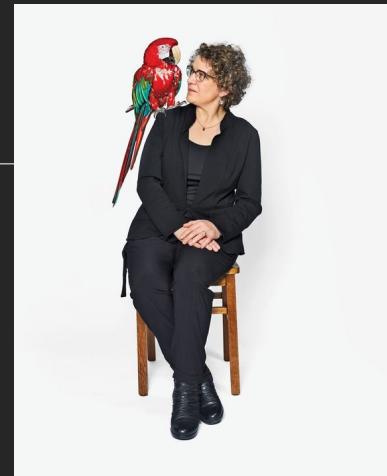


We've learned to make machines that can mindlessly generate text. But we haven't learned how to stop imagining the mind behind it.

- Emily Bender

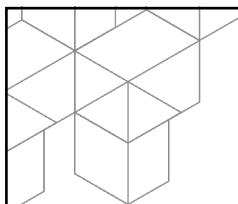


<https://nymag.com/intelligencer/article/ai-artificial-intelligence-chatbots-emily-m-bender.html>

All images from Wikimedia Commons unless specified.



1



Natural Language Processing (NLP)

COMP6713 – 2025 Term 1



Convener

Dr. Aditya Joshi

aditya.joshi@unsw.edu.au



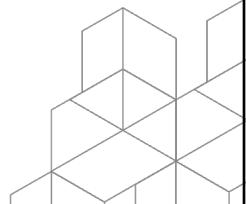
Week 4

Language Models



Schedule

2025 Term 1



2

1

UNSW SYDNEY | Australia's Global University

Week 4
(Transformer-based) Language Models

BERT Architecture Fine-tuning BERT Variants of BERT	Fine-tuning methods PEFT LoRA Prompt tuning
GPT Architecture In-context learning	Prompt Engineering
	Datasets and Libraries GLUE, BigBench, etc. LangChain
	Also... "Reading an NLP paper"

3

Announcement

Project Team Registration Form has been sent (check your email)

Registration Deadline: 19th March 2025, 7:00pm Sydney time.

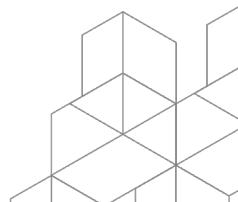
Final Project Submission Deadline: 24th April, 2025, 7:00pm Sydney time

Self-chosen project topics	Industry project
You choose the topic.	You follow the topic.
No cash prize involved.	You may win one among two cash prizes decided by SEEK. (No involvement from us at UNSW in this decision.)
Define scope based on the project scope document that will be provided by the course team	
You will be marked as per marking rubric by the course team at UNSW.	

Weekly quizzes (Moodle)

No quiz in weeks 6 and 10.

Guest speakers (30 mins) in weeks 5 and 10. (and likely, 1 more week.) Diverse.



4

Topics this week

<https://web.stanford.edu/~jurafsky/slp3/11.pdf>

Data → Features to look for → Ways to combine the features → Learn a model based on these combinations

Generation 1: Human Engineering (Icon: person with gear)

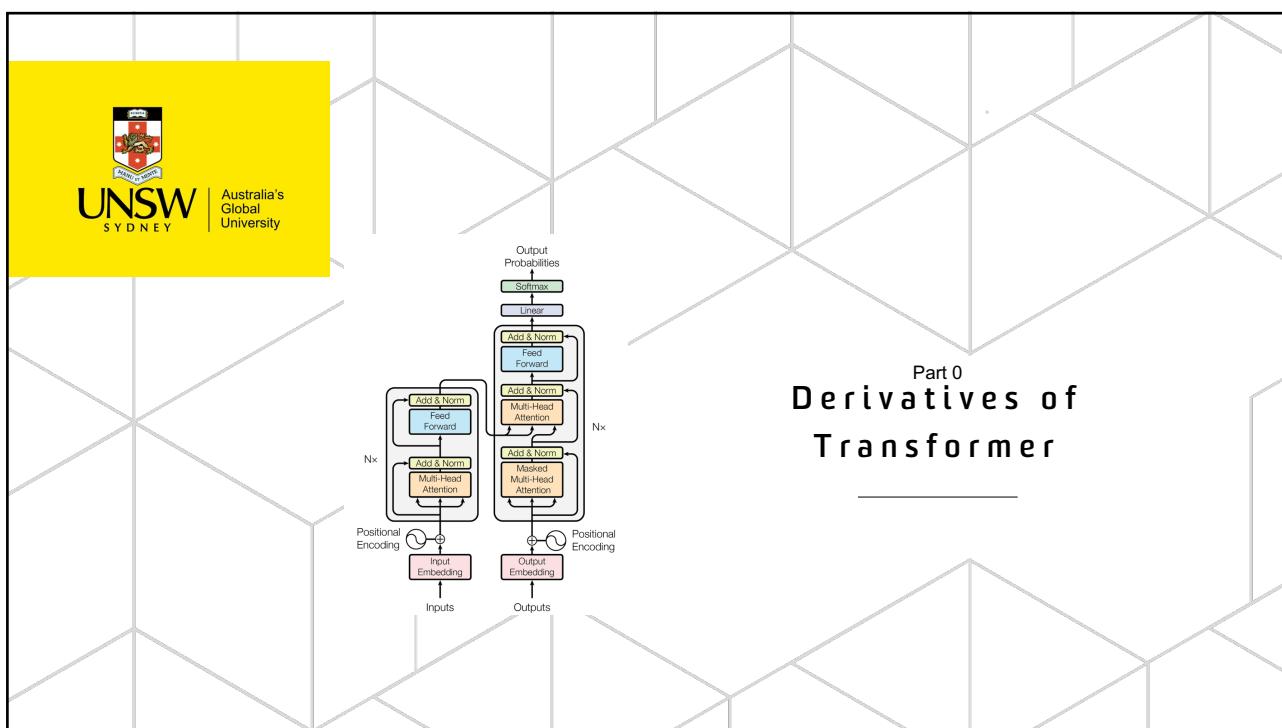
Generation 2: Computational model (Icon: computer with gear)

Generation 3: Neural model (Icon: computer with neural network nodes)

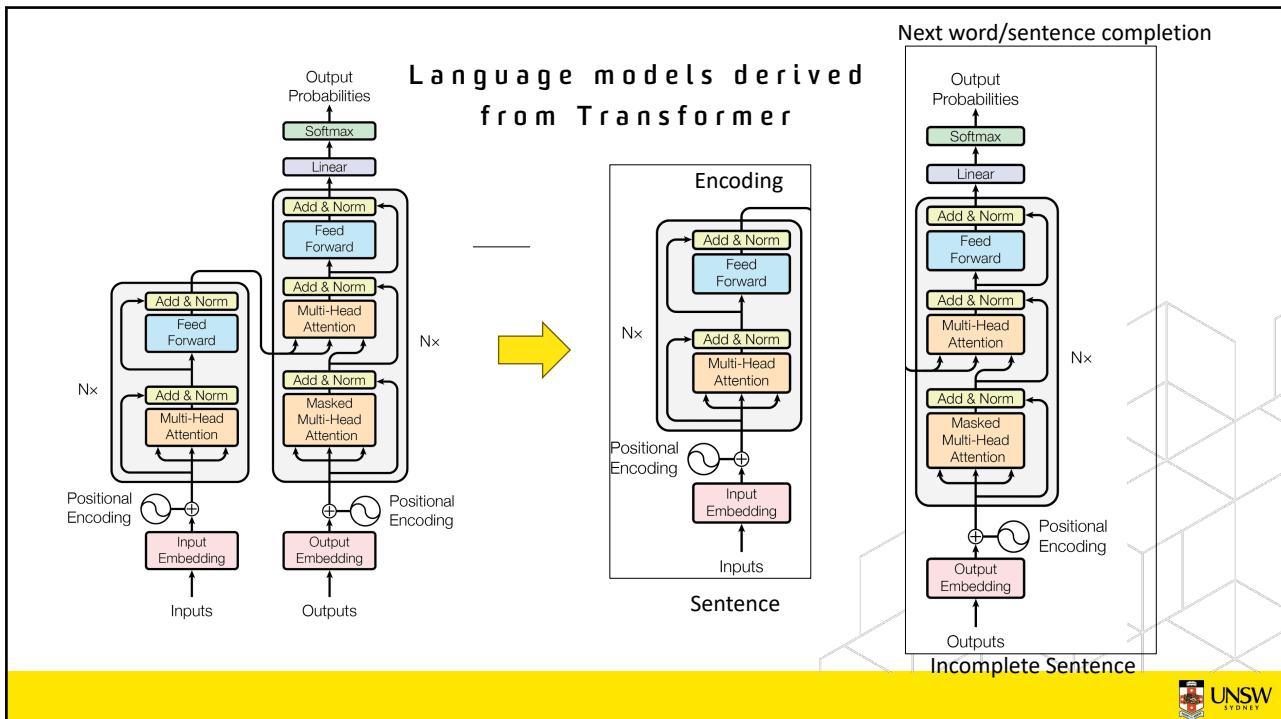
Legend: Human Engineering (Icon: person with gear), Computational model (Icon: computer with gear), Neural model (Icon: computer with neural network nodes)

BERT
GPT
Fine-tuning methods
LangChain

5



6



7



Broad categories

Language Models

Encoder models

Use the encoder of the Transformer

Auto-encoding models (i.e., Current word is estimated from neighbouring words.)

Analogy: Denoising in speech processing

Decoder models

Use the decoder of the Transformer

Auto-regressive models (i.e., Current word is estimated from previous words.)

Analogy: Time series forecasting

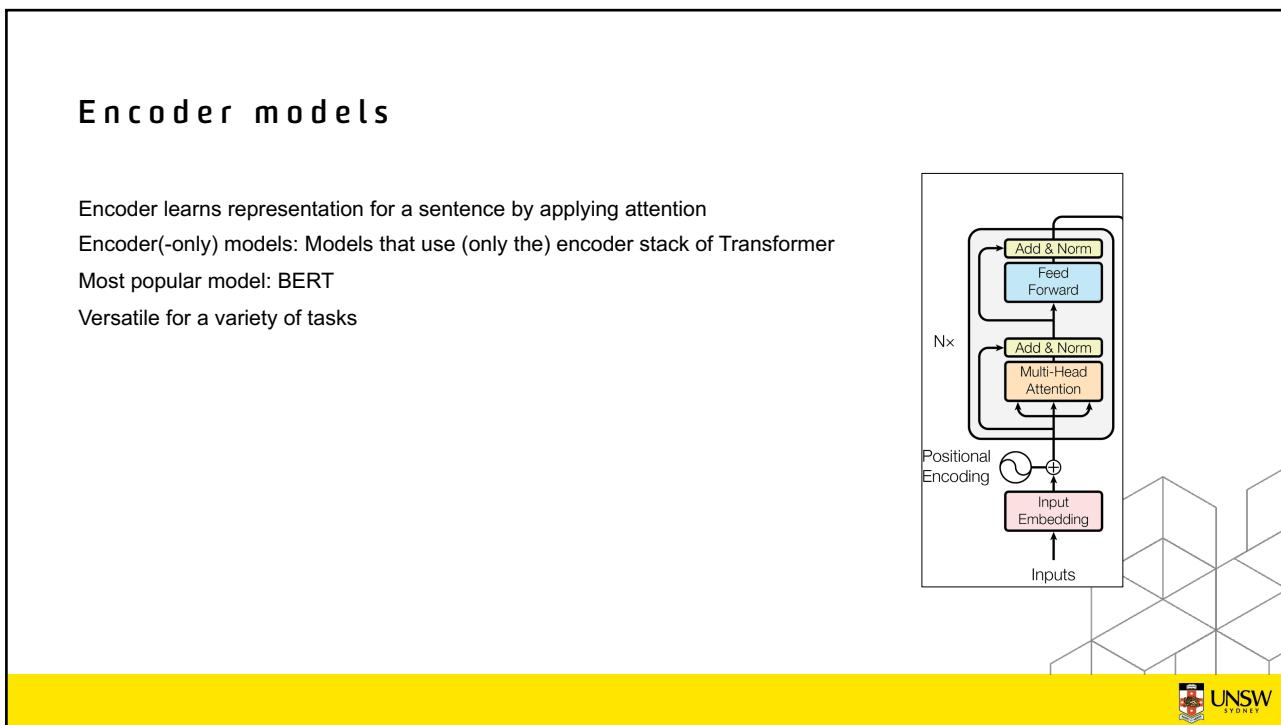
*Encoder-decoder models such as XLNet and BART have also shown to be effective. (Week 9)



8



9



10

B E R T

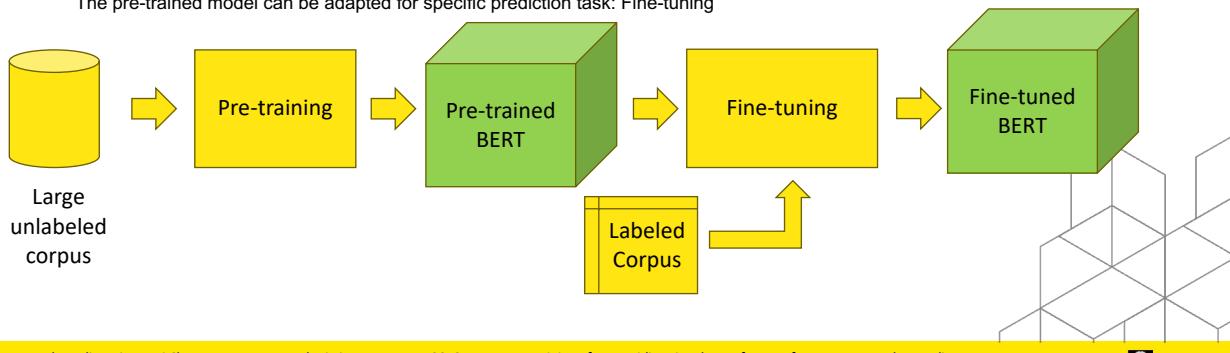
Bidirectional Encoder Representations from Transformers (BERT)

Auto-encoding model (current state based on previous and next state)

Two steps:

Encoder used to learn representations of text: Pre-training

The pre-trained model can be adapted for specific prediction task: Fine-tuning



Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In proceedings of NAACL.



11

BERT pre-training

What does it mean to know a language? BERT says, you need to be able to do two things:

"fill missing words" "understand if a sentence follows the other"

Human analogy: “Cloze task”; Understanding people with an unfamiliar accent.

Human analogy: If a text is scrambled, you will be able to arrange the sentences in the right order.

Self-supervised task:

Self-supervised task:

I went to the bank to withdraw cash.

He entered the room. He saw his ex sitting

Input: I went to the _____ to withdraw cash.

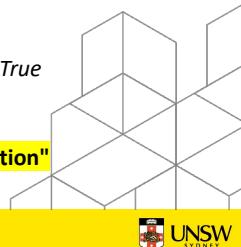
Output: bank

Input: I went to the _____ to withdraw cash.

Input: He entered the room. & He saw his ex sitting at the table.

Output: True

“Next sentence prediction”





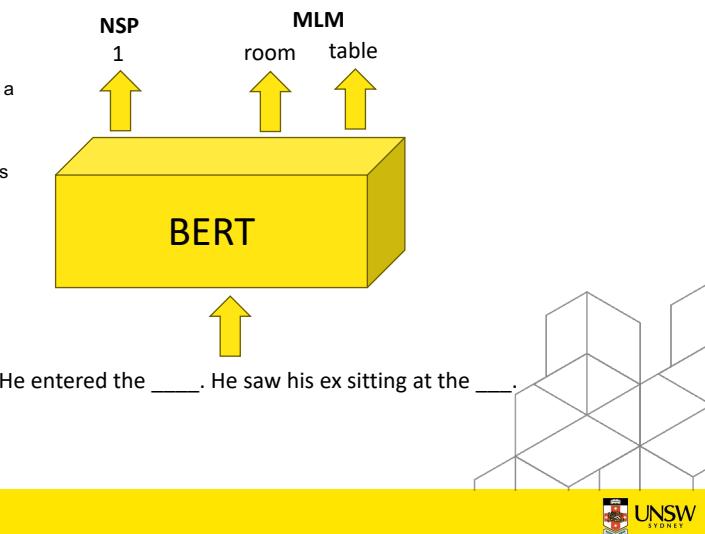
UNSW
SYDNEY

12

Pre-training objectives

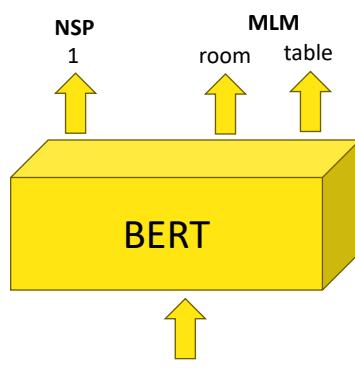
BERT uses two pre-training objectives:

1. Masked language modeling (MLM)
 - Word prediction: Learn to predict missing words in a sentence
2. Next sentence prediction (NSP)
 - Classification: Learn to predict if a sentence follows another



13

Special tokens

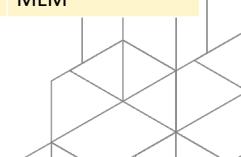


BERT uses three special tokens:

Special Token	Position	Purpose
[CLS]	Beginning of a pair of sentences.	Representation used to learn NSP
[SEP]	Between the pair.	Indicate new sentence.
[MASK]	Masked words	Representation used to learn MLM

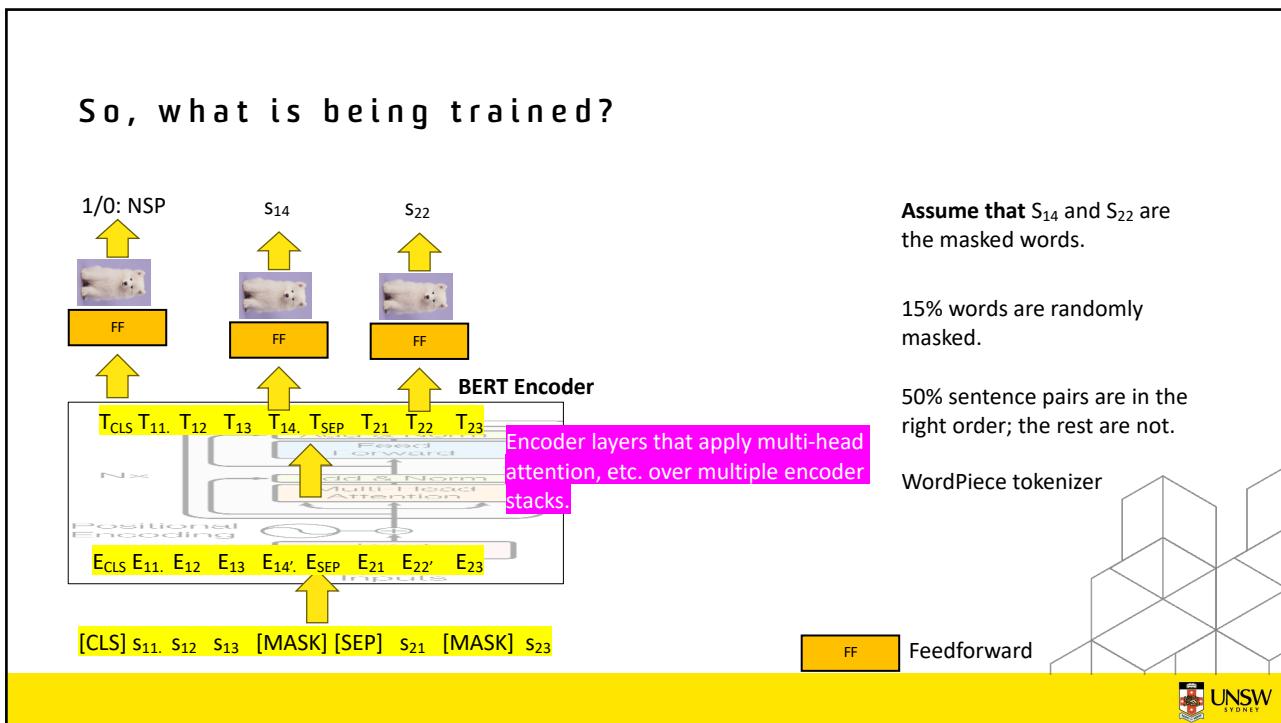
[CLS] He entered the [MASK] . [SEP] He saw his ex sitting at the [MASK] .

He entered the _____. He saw his ex sitting at the _____.
_____. _____.



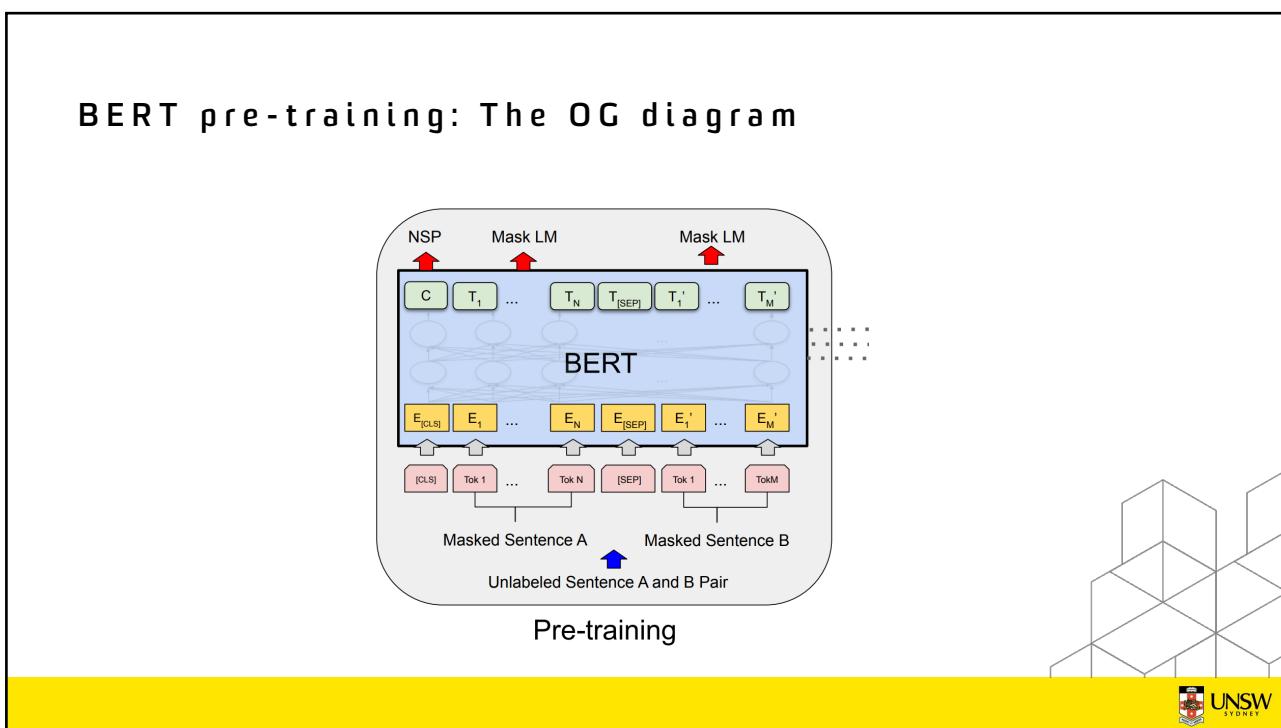
14

So, what is being trained?

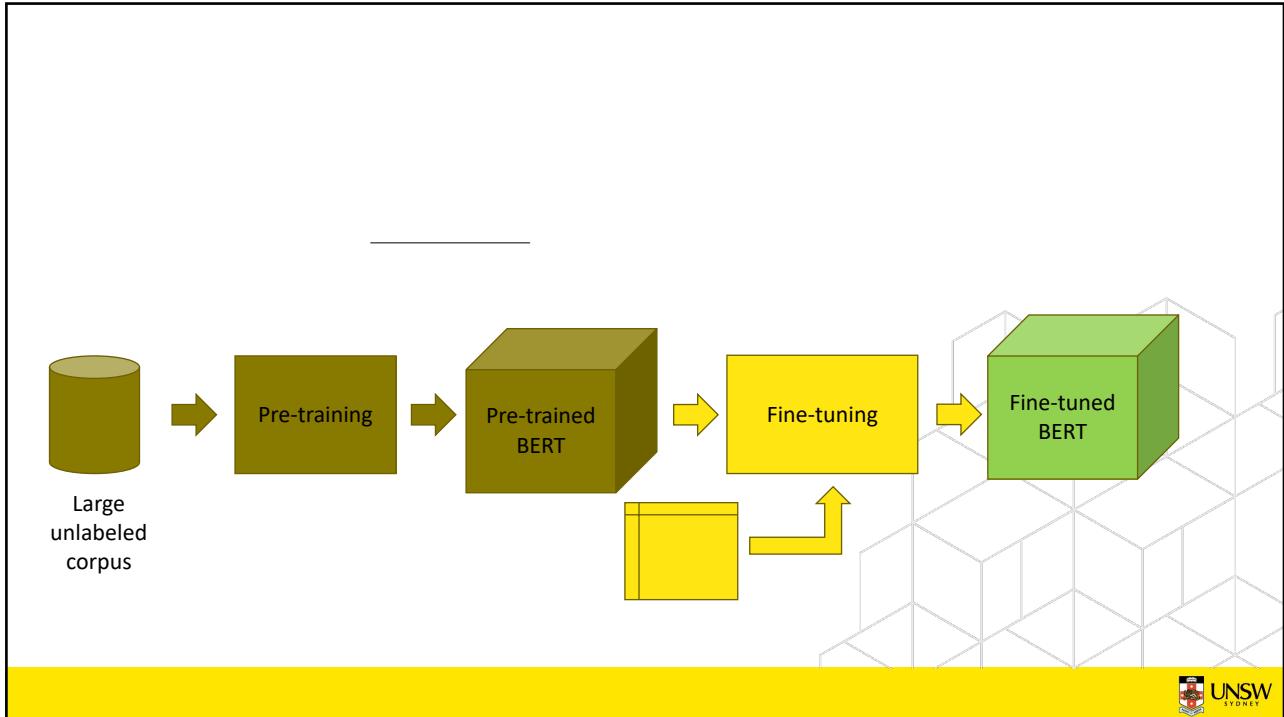


15

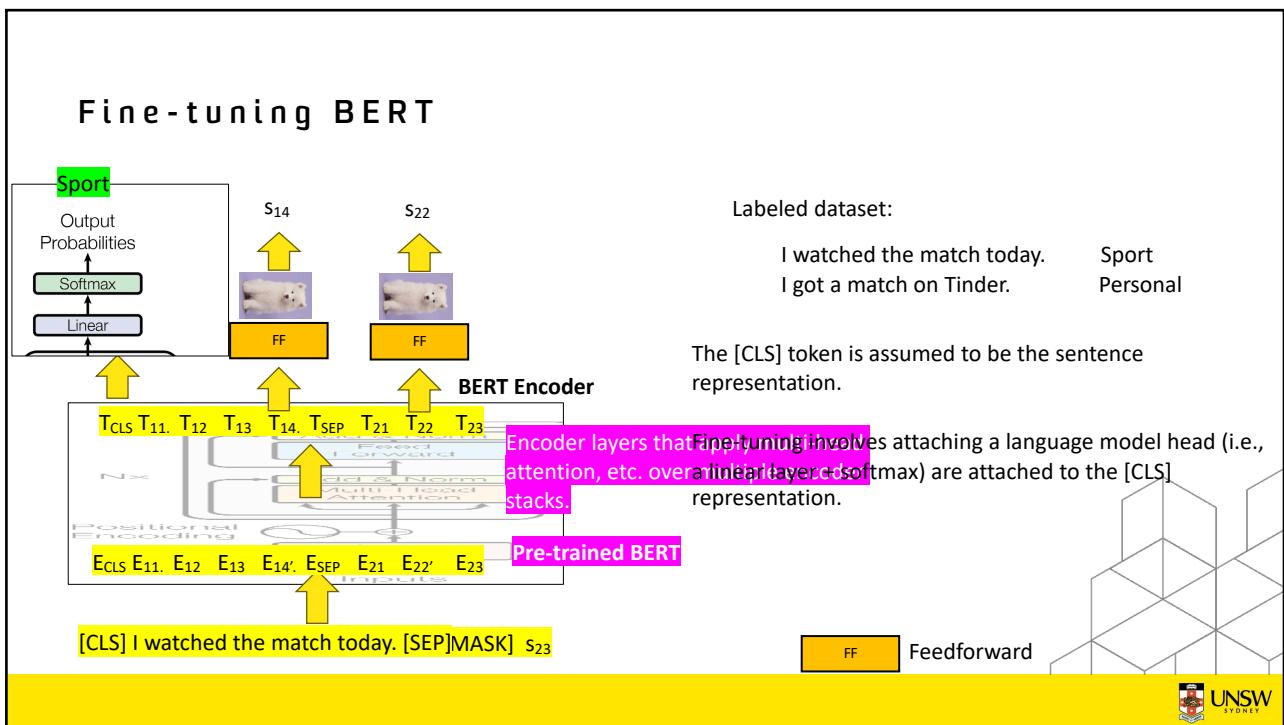
BERT pre-training: The OG diagram



16



17



18

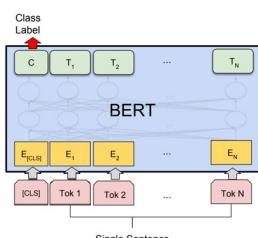
Fine-tuning BERT

Adapt a pre-trained BERT model for a specific task

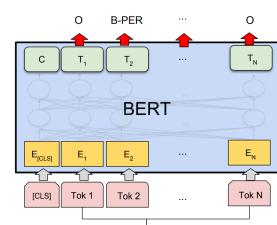
Use language model head on the appropriate representations.

All parameters may be updated.

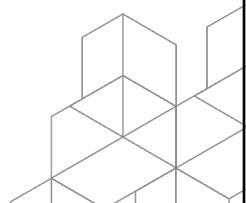
Task: Sentiment classification, Named Entity Recognition, etc.



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



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER



19

BERT-derived models



Look them up on HuggingFace!



RoBERTa

ERNIE

XLNet

"Robust" BERT
Dynamic masking, etc.

"Knowledge"-aware BERT
Entity masking, etc.

Permutation language modeling (not purely auto-encoding)



SciBERT

BioBERT

BERT pre-trained on scientific articles

BERT pre-trained on biomedical text



DistilBERT

ALBERT

"Distilled" BERT.
Resource-efficient.

"Lite" BERT.
Shared parameters across layers.



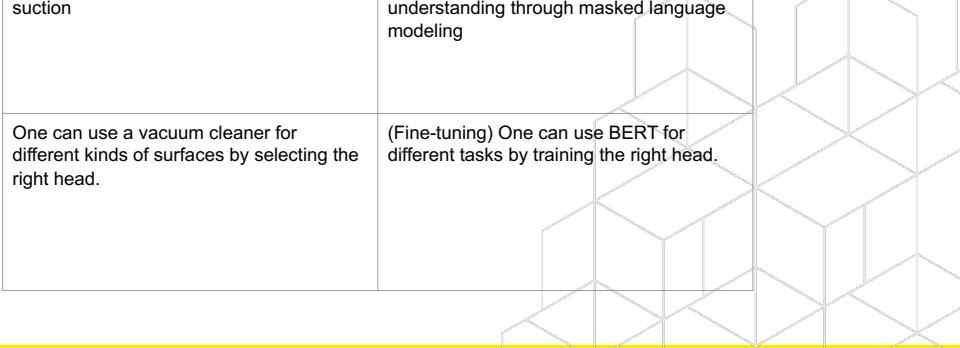
Demo time!



20

**Pre-trained language models
like BERT are a vacuum
cleaner**

Vacuum Cleaner	Language Models
	(Pre-training) Simulates language understanding through masked language modeling
	(Fine-tuning) One can use BERT for different tasks by training the right head.



UNSW SYDNEY

21



Part 2

Decoder models

UNSW SYDNEY | Australia's Global University

22

Decoder Models

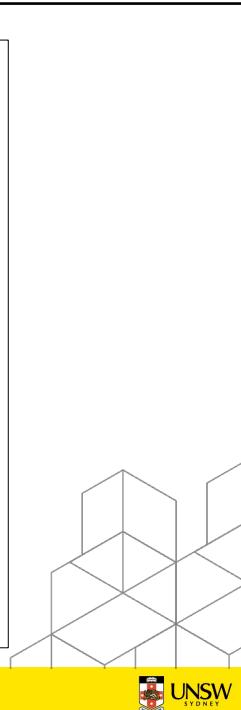
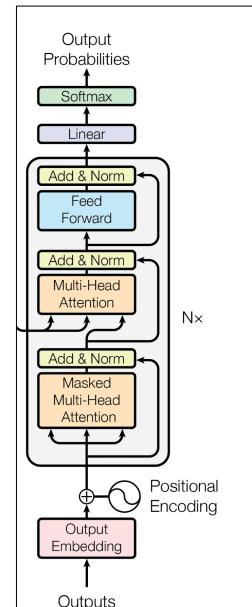
Decoder learns to complete sentences

Decoder(-only) models: Models that use (only the) decoder stack of Transformer

A popular model: GPT-3

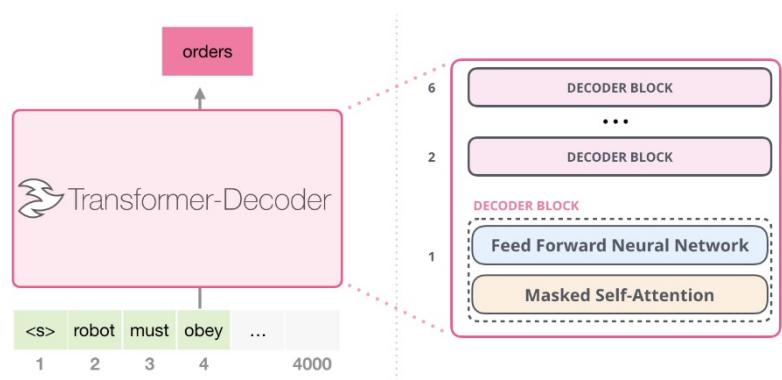
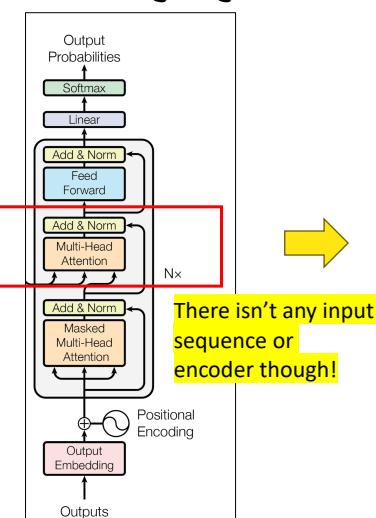
Human analogy: Best friends complete each others' sentences.

Versatile for a variety of tasks: PROMPTING



23

Modifying the decoder



24

Generative Pre-trained Transformer (GPT)-3

Auto-regressive model

175 billion parameters

You may not need to fine-tune: just use the pre-trained model to complete an input (known as the 'prompt')

Prompt:

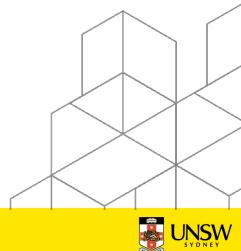
"What is the sentiment of the following sentence?

Sentence: I love the movie.

Answer: "

Expected output: "Positive"

No weights are updated.



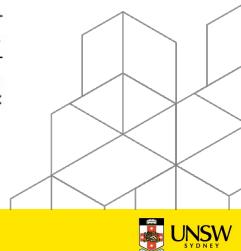
Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

25

Results from GPT-3 on translation

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	45.6^a	35.0 ^b	41.2^c	40.2 ^d	38.5^e	39.9^e
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ ⁺¹⁹]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG ⁺²⁰]	-	-	<u>29.8</u>	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

Table 3.4: Few-shot GPT-3 outperforms previous unsupervised NMT work by 5 BLEU when translating into English reflecting its strength as an English LM. We report BLEU scores on the WMT'14 Fr→En, WMT'16 De→En, and WMT'16 Ro→En datasets as measured by multi-bleu.perl with XLM's tokenization in order to compare most closely with prior unsupervised NMT work. SacreBLEU^f [Pos18] results reported in the appendix. Underline indicates an unsupervised or few-shot SOTA, bold indicates supervised SOTA with relative confidence. ^a[EOAG18] ^b[DHKH14] ^c[WXH⁺¹⁸] ^d[oR16] ^e[LGG⁺²⁰] ^f[SacreBLEU signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.intl+version.1.2.20]



26

Models similar to GPT

Open Pre-trained Transformer (OPT)

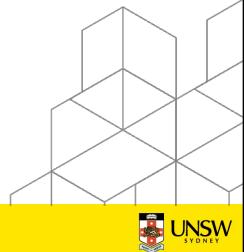
Google's T5

LLaMA

Alpaca

..and many more

Look up the pre-trained versions on HuggingFace. Use them in your projects!



27

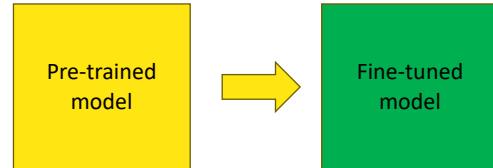


28

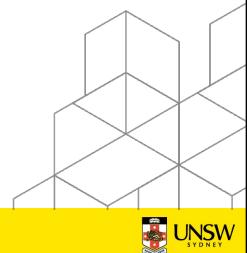
Fine-tuning methods

Fine-tuning requires updating model parameters, based on labeled data

You may freeze 'some' layers.
(`requires_grad`)



Note: We will cover how they are used for specific NLP tasks in the next few weeks.



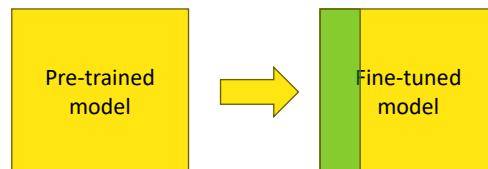
29

Prefix-tuning

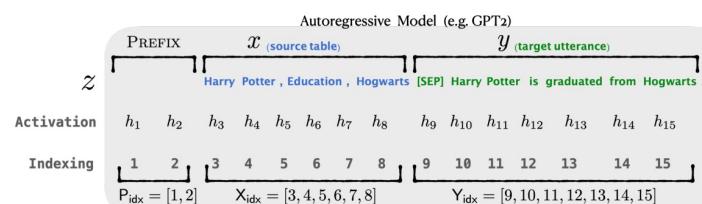
Special prefix tokens are added to the beginning of the sequence

Vectors corresponding to prefix tokens are learned too.

Majority of model parameters are kept frozen



$$h_i = \begin{cases} P_\theta[i, :], & \text{if } i \in P_{\text{idx}}, \\ \text{LM}_\phi(z_i, h_{<i}), & \text{otherwise.} \end{cases}$$

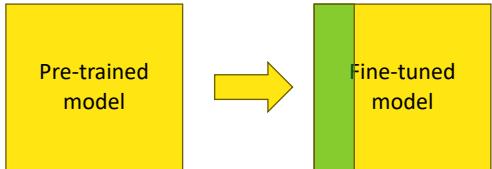


Li, X.L. and Liang, P., 2021, August. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of ACL.



30

Prompt tuning



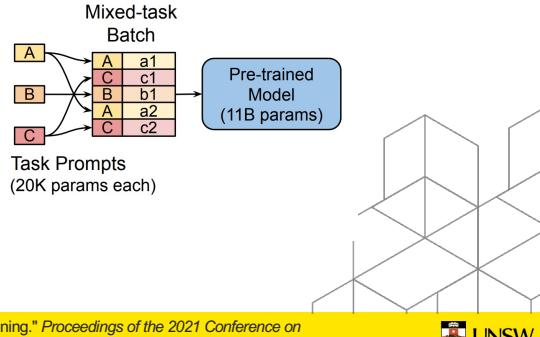
Fine-tune a decoder model for the text generation task

Generalisation of prefix-tuning

Prefixes in this case are meaningful prompts.

The resultant prompts are called "soft prompts".

Prompt Tuning



Mixed-task Batch

A	A	a1
C	C	c1
B	B	b1

A	A	a2
C	C	c2

Pre-trained Model (11B params)

Task Prompts (20K params each)

Lester, Brian, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning." *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021.

UNSW SYDNEY

31

Parameter-efficient fine-tuning (PEFT):

With larger models, full fine-tuning means a significant computational overhead.

Do not update all parameters (i.e., keep them 'frozen'). Update a smaller set of appropriately selected parameters. <https://github.com/huggingface/peft>

Several PEFT approaches have been proposed.

UNSW SYDNEY

32

Low Rank Adaptation (LoRA)

Keep the pre-trained weights frozen.
Only learn two rank decomposition matrices A and B
Because they are low-rank, they correspond to weight changes and are much faster to update.

Diagram shows initialization.

Reduces the number of parameters for GPT-3 to train by 10,000 times.

Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *ICLR*. 2021.

UNSW
SYDNEY

33

LoRA (Continued)

Read the LoRA (and qLoRA) paper if you love algorithms and data structures.

Let's read the LoRA paper!

Demo time!

UNSW
SYDNEY

34

**Let's fine-tune OPT for
sentence completion using
LoRA**

https://colab.research.google.com/drive/1pudaTi_2M6YNPVy6V6ohExbUucN8eJd3#scrollTo=orf1rmfUjwyk



Demo time!

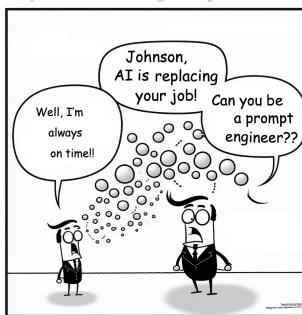




35

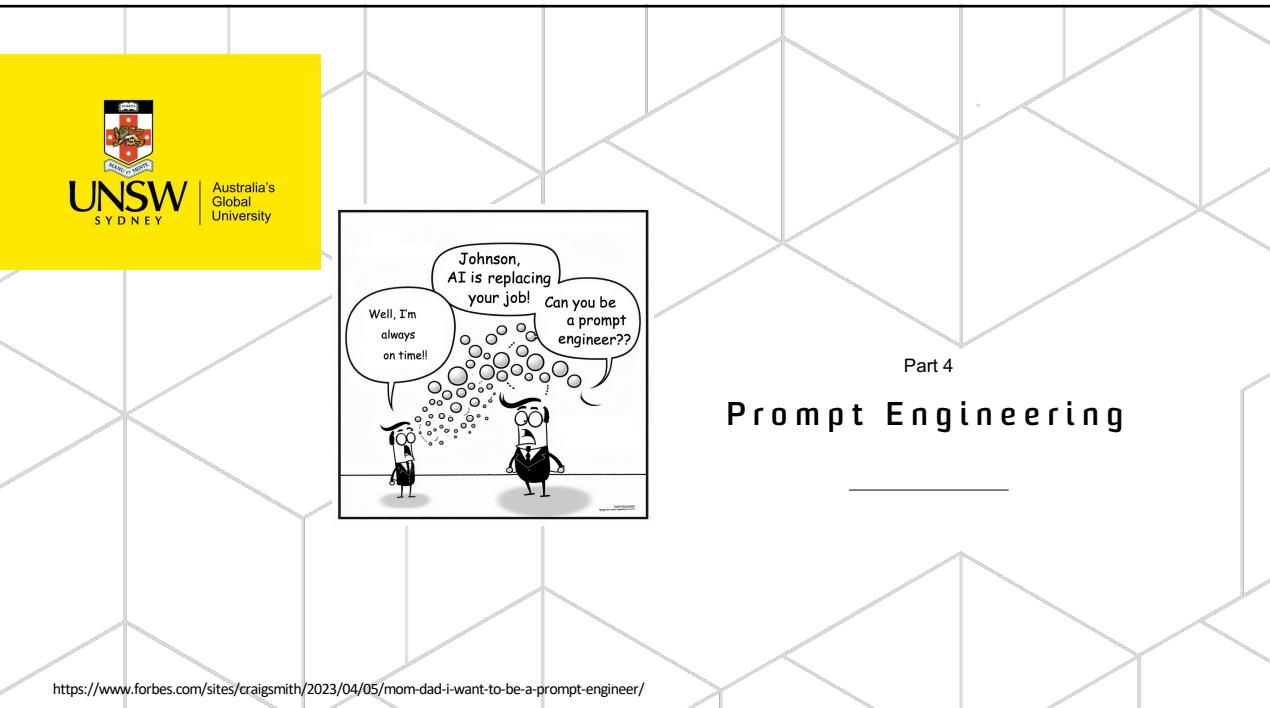


UNSW
SYDNEY | Australia's Global University



Part 4

Prompt Engineering



<https://www.forbes.com/sites/craigsmith/2023/04/05/mom-dad-i-want-to-be-a-prompt-engineer/>

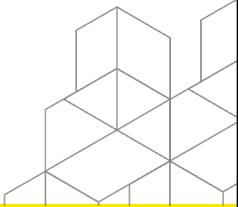
36

What is prompt engineering?

Carefully crafted instructions to obtain a desired output from a language model

"I suspect that machines to be programmed in our native tongues—be it Dutch, English, American, French, German, or Swahili—are as damned difficult to make as they would be to use."
—Edsger Dijkstra, 1979

Prompt: A natural language query to a language model.



Liang, Jenny T., Melissa Lin, Nikitha Rao, and Brad A. Myers. "Prompts are programs too! understanding how developers build software containing prompts." arXiv preprint arXiv:2409.12447 (2024).



37

NEXT GEN INVESTING 38-year-old AI prompt engineer makes over \$100,000 a year—without a tech degree: 'Don't underestimate' yourself

Published Tue, Feb 11 2025 11:05 AM EST

Morgan Smith
@THEWORDSMITHM

SHARE f X in e



Allison Harbin Photo: Allison Harbin

Trending Now

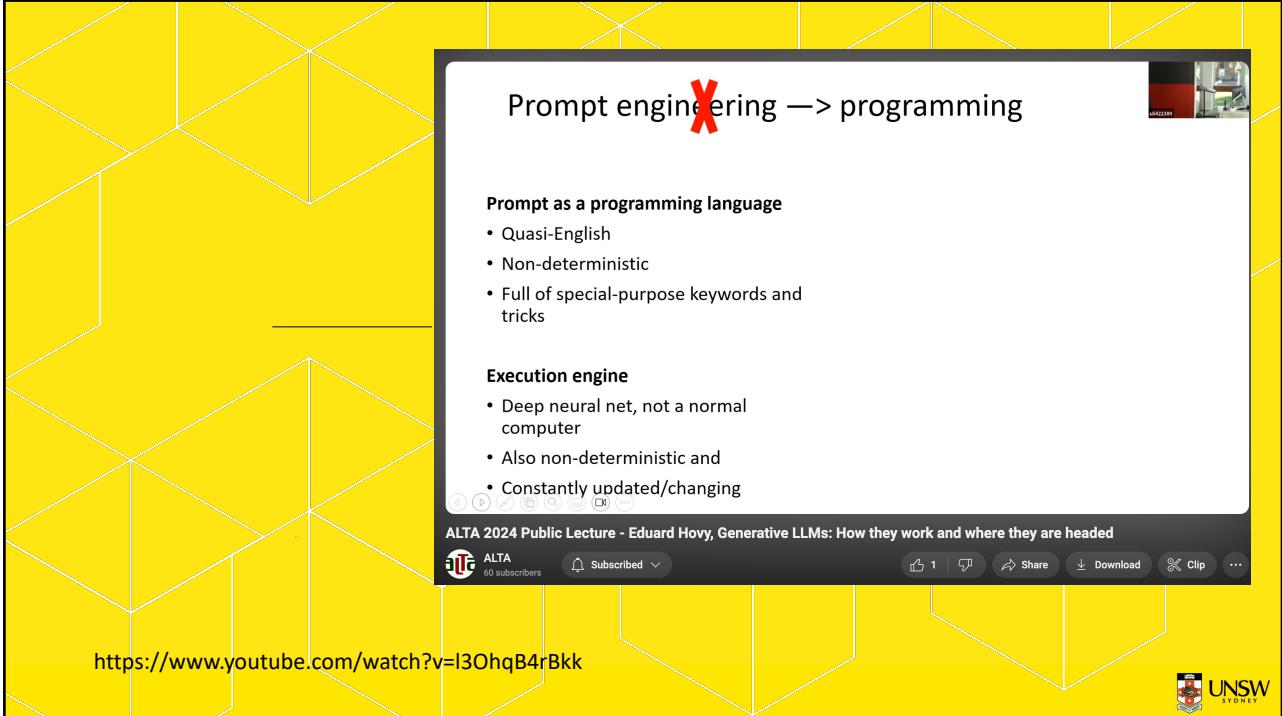
- 1 Psychology expert: 8 signs you're an ambivert—they're 'more successful' than introverts and extroverts
- 2 I spent 48 hours in Finland—here are 5 life-changing lessons I learned in the world's happiest country

<https://www.cnbc.com/2025/02/11/38-year-old-ai-prompt-engineer-makes-over-100000-a-year-without-a-tech-degree.html>



38

19



39

Prompting for New Tasks

Zero-shot prompting (shot: example, demonstration)

Few-shot prompting

Here are a few examples of English to language X translation.

English: Sea shells
Language X: ninaka panaki

English: Sea water
Language X: kolamana panaki

Now, translate the following phrase from English to language X:
"Water shells"

Based on the pattern in your examples, it seems that:

- "Sea" is translated as *kolamana* or *ninaka* (depending on context).
- "Shells" is translated as *panaki*.
- "Water" might be *kolamana* (if "sea water" follows a pattern where "sea" modifies "water").

If "Water shells" follows the same structure, it could be translated as:
kolamana panaki

Let me know if you have more context or if you're following a different pattern! ●

Few-shot prompting is helpful for (new/obscure) tasks.

Selecting shots is a skill.

1-shot prompting, 3-shot prompting, etc.



<https://arxiv.org/pdf/2402.07927.pdf>

40

Prompting for Reasoning: Chain-of-thought (CoT)

Zero-shot CoT

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: Let's think step by step.
(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Few-shot CoT

Standard Prompting	Chain-of-Thought Prompting
<p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> <p>Model Output</p> <p>A: The answer is 27. ✗</p>	<p>Model Input</p> <p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> <p>Model Output</p> <p>A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓</p>

<https://arxiv.org/pdf/2402.07927.pdf>



41

Prompting for Reasoning: Tree-of-thought

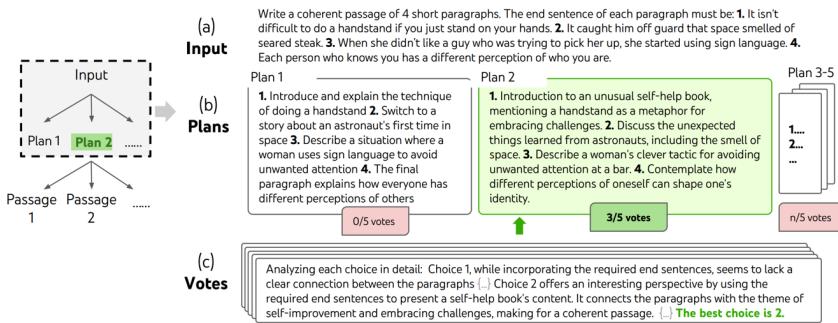


Figure 4: A step of deliberate search in a randomly picked Creative Writing task. Given the input, the LM samples 5 different plans, then votes 5 times to decide which plan is best. The majority choice is used to consequently write the output passage with the same sample-vote procedure.

Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izahak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. "Tree of thoughts: Deliberate problem solving with large language models." *Advances in neural information processing systems* 36 (2023): 11809-11822.



42

Prompting for Reasoning: Step-back prompting

The diagram illustrates the Step-back prompting process for reasoning. It starts with an **Original Question**: "What happens to the pressure, P, of an ideal gas if the temperature is increased by a factor of 2 and the volume is increased by a factor of 8?" This leads to an **Original Answer**: "If the temperature is increased by a factor of 2 and the volume is increased by a factor of 8, then the pressure will decrease by a factor of 16." A **Chain-of-Thought** bubble provides a detailed derivation: "If the temperature is increased by a factor of 2 and the volume is increased by a factor of 8, we can rewrite the ideal gas law as: $2P' (8V) = nR(2T)$, where P' represents the new pressure. Simplifying the equation, we get: $16PV = 2nRT$ or $P' = (2nRT) / (16V)$. Since n, R, and V are constants, we can write this as: $P' = (1/8) * (nRT/V)$. Since nRT/V is the original pressure P, we have: $P' = (1/8) * P$. This means the new pressure P' is 1/8 of the original pressure P..." A **Step-Back Prompting** section shows the model being prompted with "What are the physics principles behind this question?", leading to a **Stepback Answer**: "Ideal gas law: $PV = nRT$, where P is the pressure, V is the volume, n is the number of moles, R is the gas constant, and T is the temperature." The process then moves to **Step 1: Abstraction** and **Step 2: Reasoning**, which lead to the **Final Answer**: "Apply ideal gas law to find out the pressure. If the temperature is increased by a factor of 2, then T becomes 2T. If the volume is increased by a factor of 8, then V becomes 8V. Substituting these values into the ideal gas law, we get: $P(8V) = nR(2T)$. Dividing both sides by 8V, we get: $P = nR(2T) / 8V$. We can see that the pressure has decreased by a factor of 4."

Zheng, Huaixiu Steven, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. "Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models." In *The Twelfth International Conference on Learning Representations*.

43

Prompting for Reasoning: System 2 Attention

Irrelevant information in the prompt can derail the language model

E.g.: Long documents in the prompt

System 2 attention operates over two steps:

Step 1: Prompt an LLM to remove contextual information that is irrelevant to the question

Step 2: Prompt an LLM again with the updated contextual information

Given the following text by a user, extract the part that is unbiased and not their opinion, so that using that text alone would be good context for providing an unbiased answer to the question portion of the text.

Please include the actual question or query that the user is asking. Separate this into two categories labeled with "Unbiased text context (includes all content except user's bias)" and "Question/Query (does not include user bias/preference)".

Text by User: [ORIGINAL INPUT PROMPT]

Problem: Mary has 3 times as much candy as Megan. Mary then adds 10 more pieces of candy to her collection. Max has 1000 more books than Mary. If Megan has 5 pieces of candy, how many does Mary have in total?

LLaMA-2-70B-chat: Solution: ...
Max has 1000 more books than Mary, so Max has $1000 + 3x + 10$ pieces of candy.
2. Now, we can set up an equation using the information we have:
 $1000 + 3x + 10 = 3(5) + 10$
Simplifying the equation:
 $1000 + 3x + 10 = 15 + 10$
 $3x + 1000 - 10 = 15 - 10$
 $3x = 985$
Dividing both sides by 3:
 $x = 985 / 3 = 328$
Therefore, Mary has 328 pieces of candy in total.
Final answer (in numbers): 328

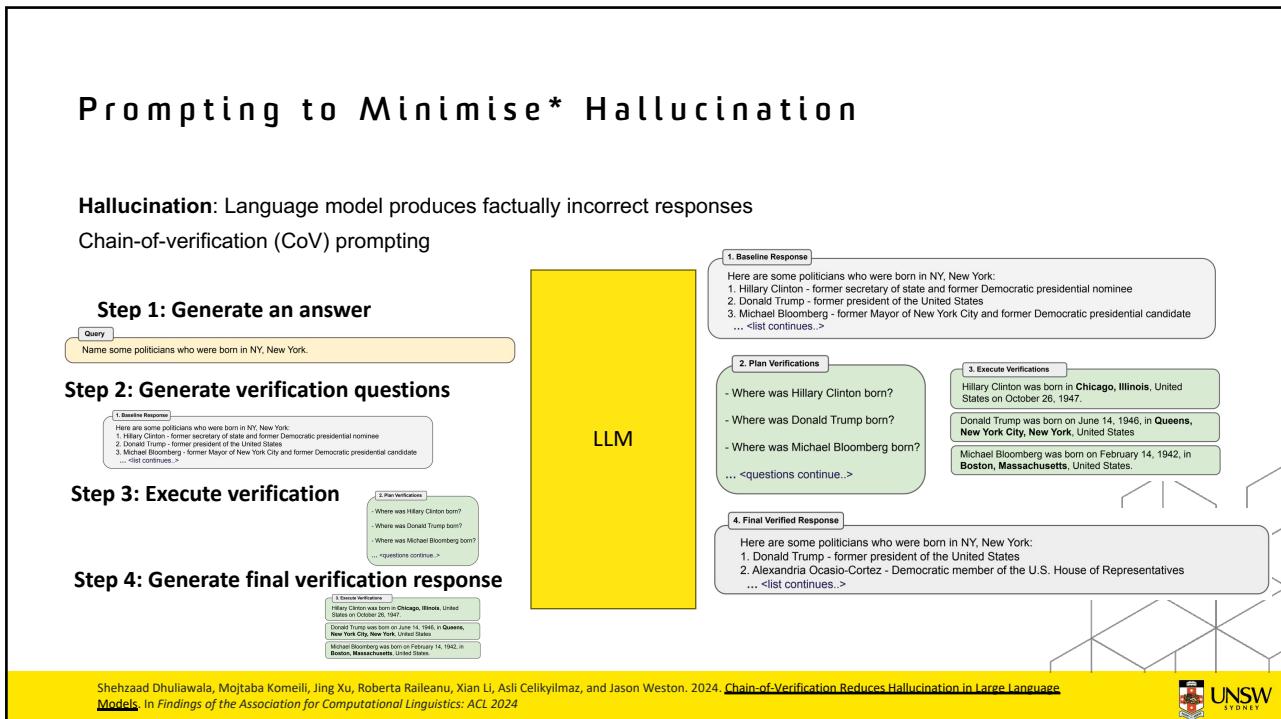
Problem: Mary has 3 times as much candy as Megan. Mary then adds 10 more pieces of candy to her collection. Max has 1000 more books than Mary. If Megan has 5 pieces of candy, how many does Mary have in total?

S2A: Mary has 3 times as much candy as Megan. Mary then adds 10 more pieces of candy to her collection. If Megan has 5 pieces of candy, how many does Mary have in total? Question: How many pieces of candy does Mary have in total?

LLaMA-2-70B-chat: Solution: Let's say Megan has 5 pieces of candy. Mary has 3 times as much candy, so Mary has $3 \times 5 = 15$ pieces of candy. Mary then adds 10 more pieces of candy, so Mary has $15 + 10 = 25$ pieces of candy.
Final answer (in numbers): 25

Weston, Jason, and Sainbayar Sukhbaatar. "System 2 attention (is something you might need too)." *arXiv preprint arXiv:2311.11829* (2023).

44



45



46

GLUE

General Language Understanding Evaluation (GLUE)

Corpus	Train	Test	Task	Metrics	Domain
Single-Sentence Tasks					
CoLA	8.5k	1k	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
Similarity and Paraphrase Tasks					
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	391k	paraphrase	acc./F1	social QA questions
Inference Tasks					
MNLI	393k	20k	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	146	coreference/NLI	acc.	fiction books

Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-B, which is a regression task. MNLI has three classes; all other classification tasks have two. Test sets shown in bold use labels that have never been made public in any form.

Wang, Alex, et al. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." ICLR 2019.

"How do I bake a cake?" &
 "Tell me ways to bake a cake"
 -> Duplicate
 "How do I bake a cake?" &
 "Are cakes healthy" -> Not

He is sleeping. & He is snoring
 -> Entailment
 He is happy. & He is crying. ->
 Contradiction



47

BigBench

User-contributed repository of tasks and datasets

Two kinds of tasks: JSON tasks and programmatic tasks

Evaluation code to test on language models

Explore the tasks on BigBench for project ideas!

Srivastava et al. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models, Transactions on Machine Learning Research.



48

Shared Tasks

Competitions run by workshops/conferences
Datasets are provided to participating teams

SemEval

Semantic Relations

- **Task 1: Semantic Textual Relatedness for African and Asian Languages** ([\[contact organizers\]](#), [\[join task mailing list\]](#))
Nedjma OUSIDHOUM, Shamsudeen Hassan Muhammad, Mohamed Abdalla, Krishnapriya Vishnubhotla, Vladimir Araujo, Meriem Beloucif, Idris Abdulmumin, Seid Muhib Yimam, Thamar Solorio, Monojit Choudhury, Saif M. Mohammad

- **Task 2: Safe Biomedical Natural Language Inference for Clinical Trials** ([\[contact organizers\]](#), [\[join task mailing list\]](#))
Mael Jullien, Marco Valentino, Andre Freitas

Discourse and Argumentation

- **Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations** ([\[contact organizers\]](#), [\[join task mailing list\]](#))
Rui Xia, Jianfei Yu, Fanfan Wang, Erik Cambria
- **Task 4: Multilingual Detection of Persuasion Techniques in Memes** ([\[contact organizers\]](#), [\[join task mailing list\]](#))
Dimitar Iliyanov Dimitrov, Giovanni Da San Martino, Fabrizio Silvestri, Preslav Nakov, Firoj Alam
- **Task 5: Argument Reasoning in Civil Procedure** ([\[contact organizers\]](#), [\[join task mailing list\]](#))
Lena Held, Ivan Habernal

<https://semeval.github.io/SemEval2024/tasks.html>



49

Time for an interesting
library...

Also see: Labeled datasets available on HuggingFace



50

LangChain

"Augmented language models"

LangChain is a framework that allows to combine LLMs with other components through composability.

Primarily used to 'productionise' LLMs, LangChain has the potential to be used in inter-disciplinary applications of LLMs.

Many tools and plugins – we will cover a *really* small subset. Do check out the documentation.

Hope it inspires project ideas!

<https://github.com/hwchase17/langchain>
https://python.langchain.com/docs/get_started/introduction.html

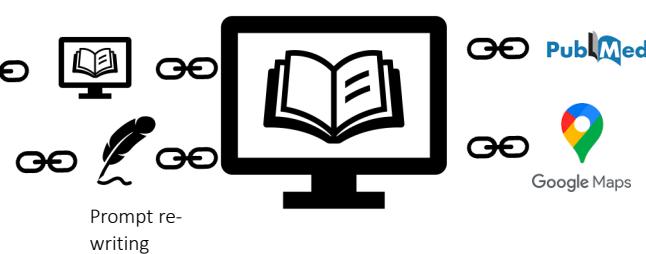


51

LangChain allows language model augmentation

Augmented language models (LMs) combine foundation LMs with reasoning and external tools.

Reasoning may decompose a complex task into subtasks or chain-of-thoughts.



Augmented LMs utilise the additional information to guide their output.

<https://arxiv.org/abs/2302.07842>



52

Key Elements of LangChain

			
Prompt	LLMs	Tools	Chains
Prompt Templates ..with input parameters	HuggingFace OpenAI Databricks ...also chat versions of LLMs	Google Search/Places DuckDuckGo ShellTool Wikipedia PubMed/arXiv	Sequential Router (!!)



53

Chains (1/3)

Write an essay.  Here's the essay ...

LLM as an essay generator

"kangaroos"  "Write an essay about kangaroos"  Here's the essay about...:

"kangaroos" "Australia"  "Write an essay about kangaroos in Australia"  Here's the essay: ...

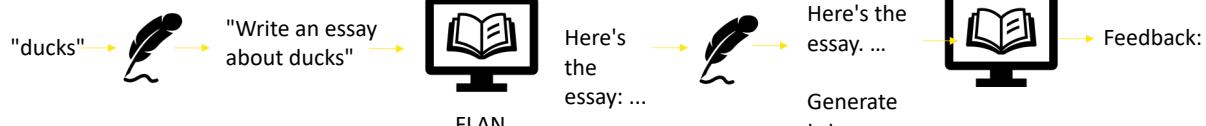
Input parameters can be chained into a prompt.



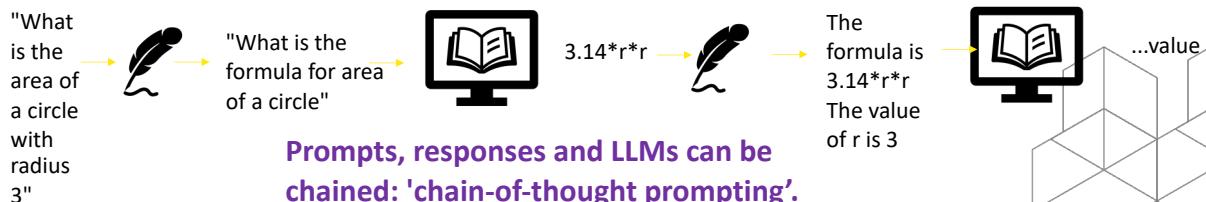
54

Chains (2/3)

LLM as an essay and feedback generator



LLM for complex reasoning



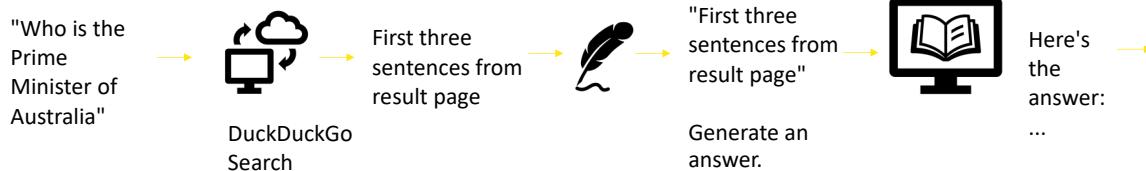
Prompts, responses and LLMs can be chained: 'chain-of-thought prompting'.



55

Chains (3/3)

LLMs and tools chained together



LLMs and Tools/APIs can be chained for recent/reliable information.



56

LangChain Demo

<https://github.com/adityajo/langchain-tutorial>

The sample notebook is ONLY illustrative. LangChain is a fast-evolving library.

If you do update the notebooks with recent LangChain functions, please send a pull request.



Demo time!



57

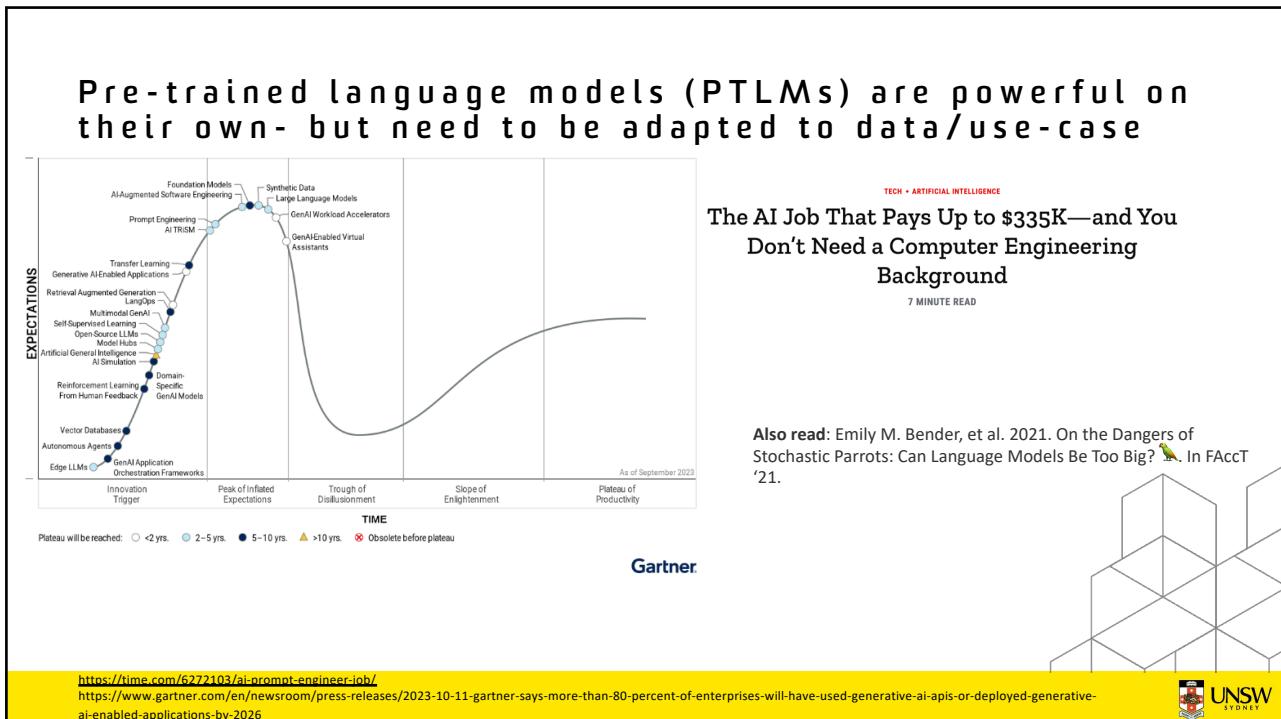
Summary

Part	Key Idea	Demos
Derivatives of Transformer	Encoder and decoder models	
Encoder models	BERT pre-training and fine-tuning; Variants of BERT. “BERT is a vacuum cleaner”	BERT
Decoder models	GPT; prompting	GPT prompting
Prompt engineering	Zero/few-shot prompting, Chain-of-thought, tree-of-thought, etc.	Examples
Fine-tuning methods	PEFT, Prompt/prefix-tuning, LoRA	Simple LoRA, OPT Fine-tuning using PEFT
Datasets & Libraries	NLP benchmarks and LangChain	LangChain

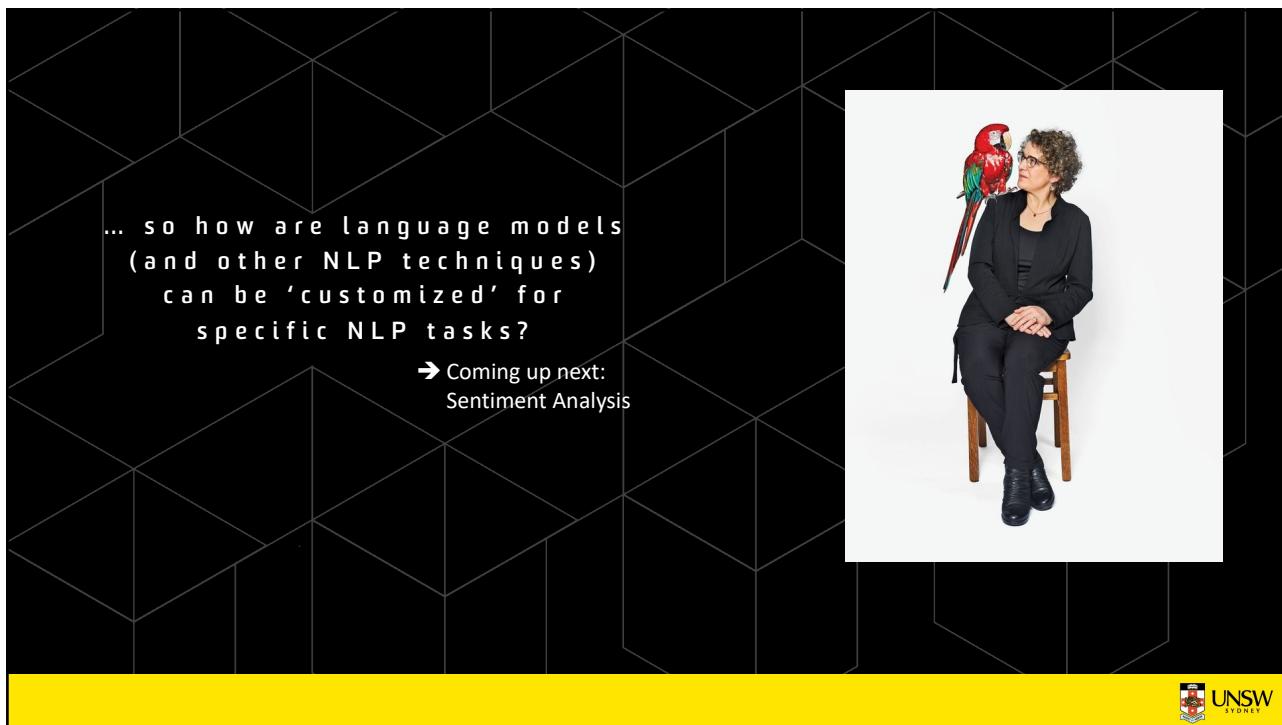


58

29



59



60

Suggested Reading

Textbook chapter: <https://web.stanford.edu/~jurafsky/slp3/11.pdf>

BERT: Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL.

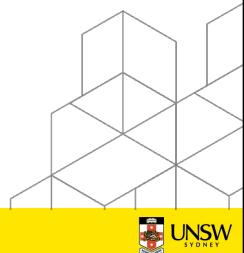
<https://jalammar.github.io/illustrated-bert/>

GPT: Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

LoRA: Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *ICLR*. 2021.

Strongly recommended: <https://github.com/huggingface/peft>

Also see: References on previous slides.



61

Epilogue

If you are interested ...



62

Input in BERT is represented as an addition of token embeddings, segment embeddings and position embeddings

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}

