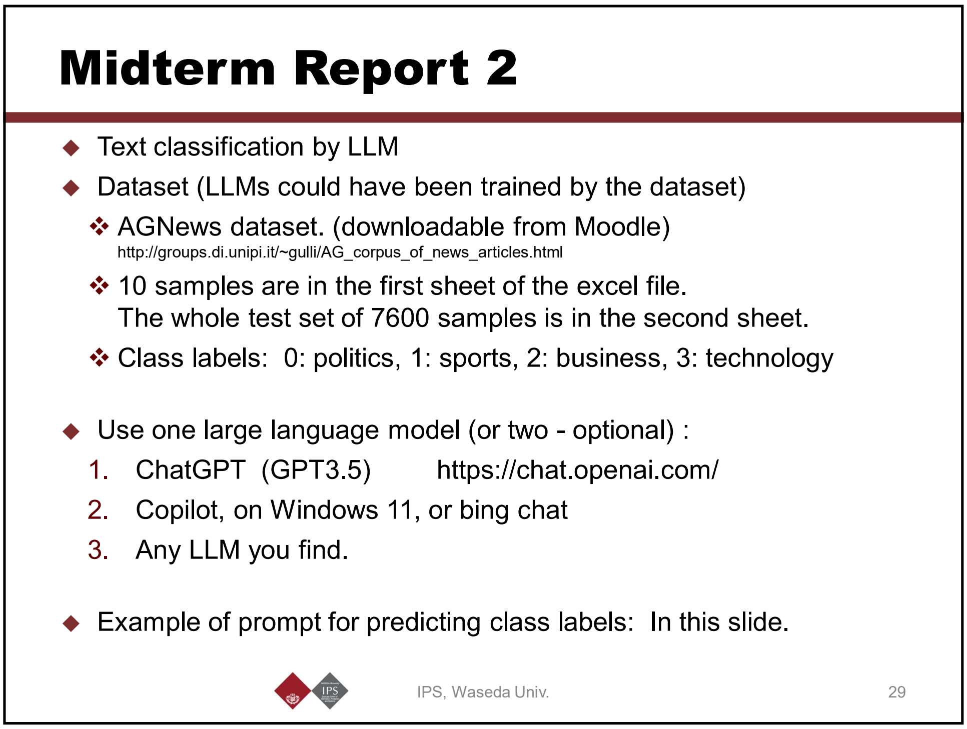


LoRA

- Smaller $r \ll d_{model}$ can reduce memory usage 2/3 from 1.2TB to 350GB for GPT-3 175B.
- With r=4, 350GB is reduced to 35MB.
- To switch tasks, we only need to swap small LoRA parameters.
  - Contrast to swapping 175B parameters
- Comparable performance with full finetuning, on RoBERTa, and GPT-3.

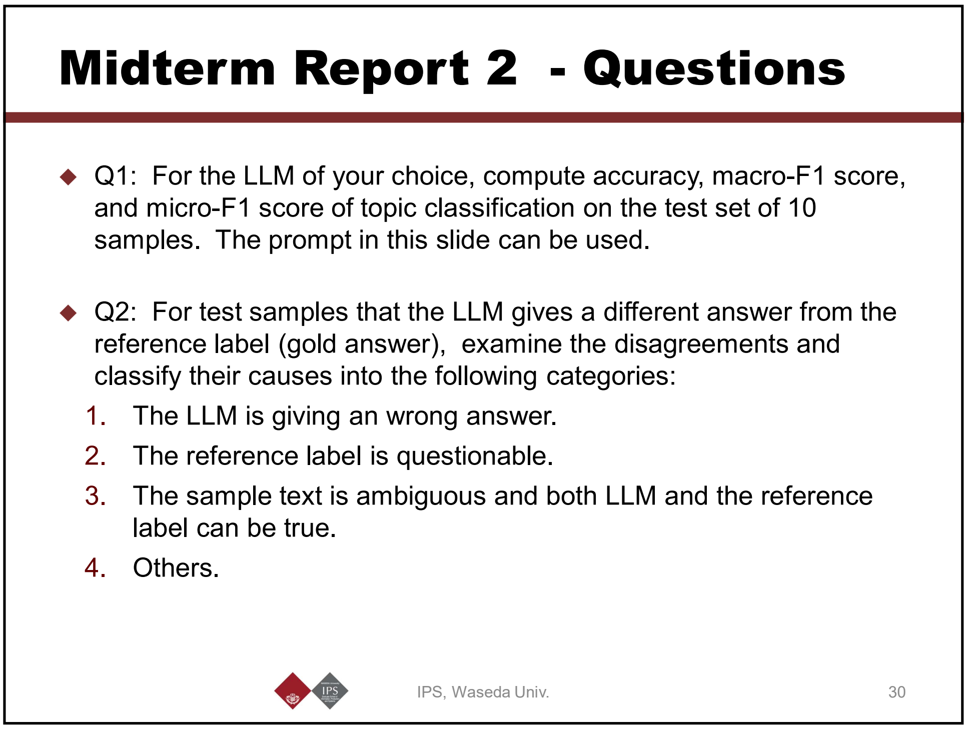
Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter^H)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter^H)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

IPS, Waseda Univ. 28



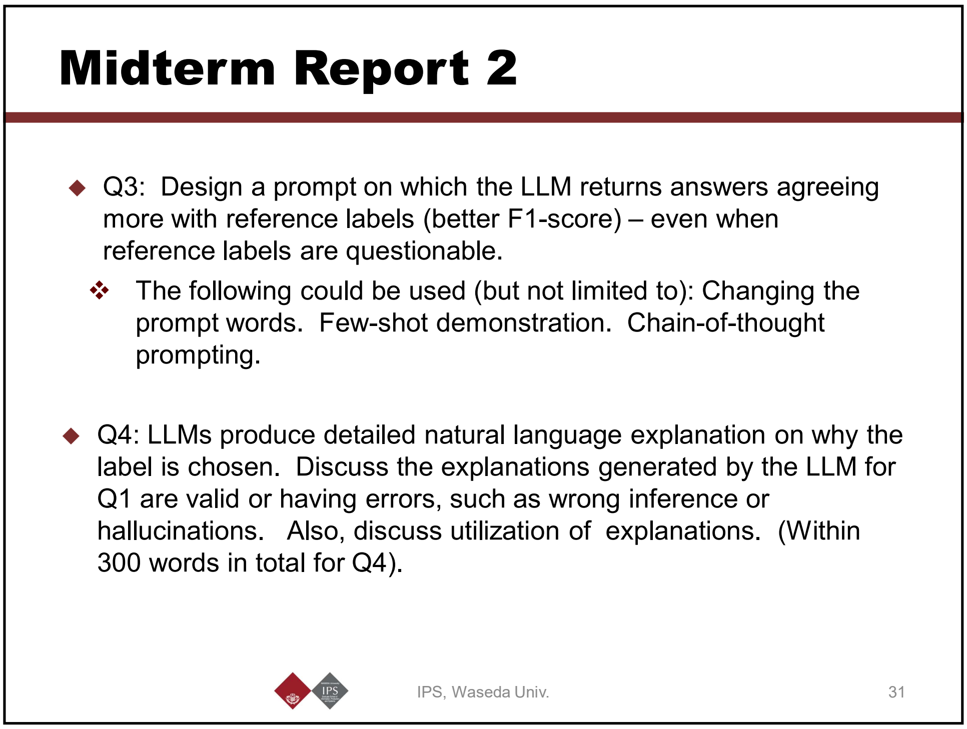
Midterm Report 2

- Text classification by LLM
- Dataset (LLMs could have been trained by the dataset)
  - AGNews dataset. (downloadable from Moodle)
    http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
  - 10 samples are in the first sheet of the excel file.
    The whole test set of 7600 samples is in the second sheet.
  - Class labels: 0: politics, 1: sports, 2: business, 3: technology
- Use one large language model (or two - optional) :
  1. ChatGPT (GPT3.5) https://chat.openai.com/
  2. Copilot, on Windows 11, or bing chat
  3. Any LLM you find.
- Example of prompt for predicting class labels: In this slide.



Midterm Report 2 - Questions

- Q1: For the LLM of your choice, compute accuracy, macro-F1 score, and micro-F1 score of topic classification on the test set of 10 samples. The prompt in this slide can be used.
- Q2: For test samples that the LLM gives a different answer from the reference label (gold answer), examine the disagreements and classify their causes into the following categories:
  1. The LLM is giving an wrong answer.
  2. The reference label is questionable.
  3. The sample text is ambiguous and both LLM and the reference label can be true.
  4. Others.



Midterm Report 2

- Q3: Design a prompt on which the LLM returns answers agreeing more with reference labels (better F1-score) – even when reference labels are questionable.
  - The following could be used (but not limited to): Changing the prompt words. Few-shot demonstration. Chain-of-thought prompting.

- Q4: LLMs produce detailed natural language explanation on why the label is chosen. Discuss the explanations generated by the LLM for Q1 are valid or having errors, such as wrong inference or hallucinations. Also, discuss utilization of explanations. (Within 300 words in total for Q4).