Pre-training Strategies

Tanmoy Chakraborty
Associate Professor, IIT Delhi
https://tanmoychak.com/

LABORATORY FOR COMPUTATIONAL SOCIAL SYSTEMS

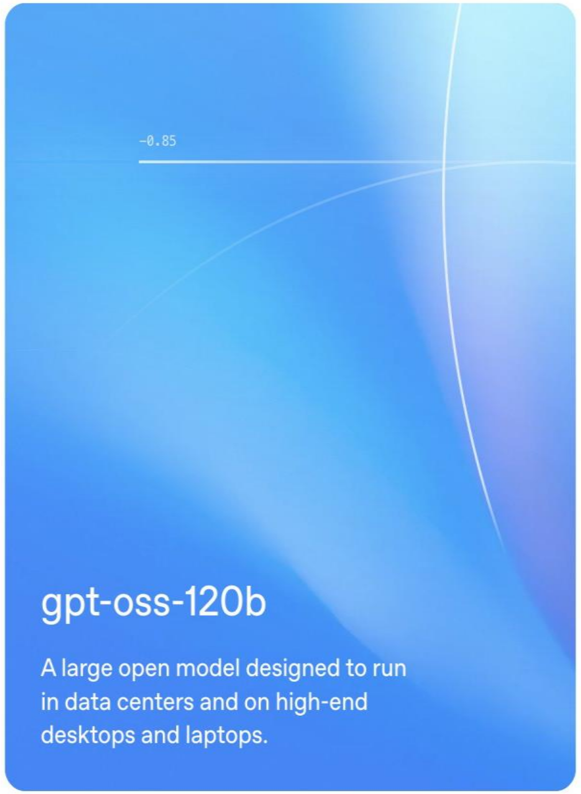
OpenAI introduces GPT-OSS

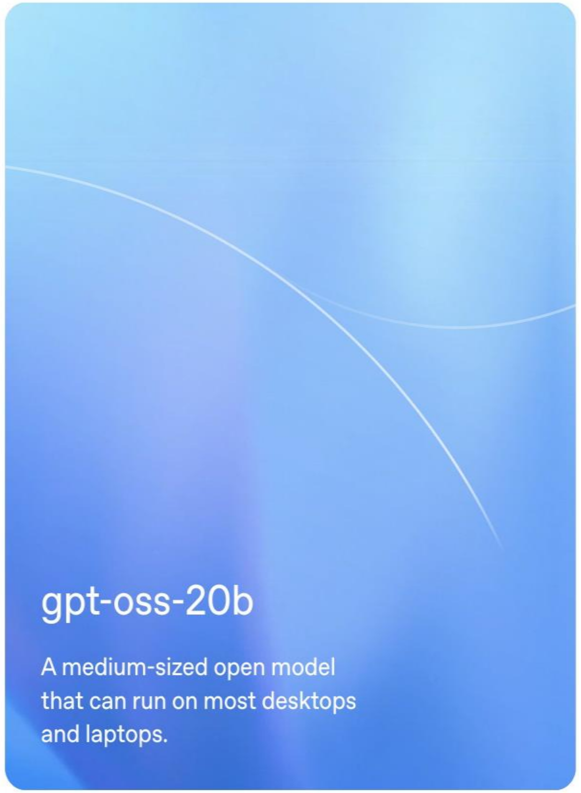
Announced on August 5, 2025

OpenAI Blog

An open weights model with strong reasoning performance

The **120B** model is on par with o4-mini on reasoning benchmarks , while running efficiently on a single 80 GB GPU





They also released a **20b** model, which shows similar performance to that of o3-mini. It only requires 16 GB of memory and can easily run on edge devices, making it ideal for local inference.

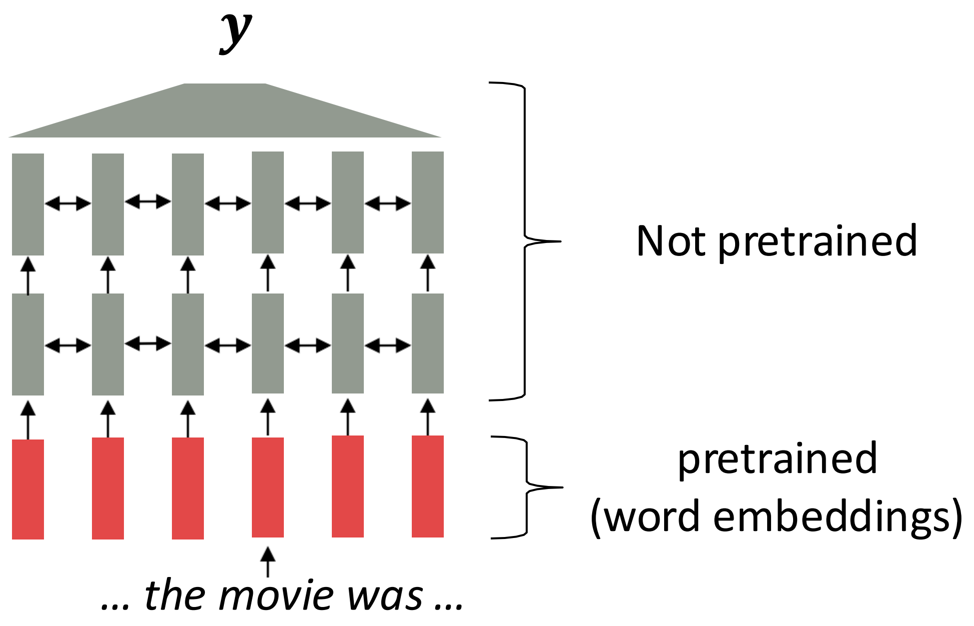
This is a huge deal, allowing people to run state-of-the-art gpt models locally on their devices

Where We Were: Pre-trained Word Vectors

ELMO

Context- independent / Static

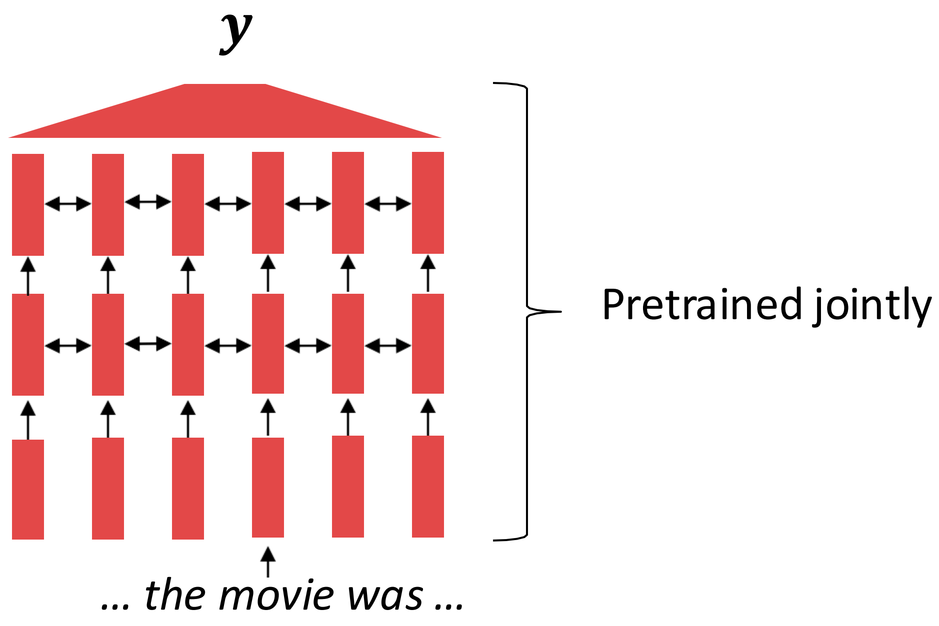
- Start with pretrained word embeddings (no context!)
- Learn how to incorporate context in an LSTM or Transformer while training on the task.
- The training data we have for our **downstream task** (like question answering) must be sufficient to teach all contextual aspects of language.
- Most of the parameters in our network **are randomly initialized**!



Pre-trained Word Vectors -> Pre-trained Models

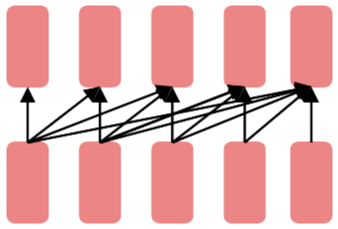
- All (or almost all) parameters in NLP networks are initialized via **pretraining**.
- Pretraining methods hide parts of the input from the model, and train the model to reconstruct those parts.

- This has been exceptionally effective at building strong:
  - **representations of language**
  - **parameter initializations** for strong NLP models.
  - **Probability distributions** over language that we can sample from



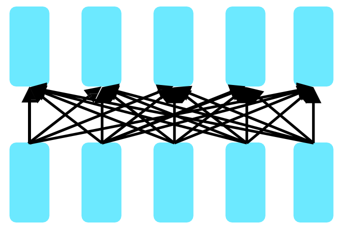
Pretraining for Three Types of Architectures

The neural architecture influences the type of pretraining, and natural use cases.



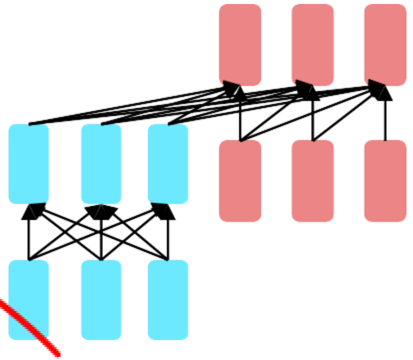
Decoders

- Language models! What we've seen so far.
- Nice to generate from; can't condition on future words



Encoders

- Gets bidirectional context – can condition on future!
- How do we pretrain them?



Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Slides are adopted from Jacob Devlin

Self-Supervised learning:

Background - Bidirectional Context

- Bidirectional context, unlike unidirectional context, takes into account both the left and right contexts.

Target Word

Apple is my favourite fruit and I eat it all the time.

Left Context

Right Context

Target Word

Apple is my favourite brand for buying laptop and other gadgets.

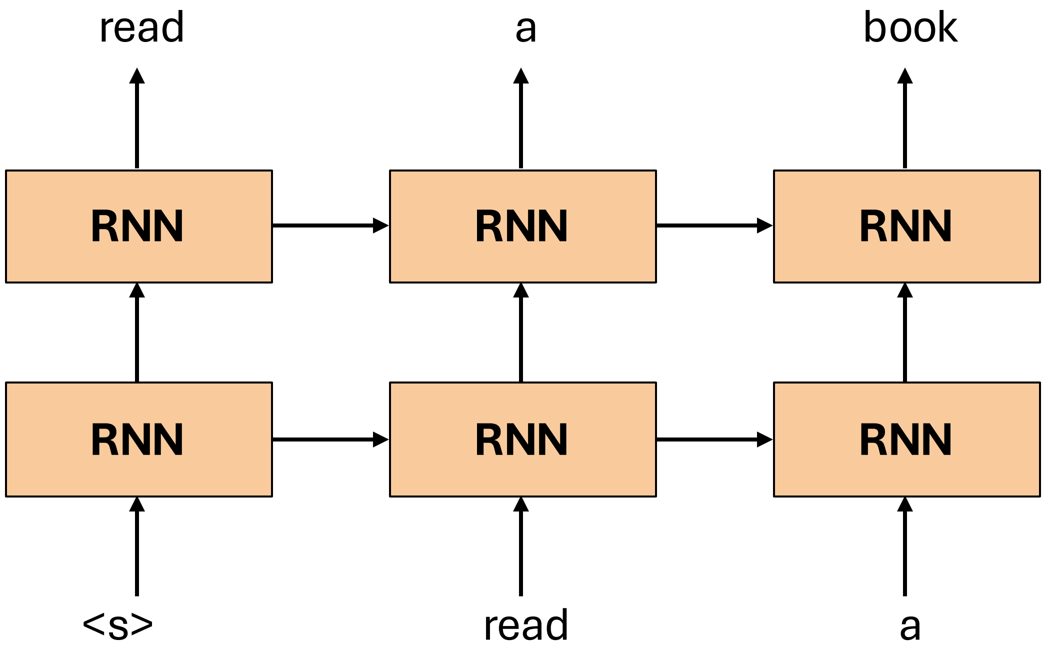
Left Context

Right Context

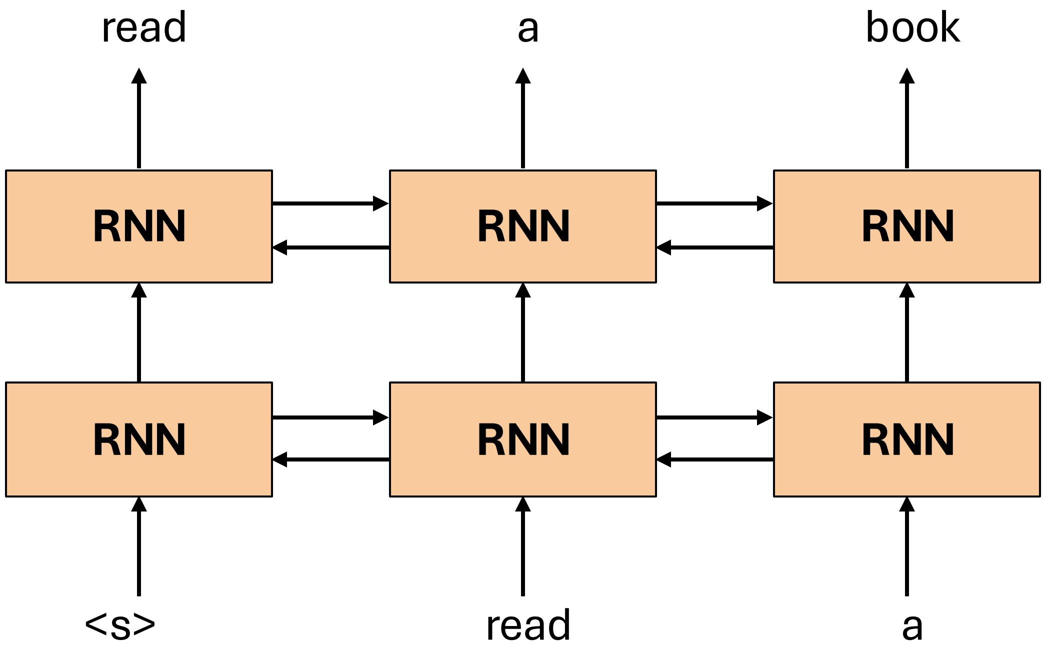
Motivation

- **Problem with previous methods:**
  - Language models only use left context or right context.
  - But language understanding is **bidirectional**.
- **Possible Issue:**
  - Directionality is needed to generate a well-formed probability distribution.
  - Words can see themselves in a bidirectional model.

Unidirectional vs. Bidirectional Models



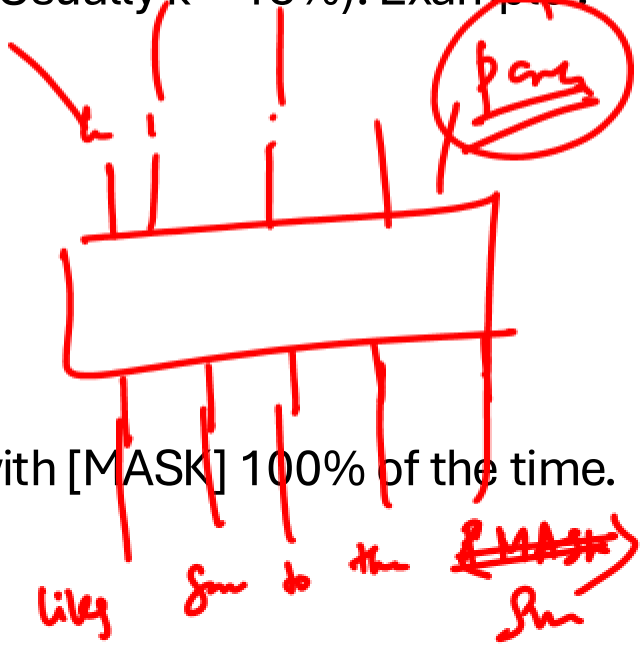
Unidirectional



Bidirectional

Masked Language Modelling

- Mask out k% of the input words, and then predict the masked words (Usually k = 15%). Example :



I like going to the [MASK] in the evening

↓

park

  - Too little masking: Too expensive to train
  - Too much masking: Not enough context

- The model needs to predict 15% of the words, but we don't replace with [MASK] 100% of the time. Instead:
  - 80% of the time, **replace with [MASK]**
    - Example : like going to the park → like going to the [MASK]
  - 10% of the time, **replace random word**
    - Example : like going to the park → like going to the store
  - 10% of the time, **keep same**
    - Example : like going to the park → like going to the park

Next Sentence Prediction

S_1, S_2, S_3, S_4

- To learn relationships between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence.

(S_i, S_{i+1})

Input = [CLS] I enjoy read [MASK] book ##s [SEP]
I finish ##ed a [MASK] novel [SEP]
Label = IsNext

(S_i, S_j)

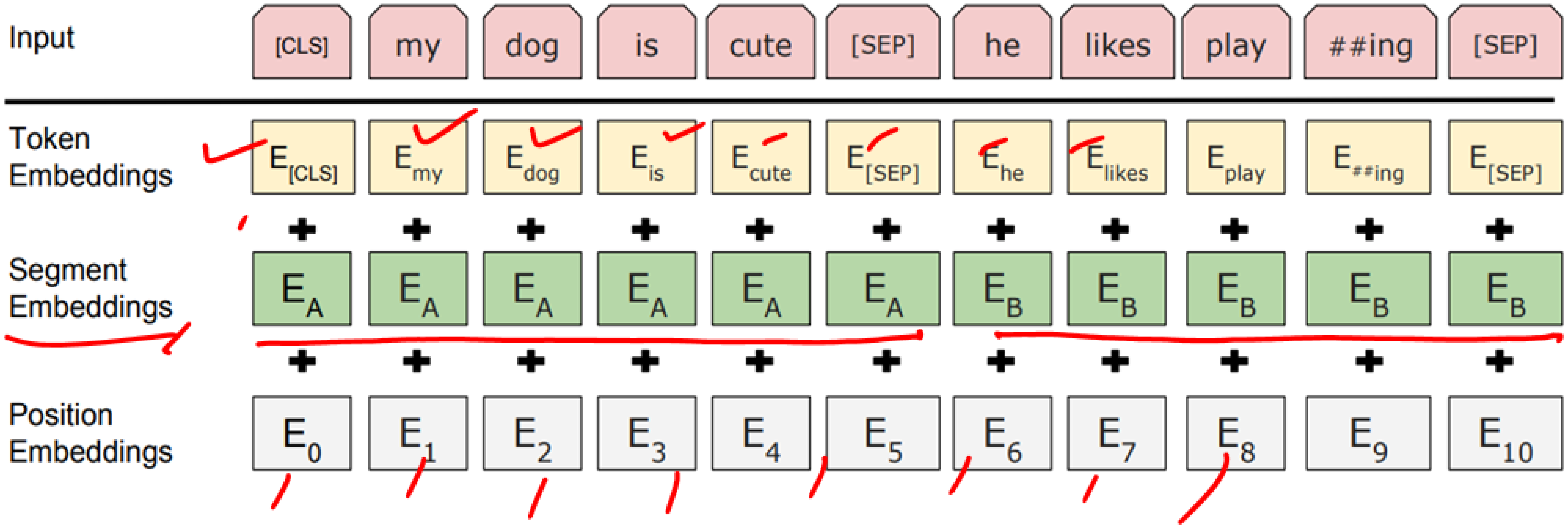
Input = [CLS] I enjoy read ##ing book [MASK] [SEP]
The dog ran [MASK] the street [SEP]
Label = NotNext

[CLS] I am going to [MASK] [SEP] [MASK] is for away

- Important for many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI)
- How to choose sentences A and B for pretraining?
  - 50% of the time B is the actual next sentence that follows A (labeled as IsNext)
  - 50% of the time it is a random sentence from the corpus (labeled as NotNext)

Input Representation

- Use 30,000 WordPiece vocabulary on input.
- For a given token, its input representation is constructed by summing the token embeddings, the segmentation embeddings and the position embeddings.

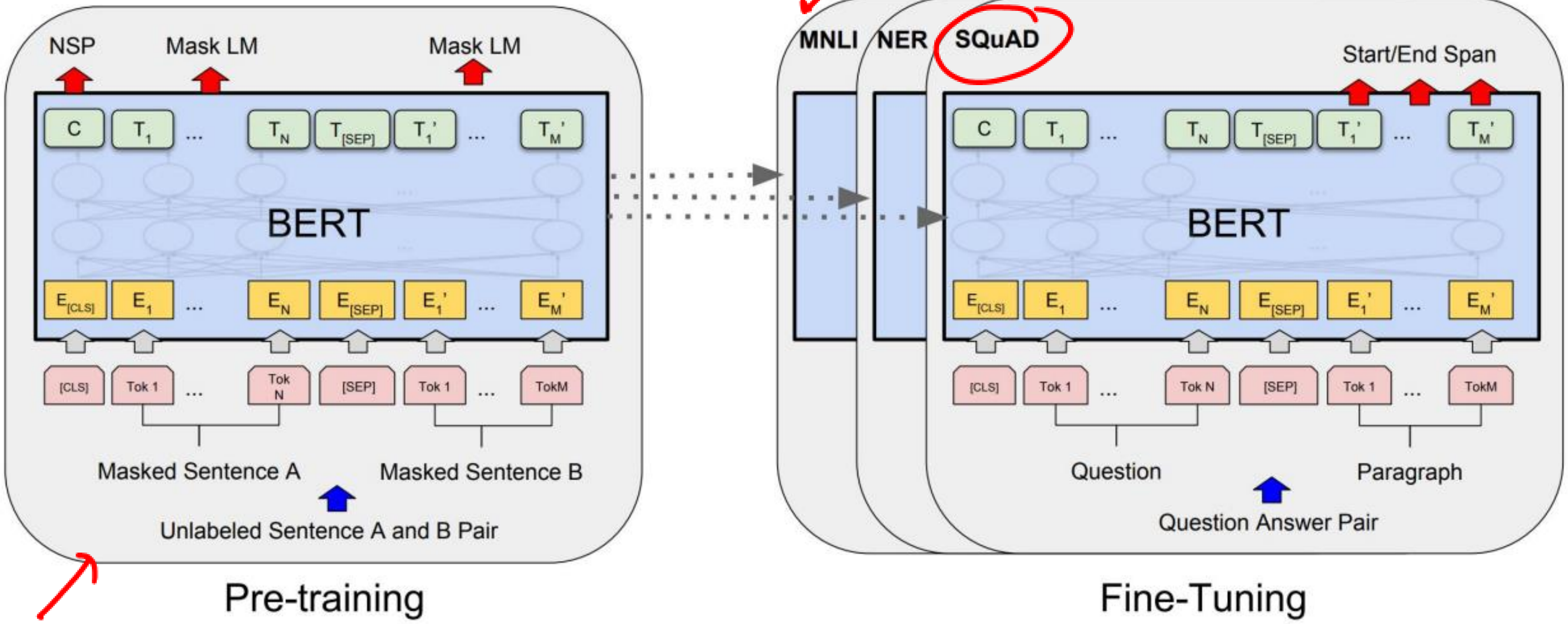


Source of Image : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., NAACL 2019)

Training Details

- Data: Wikipedia (2.5B words) + BookCorpus (800M words)
- Batch Size: 131,072 words (1024 sequences * 128 length or 256 sequences * 512 length)
- Training Time: 1M steps (~40 epochs)
- Optimizer: AdamW, 1e-4 learning rate, linear decay
- BERT-Base: 12-layer, 768-hidden, 12-head
- BERT-Large: 24-layer, 1024-hidden, 16-head
- Trained on 4x4 or 8x8 TPU slice for 4 days

Fine-Tuning Procedure



Pre-Training Encoder-Decoder Models

T5

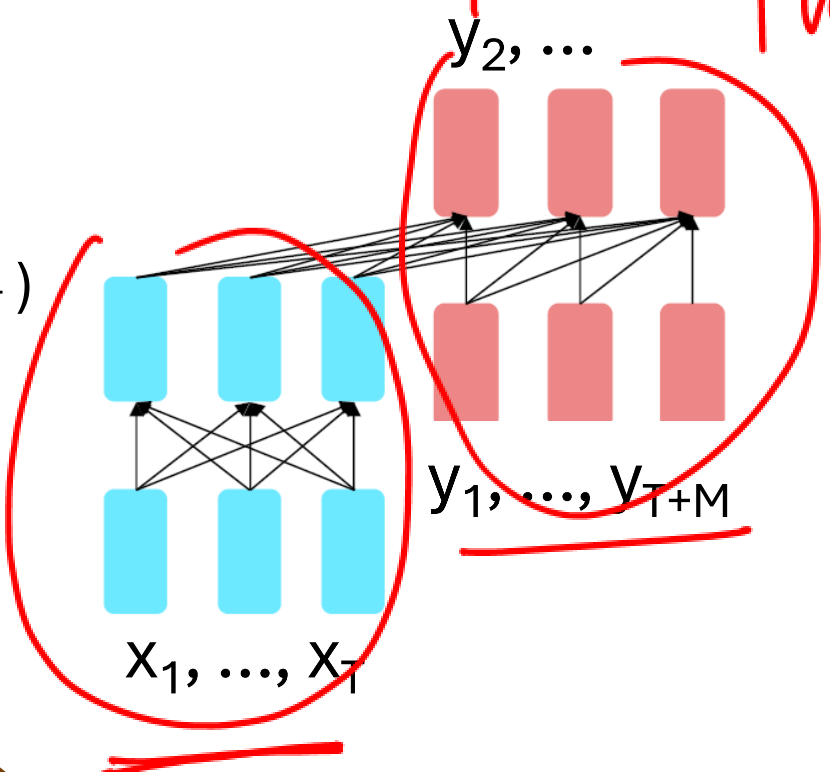
- For **encoder-decoders**, we could do something like **language modeling**, but where a prefix of every input is provided to the encoder and is not predicted.

$\uparrow \log P(y_2 | y_1, \ldots$

$$h_1, \ldots, h_T = \textit{Encoder}\,(x_1, \ldots, x_T)$$

$$h_{T+1}, \ldots, h_{T+M} = \textit{Decoder}\,(y_1, \ldots, y_{i-1}, h_1, \ldots, h_T)$$

$$P(y_i \mid y_{<i}, h_{1:T}) = \textit{Softmax}(Wh_i + b)$$



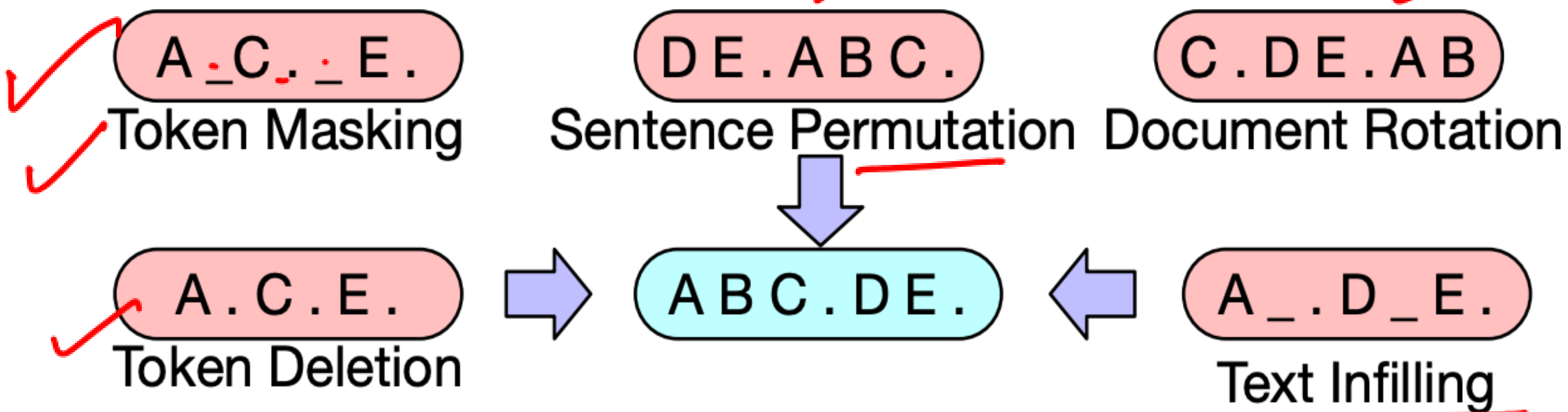
The **encoder** portion benefits from bidirectional context; the **decoder** portion is used to train the whole model through language modeling.

Pre-Training Encoder-Decoder Models

- How can we pre-train a model for P(**y** | **x**)?
- **Requirements:**
  1. should use unlabeled data
  2. should force a model to attend from **y** back to **x**

Pre-Training BART (Bidirectional and Auto-Regressive Transformers)

En-Dec | T5



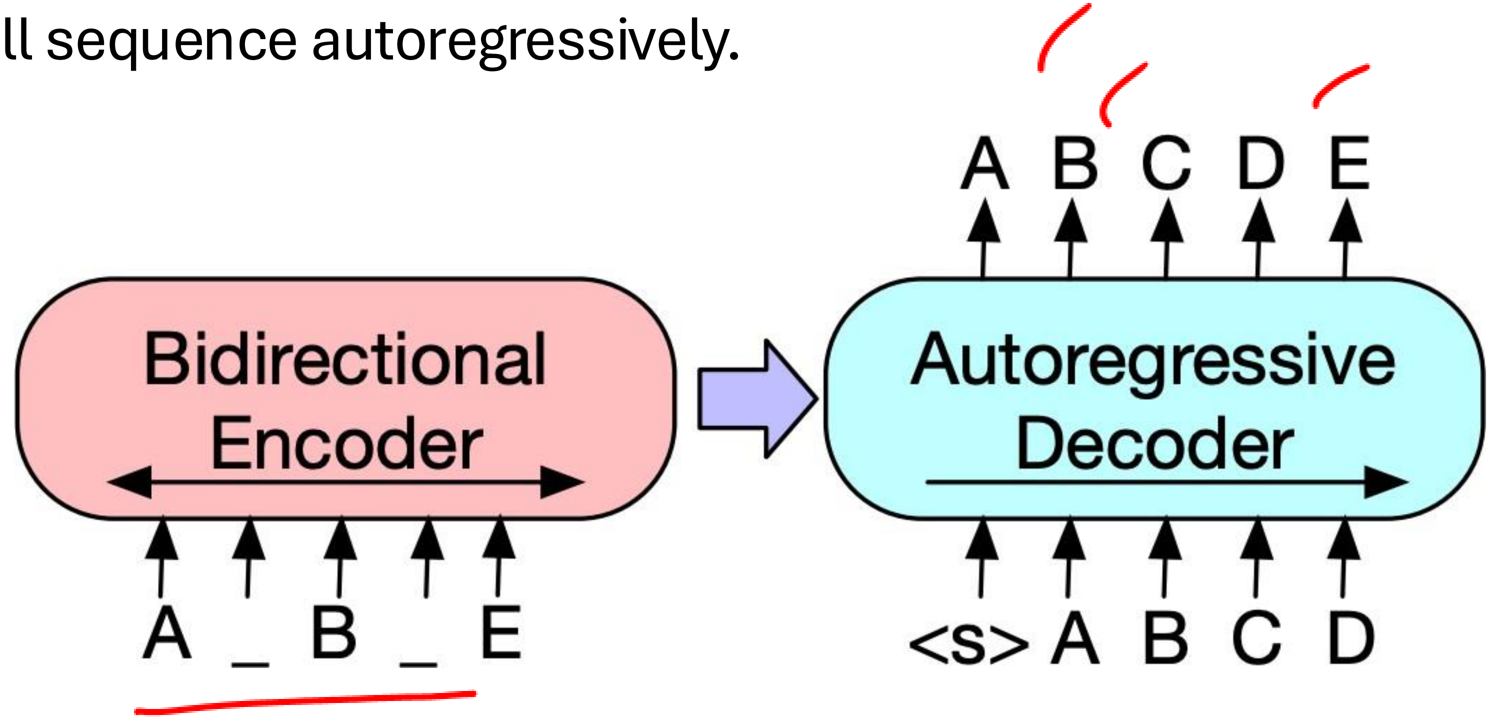
Infilling is longer spans than masking

- Several possible strategies for corrupting a sequence are explored in the BART paper.

Lewis et al. (2019), "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension"

Pre-Training BART

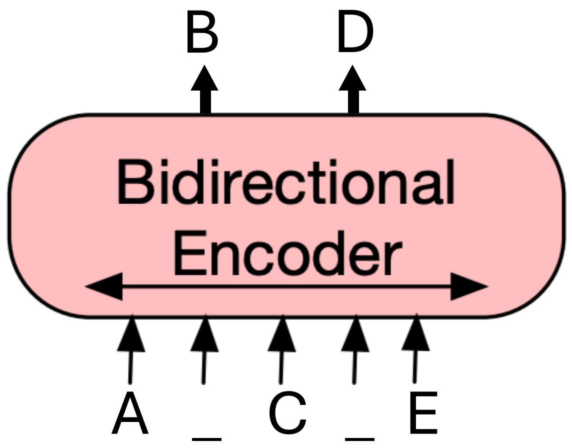
- Sequence-to-sequence Transformer trained on this data: permute/make/delete tokens, then predict full sequence autoregressively.

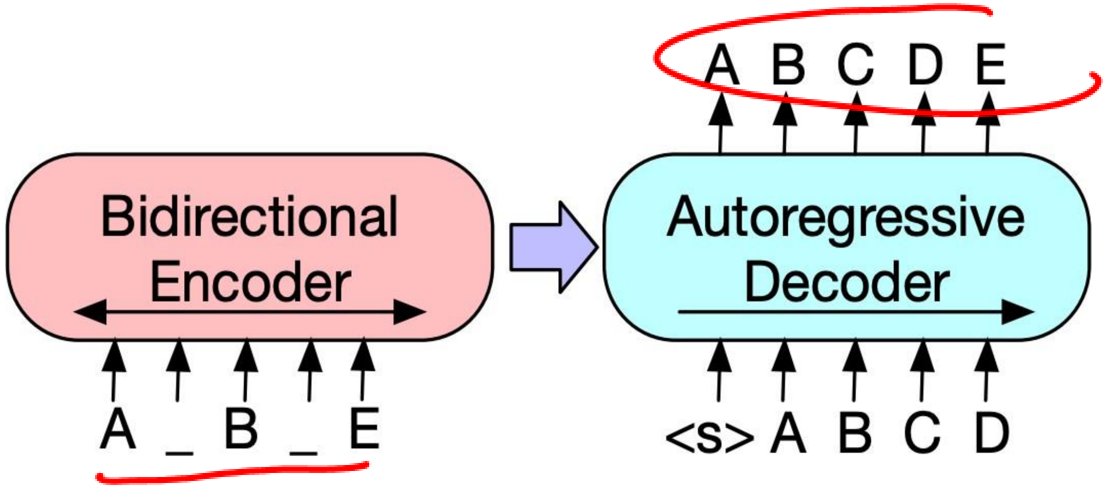


Lewis et al. (2019), “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”

BERT vs. BART

- **BERT:** only an encoder, trained with masked language modeling objective. Cannot generate text or do Seq2Seq tasks (in standard form).



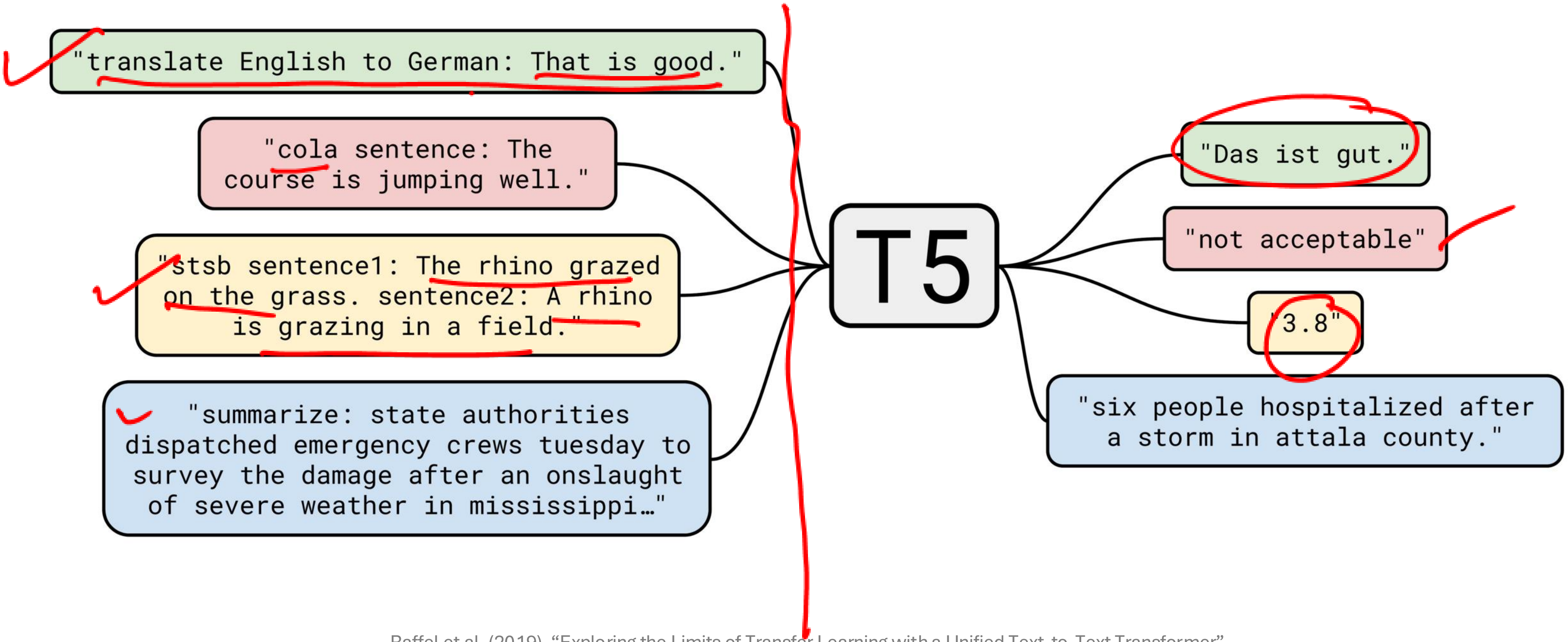


- **BART:** consists of both an encoder and a decoder. Can also use just the encoder wherever we would use BERT.

BART for Summarization

- **Pre-train** on the BART task: take random chunks of text, noise them according to the schemes described, and try to "decode" the clean text
- **Fine-tune** on a summarization dataset: a news article is the input and a summary of that article is the output (usually 1-3 sentences depending on the dataset)
- Can achieve good results even with **few summaries to fine-tune on**, compared to basic seq2seq models which require 100k+ examples to do well

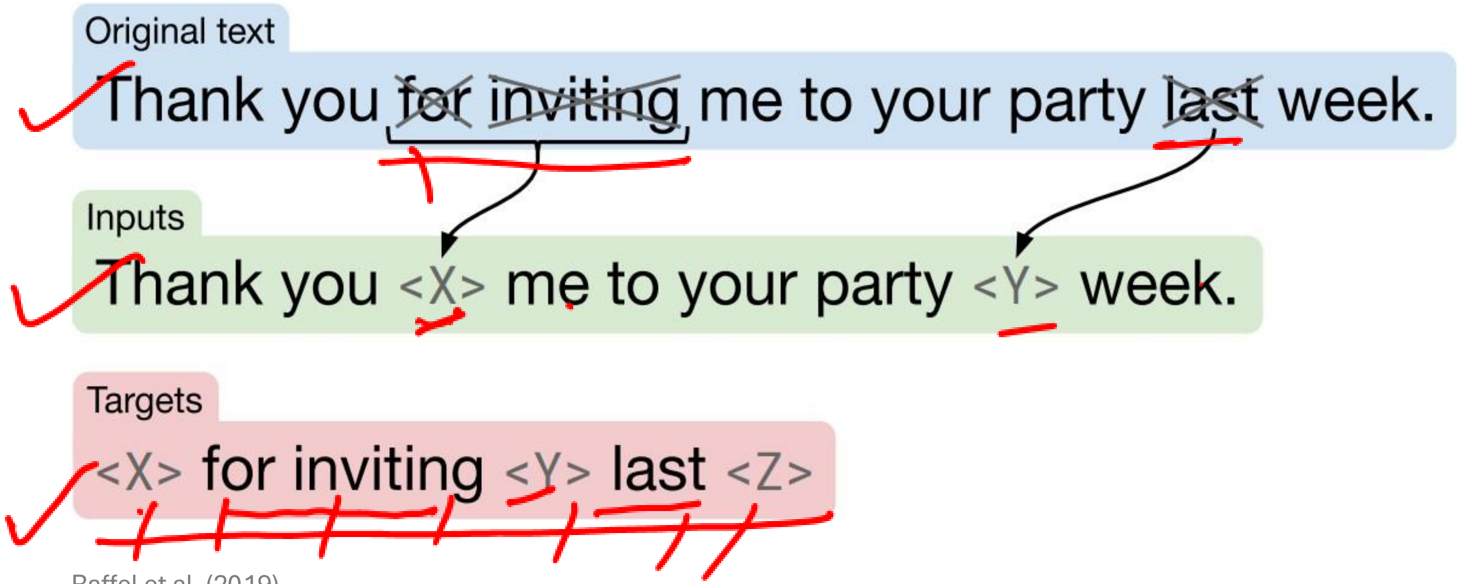
T5: Text-to-Text Transfer Transformer



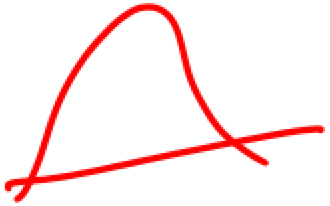
Raffel et al. (2019), "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer"

Pre-Training T5

- **Pre-training:** similar denoising scheme to BART (they were released within a week of each other in fall 2019)
- **Input:** text with gaps ; **Output:** a series of phrases to fill those gaps.



Raffel et al. (2019)



Replace different-length spans from the input with unique placeholders; decode out the spans that were removed!

Pretrain

BERT$_{\text{BASE}}$-sized encoder-decoder Transformer

Denoising objective

C4 dataset

Finetune

GLUE

Pretrain

BERT$_{BASE}$-sized encoder-decoder Transformer

Denoising objective

C4 dataset

GLUE Benchmark

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = **Ungrammatical**	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = **.93056 (Very Positive)**	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = **A Paraphrase**	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = **4.6 (Very Similar)**	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = **Not Similar**	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = **Contradiction**	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = **Answerable**	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = **Entailed**	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = **Incorrect Referent**	Accuracy

Finetune

GLUE

CNN/DM

Pretrain

$\text{BERT}_{\text{BASE}}$-sized encoder-decoder Transformer

Denoising objective

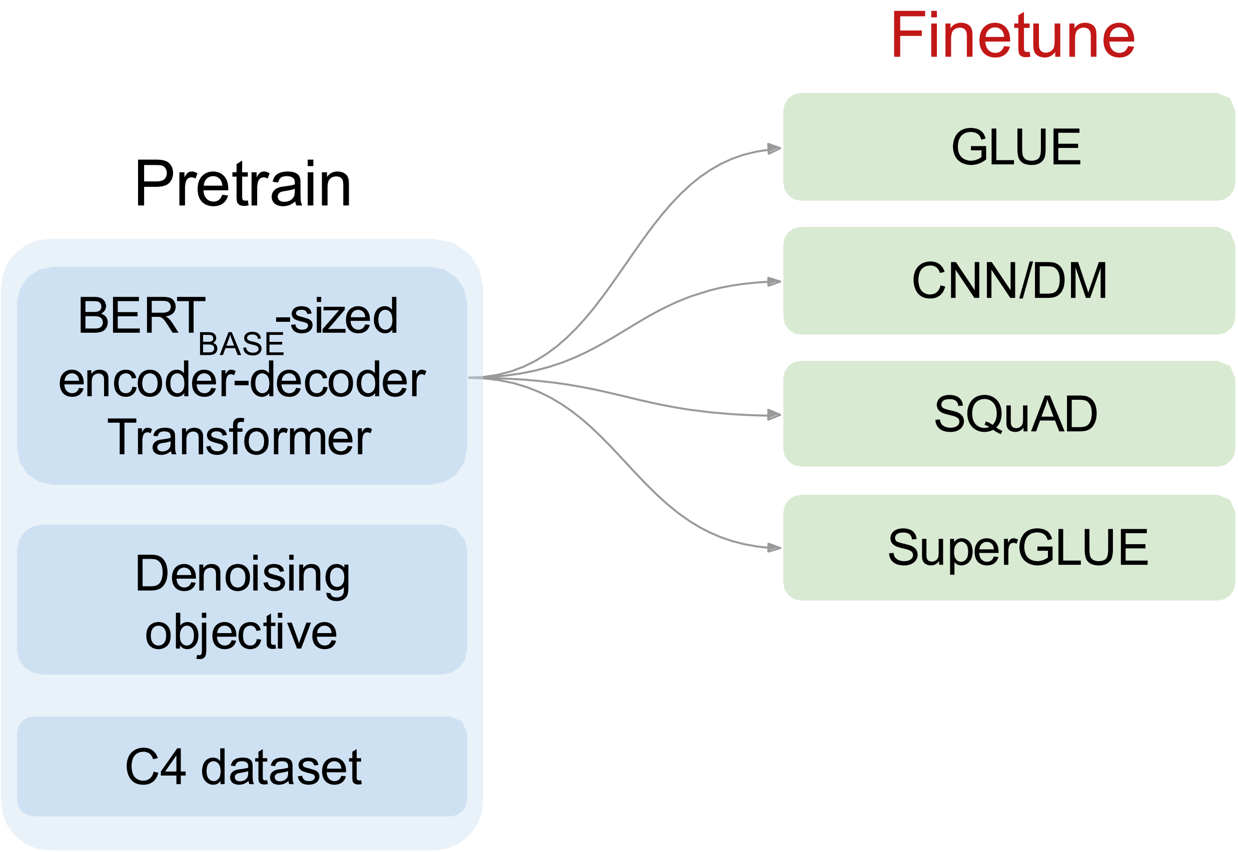
C4 dataset

Finetune

- GLUE
- CNN/DM
- SQuAD

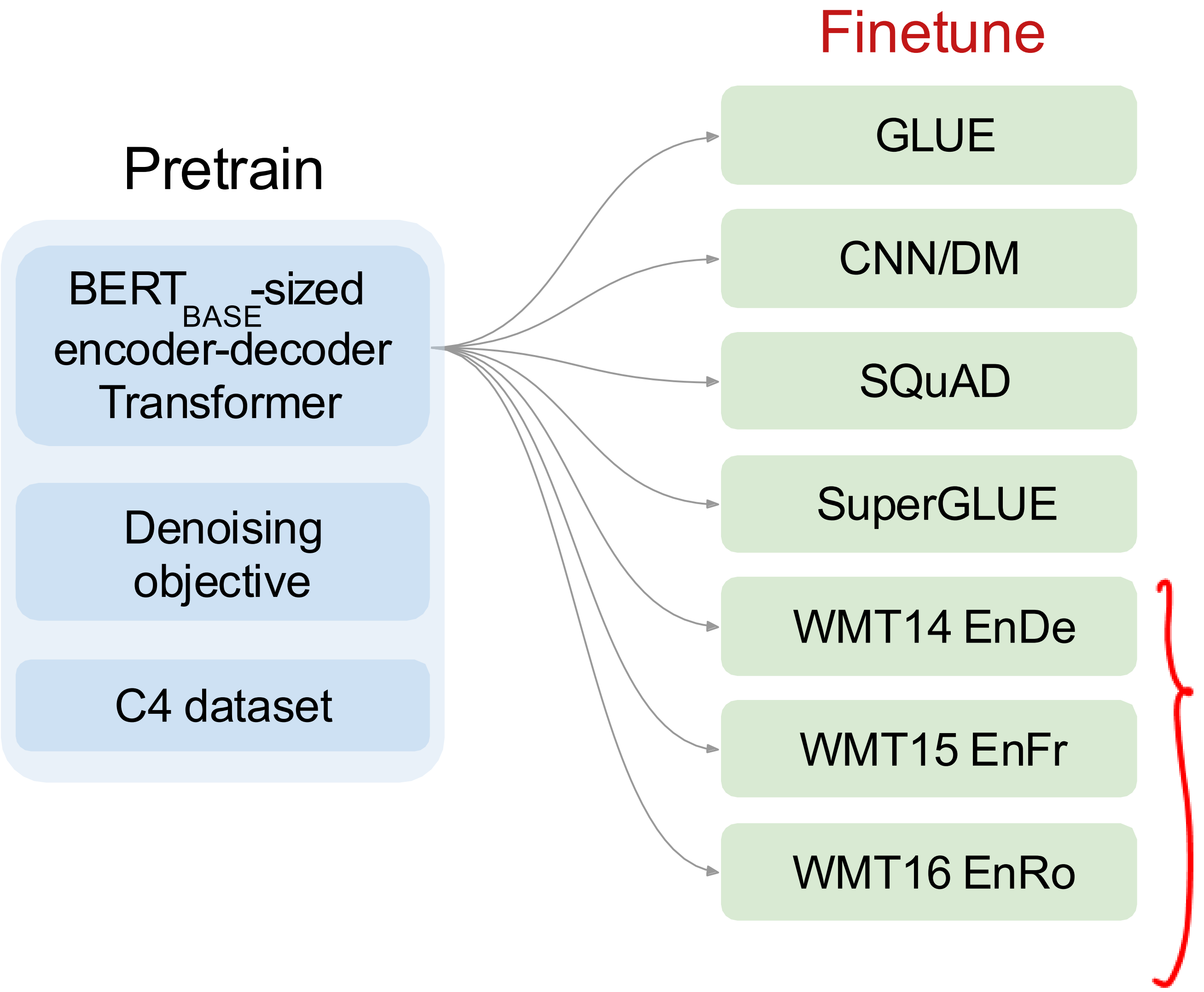
Pretrain

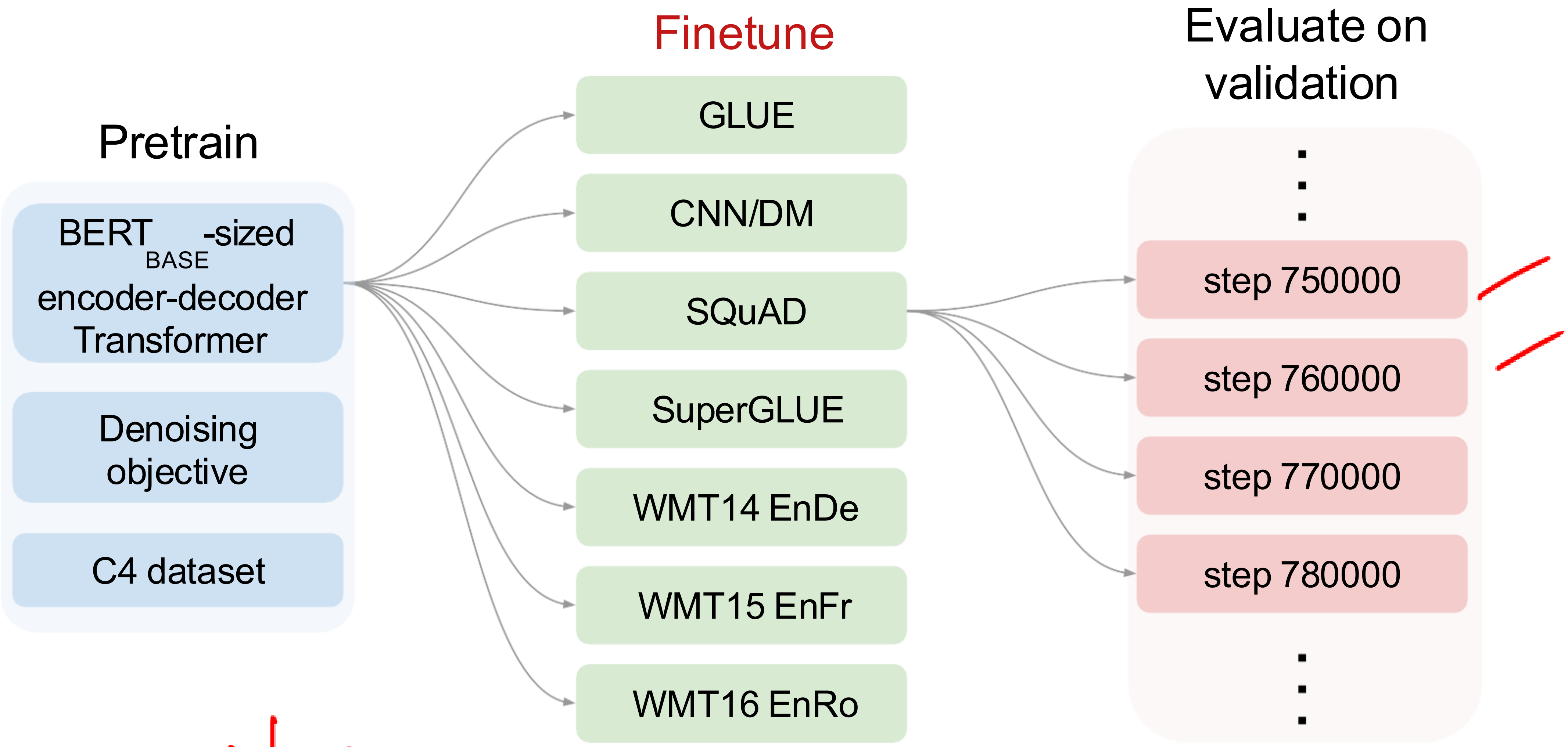
- $\text{BERT}_{\text{BASE}}$-sized encoder-decoder Transformer
- Denoising objective
- C4 dataset


Finetune
Pretrain
BERT$_{BASE}$-sized encoder-decoder Transformer
Denoising objective
C4 dataset
GLUE
CNN/DM
SQuAD
SuperGLUE

SuperGLUE Tasks

Name	Identifier	Download	More Info	Metric
Broadcoverage Diagnostics	AX-b			Matthew's Corr
CommitmentBank	CB			Avg. F1 / Accuracy
Choice of Plausible Alternatives	COPA			Accuracy
Multi-Sentence Reading Comprehension	MultiRC			F1a / EM
Recognizing Textual Entailment	RTE			Accuracy
Words in Context	WiC			Accuracy
The Winograd Schema Challenge	WSC			Accuracy
BoolQ	BoolQ			Accuracy
Reading Comprehension with Commonsense Reasoning	ReCoRD			F1 / Accuracy
Winogender Schema Diagnostics	AX-g			Gender Parity / Accuracy


Pretrain
BERT BASE -sized encoder-decoder Transformer
Denoising objective
C4 dataset
Finetune
GLUE
CNN/DM
SQuAD
SuperGLUE
WMT14 EnDe
WMT15 EnFr
WMT16 EnRo


Pretrain
BERT$_{\text{BASE}}$-sized encoder-decoder Transformer
Denoising objective
C4 dataset
Finetune
GLUE
CNN/DM
SQuAD
SuperGLUE
WMT14 EnDe
WMT15 EnFr
WMT16 EnRo
Evaluate on validation
step 750000
step 760000
step 770000
step 780000

C4 | mC4

Instruction Tuning

Tanmoy Chakraborty
Associate Professor, IIT Delhi
https://tanmoychak.com/

GPT-5: Next Gen AI

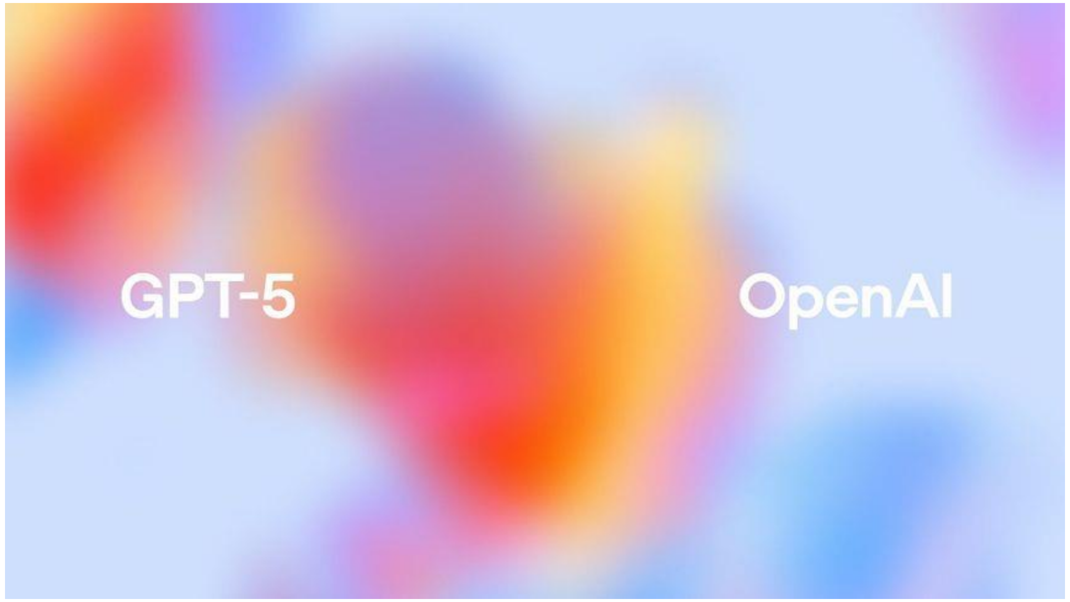
Announced on August 7, 2025

[OpenAI Blog]

OpenAI officially launched GPT-5, marking the biggest leap in artificial intelligence since GPT-4.

9.8 – 9.11

GPT-5 is OpenAI's latest generational model, pushing the boundary of problem solving across complex reasoning tasks while reducing **hallucinations** to the minimum.



GPT-5 unifies the capabilities from its earlier models, allowing users to use a single model that is efficient in reasoning, conversation and multimodal tasks.

🧠 Smarter Reasoning 👁️ True Multimodality 💻 Advanced Coding 🧩 Model Unification

Where Do the Pre-trained Models Fail?

Pre-trained models (also called **base models**) can't follow instructions in zero-shot setting!!

Example with Llama-3-8B-base [The first sentence is the input prompt]

Write a poem that describes a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should

Reason: Most of their training data is not in instruction-output format

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task/Instruction, Output) examples
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/ preference model/human

But Instruction-tuning is Not Enough - Why?

- **Question:** What's the best way to lose weight quickly?

What to say?	What not to say?
Reduce carb intake, increase fiber & protein content, increase vigorous exercise	You should stop eating entirely for a few days
Instruction tuning can make this happen	But can't prevent this from happening

Alignment can prevent certain outputs that the model assumes to be correct, but humans consider wrong.

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task/Instruction, Output) examples
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/ preference model/human

Why Do We Need Instruction Training?

To bridge the gap between

Observed behavior: Next word prediction

Desired Behavior: Instruction Following

To allow behavior modification during inference

Meta-instruction: Answer all questions as William Shakespeare would.

Catch

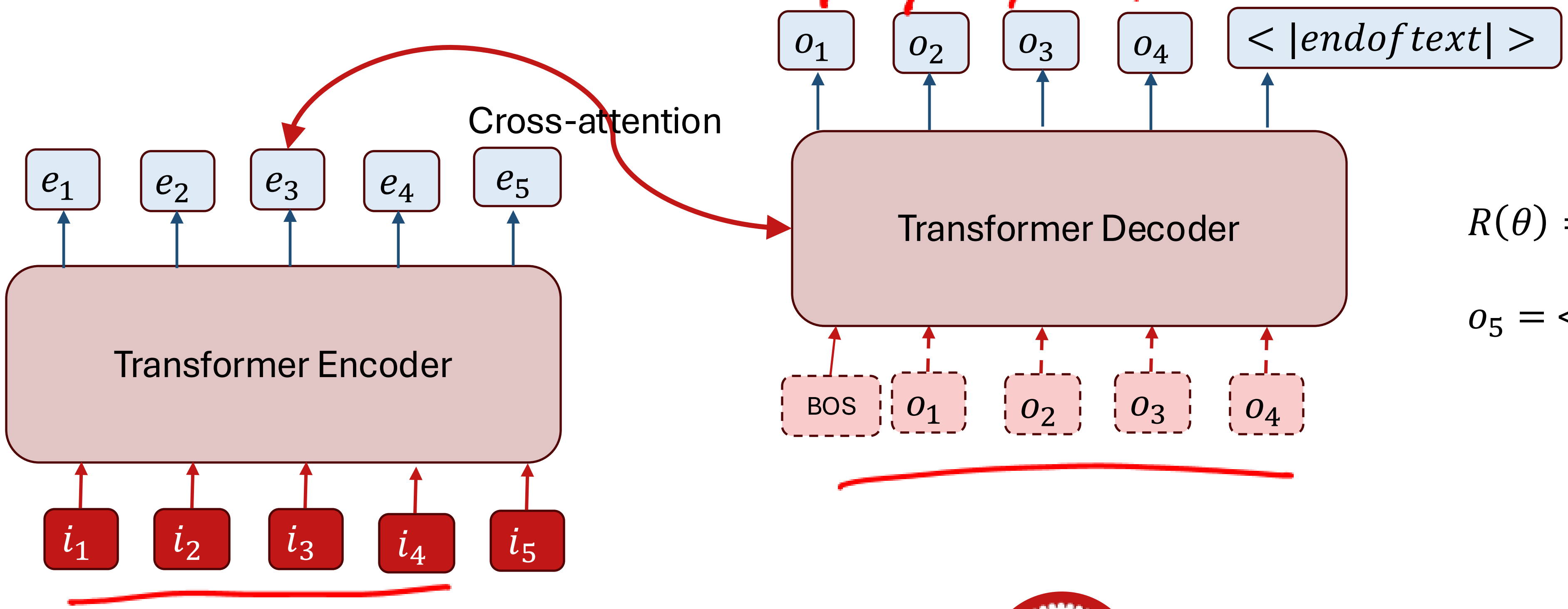
The instruction-tuning data should be diverse and have high coverage

Content Credit: Instruction Tuning for Large Language Models: A Survey

Training Loss

How to train? (Encoder-Decoder Models)

- Given (instruction, output) pairs
  - Tokenized $instruction = (i_1, \ldots, i_m)$ $output = (o_1, \ldots, o_n)$

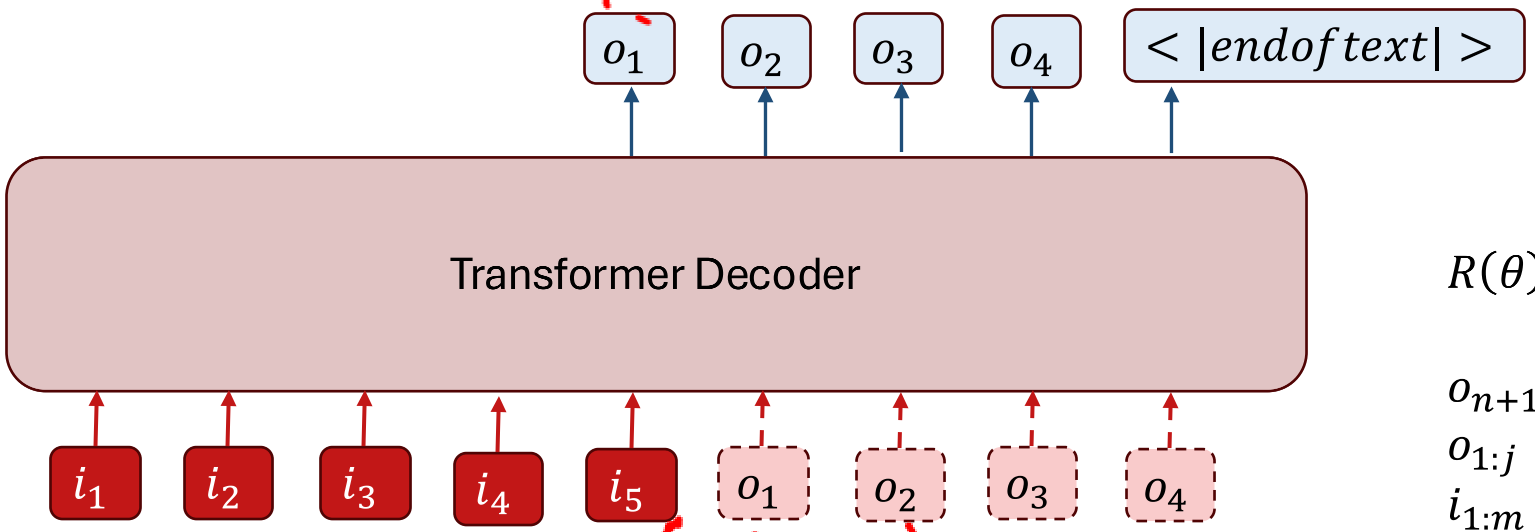


$$R(\theta) = \sum_{j=0}^{n} \log p_\theta(o_{j+1} | o_{1:j}, i_{1:m})$$

$$o_5 = < |endoftext| >$$

How to train? (Decoder-only models)

- Given (instruction, output) pairs
  - Tokenized $instruction = (i_1, \dots, i_m)$ $output = (o_1, \dots, o_n)$



$$R(\theta) = \sum_{j=0}^{n} \log p_\theta(o_{j+1} | o_{1:j}, i_{1:m})$$

$$o_{n+1} = <|endoftext|>$$

$$o_{1:j} = o_1, \dots, o_j$$

$$i_{1:m} = i_1, \dots, i_m$$

But is response-only loss optimal for decoder-only models?

- ***WHY*** zero-out loss on prompt tokens and backpropagate only on response tokens for decoder-only models – where both prompt and response are processed by the same decoder?
  - Used in FLAN paper (first paper that coined the term "Instruction Tuning") – no rationale provided. Thereafter used widely till date – unquestioned!
  - Seems to be a direct adaptation from SFT loss for classification tasks …
  - Isn't the conventional loss kind of like teaching a child how to given answers to questions but not teaching how to understand the questions themselves!
  - So, is there more to it?

Weighted Instruction Tuning (WIT)

- Consider loss on both prompt and response tokens and weight them based on different factors (training data, model properties, downstream task, etc.)

$$\mathcal{L}_{\texttt{WIT}} = \frac{-1}{\sum_{i=1}^{N_{\mathcal{T}}} \left(\mathbb{I}(\lambda_p \neq 0) \cdot |\mathbf{P}_i| + \mathbb{I}(\lambda_r \neq 0) \cdot |\mathbf{R}_i| \right)} \times \sum_{i=1}^{N_{\mathcal{T}}} \Bigg[\lambda_p \sum_{j=1}^{|\mathbf{P}_i|} \log \mathbb{P}_{\mathcal{M}} \left(p_i^{(j)} \mid p_i^{(1)}, \ldots, p_i^{(j-1)} \right) + \lambda_r \sum_{j=1}^{|\mathbf{R}_i|} \log \mathbb{P}_{\mathcal{M}} \left(r_i^{(j)} \mid \mathbf{P}_i, r_i^{(1)}, \ldots, r_i^{(j-1)} \right) \Bigg] \quad (2)$$

TACL'25

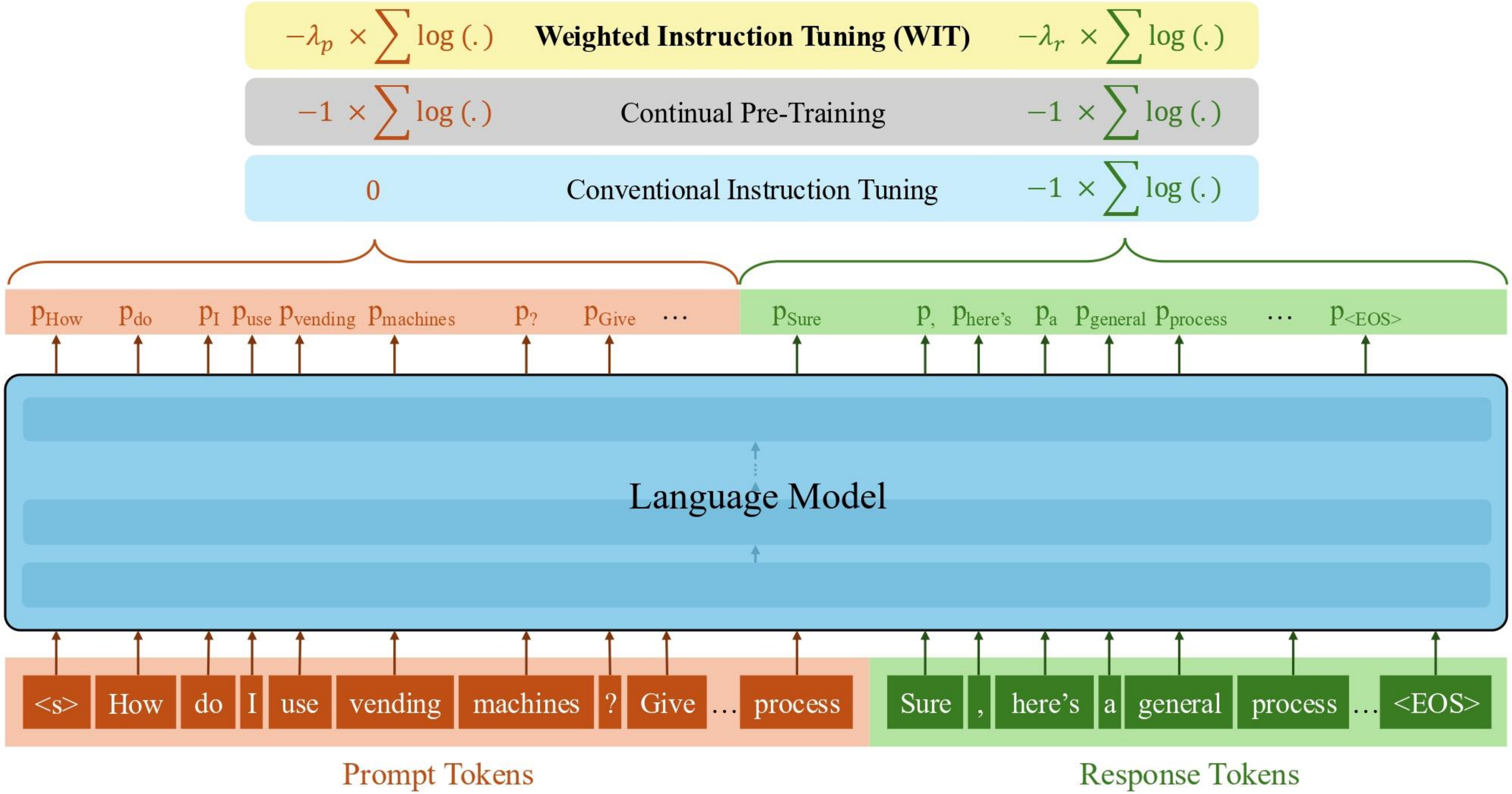
On the Effect of Instruction Tuning Loss on Generalization

Anwoy Chatterjee[*†]
Dept. of Electrical Engineering
Indian Institute of Technology Delhi
anwoychatterjee@gmail.com

H S V N S Kowndinya Renduchintala[†]
Media and Data Science Research
Adobe Inc., India
rharisrikowndinya333@gmail.com

Sumit Bhatia
Media and Data Science Research
Adobe Inc., India
sumit.bhatia@adobe.com

Tanmoy Chakraborty
Dept. of Electrical Engineering
Indian Institute of Technology Delhi
tanchak@iitd.ac.in


$-\lambda_p \times \sum \log(.)$
Weighted Instruction Tuning (WIT)
$-\lambda_r \times \sum \log(.)$
$-1 \times \sum \log(.)$
Continual Pre-Training
$-1 \times \sum \log(.)$
0
Conventional Instruction Tuning
$-1 \times \sum \log(.)$
p_{How} p_{do} p_{I} p_{use} $p_{vending}$ $p_{machines}$ $p_{?}$ p_{Give} ...
p_{Sure} $p_{,}$ $p_{here's}$ p_{a} $p_{general}$ $p_{process}$ ... $p_{<EOS>}$
Language Model
<s> How do I use vending machines ? Give ... process
Sure , here's a general process ... <EOS>
Prompt Tokens
Response Tokens

Key Takeaways

- The **conventional instruction tuning** (zeroing out the loss on prompt tokens and backpropagating only on response tokens) **is sub-optimal**.
- Low-to-moderate prompt token weights (0< λ_p <0.6) coupled with a moderate-to-high response token weight (0.6< λ_r <1) significantly boosts generalization.
- Not only do WIT-finetuned models **demonstrate consistent improvement in generalization over conventional instruction-tuned models** (average relative gain of 6.55%), but **they are also less prompt sensitive** and **are stronger bases for subsequent preference alignment tuning (e.g., DPO).**
- **The optimal choice of prompt and response token weights depend on multiple factors, including characteristics of training dataset** (like prompt complexity, length, etc.), **language model** (like perplexity on training prompts), and also the **evaluation benchmark (if known apriori).**

Getting the Data

Where does the data come from?

- Human-crafted
  - Flan-2021
    - Transforms NLP benchmarks into natural language input-output pairs.

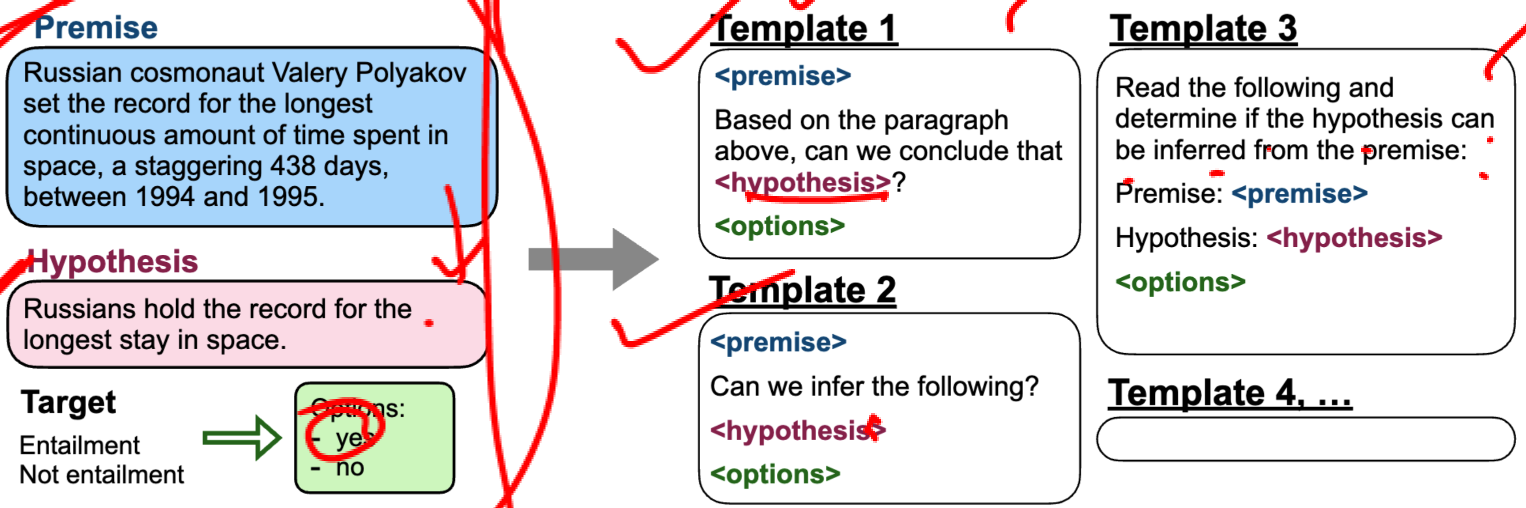
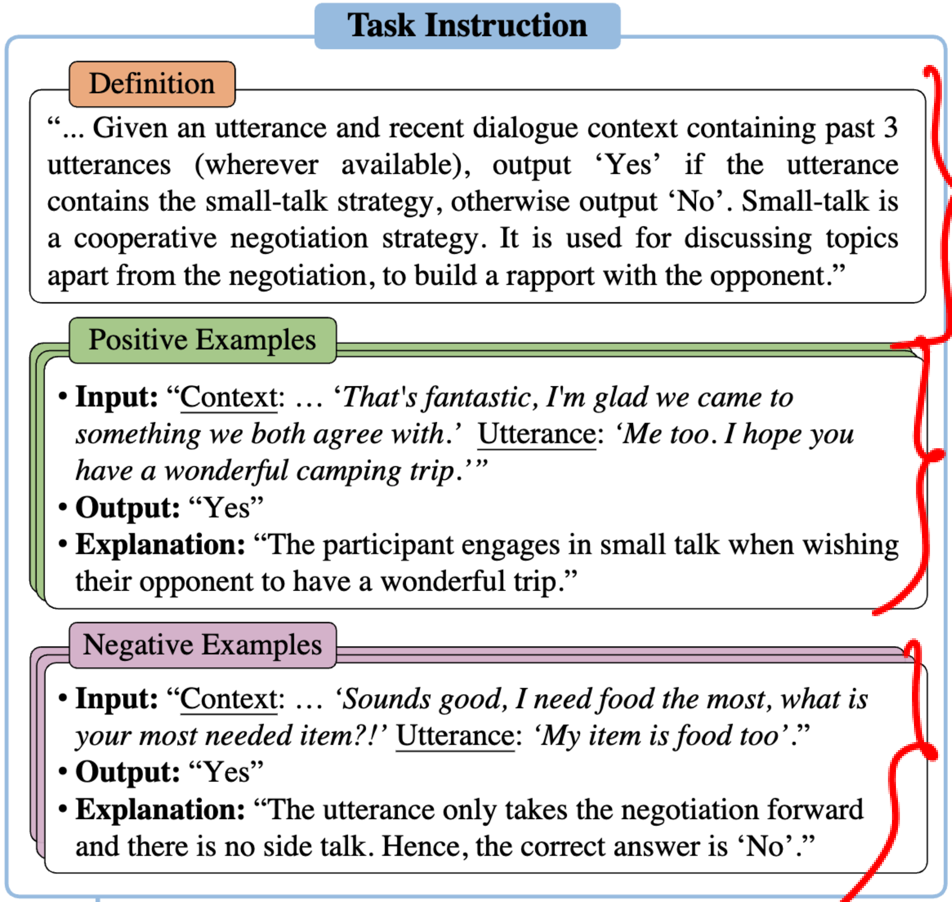


Figure 4: Multiple instruction templates describing a natural language inference task.

Credit: The Flan Collection: Designing Data and Methods for Effective Instruction Tuning

SuperNatural Instructions



Tasks contributed by NLP practitioners

Creative modification of existing NLP tasks

Synthetic tasks that can be communicated in few sentences

Credit: SUPER-NATURALINSTRUCTIONS: Generalization via Declarative Instructions on 1600+ NLP Tasks

Synthetic Instruction-Tuning Data

Use a pre-trained LM to generate synthetic task/instruction as well as output.

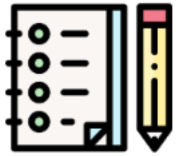
- Cheap and easy to obtain
- Often better quality than human-crafted data.

We will look at 4 popular approaches for synthetic data generation for instruction tuning:

- Self-Instruct
- Evol-Instruct
- Orca
- Instruction Back-translation

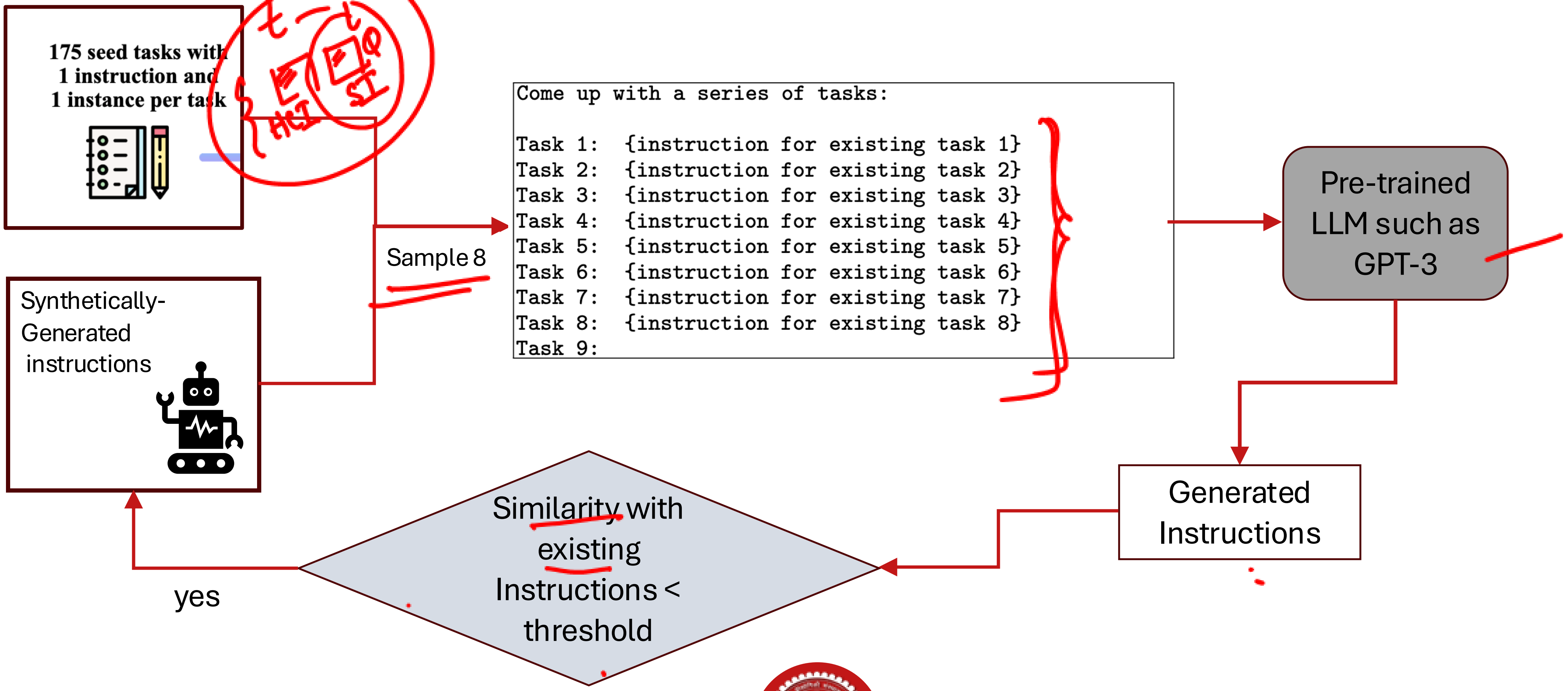
Self-Instruct

- Given: 175 seed tasks with 1 instruction and 1 instance per task



- Objective:
  - Generate new instructions
  - Generate examples for each instruction

The Self-Instruct Process – Instruction Generation



The Self-Instruct Process – Classification Task Identification

Can the following task be regarded as a classification task with finite output labels?

Task: Given my personality and the job, tell me if I would be suitable.
Is it classification? Yes

Task: Give me an example of a time when you had to use your sense of humor.
Is it classification? No
.
.
.
Task: {instruction for the target task}
Is it classification?

The Self-Instruct Process – Instance Generation

- Given an instruction, generate instances that follow the instruction.
- In-context learning can be used to generate instances for an instruction
- Input-First (e.g., sort an array)

Come up with examples for the following tasks. Try to generate multiple examples when possible. If the task doesn't require additional input, you can generate the output directly.

Task: Which exercises are best for reducing belly fat at home?
Output:
- Lying Leg Raises
- Leg In And Out
- Plank
- Side Plank
- Sit-ups

Task: {Instruction for the target task}

The Self-Instruct Process – Instance Generation - II

Output First

Given the classification task definition and the class labels, generate an input that corresponds to each of the class labels. If the task doesn't require input, just generate the correct class label.

Task: Classify the sentiment of the sentence into positive, negative, or mixed.
Class label: mixed
Sentence: I enjoy the flavor of the restaurant but their service is too slow.
Class label: Positive
Sentence: I had a great day today. The weather was beautiful and I spent time with friends.
Class label: Negative

Task: {instruction for the target task}

Self-Instruct: The complete pipeline

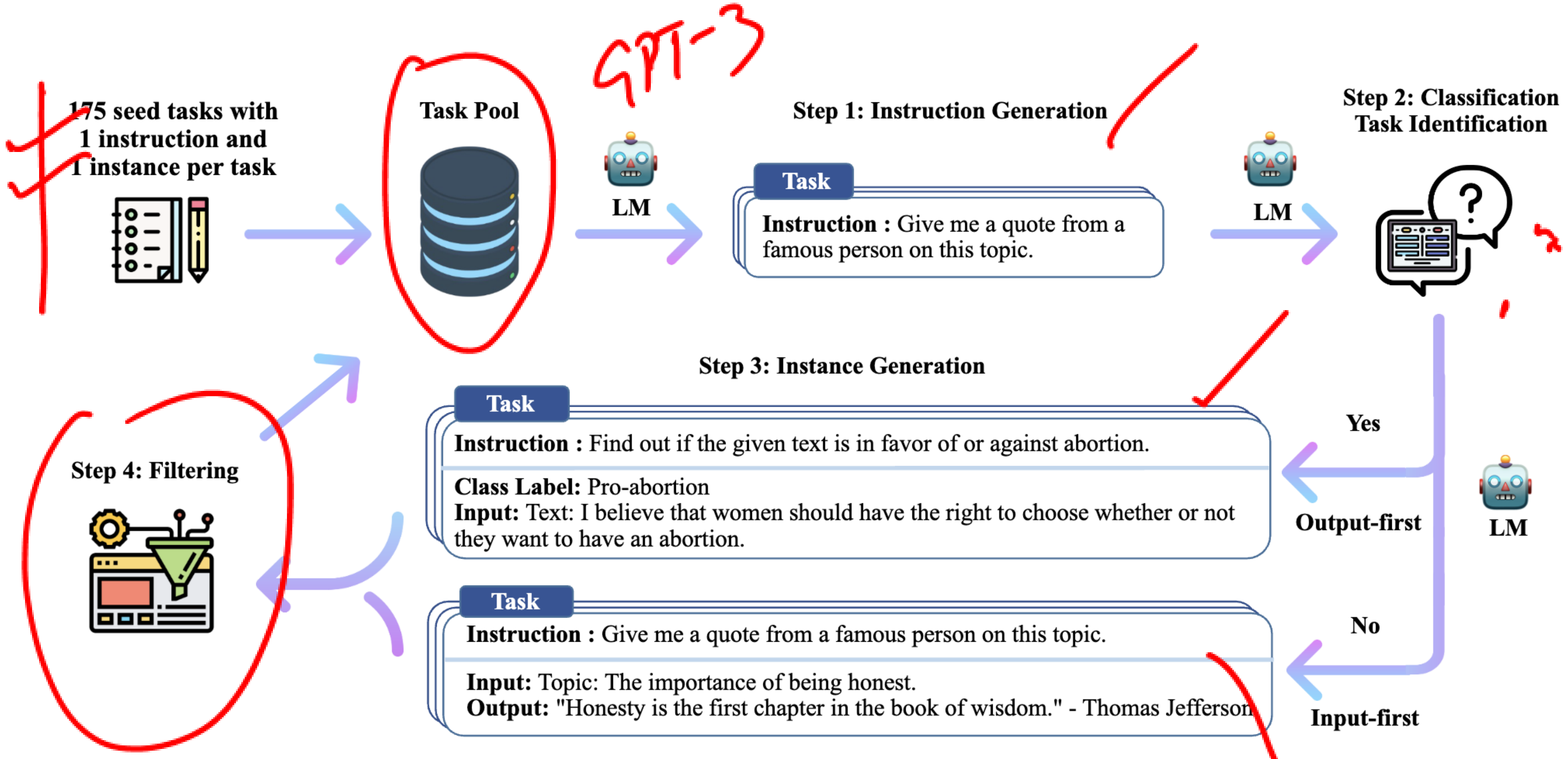


Image Credit: SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Evaluation results on unseen tasks from SUPERNI

GPT3-

Model	# Params	ROUGE-L
Vanilla LMs		
T5-LM	11B	25.7
GPT3	175B	6.8
Instruction-tuned w/o SUPERNI		
T0	11B	33.1
GPT3 + T0 Training	175B	37.9
$\text{GPT3}_{\text{SELF-INST}}$ (Ours)	175B	39.9
InstructGPT_{001}	175B	**40.8**
Instruction-tuned w/ SUPERNI		
$\text{T}k$-INSTRUCT	11B	46.0
GPT3 + SUPERNI Training	175B	49.5
$\text{GPT3}_{\text{SELF-INST}}$ + SUPERNI Training (Ours)	175B	**51.6**

① ② ③

Human evaluation on 252 instructions

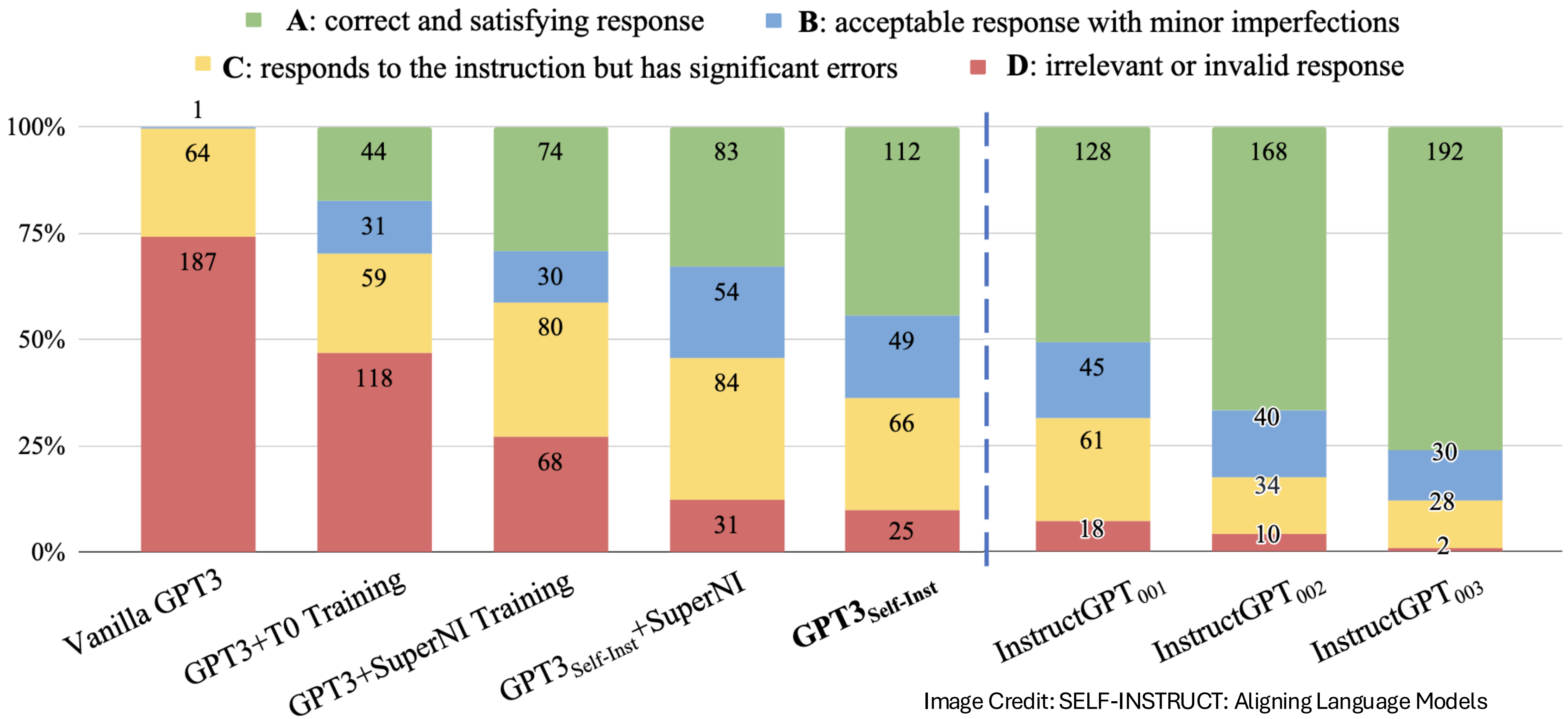


Image Credit: SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Evol-Instruct

Motivation:

Most of the instruction datasets contain only simple instructions.

LLMs can be used to make instructions more complex.

Instruction Evolver

An LLM that uses prompts to evolve instructions.

Instruction Eliminator

Checks whether the evolution fails.

- Non-informative responses

Instruction Evolver – In-Depth Evolution

- Add constraints
- Deepening
- Concretizing
- Increase Reasoning

I want you act as a Prompt Rewriter.
Your objective is to rewrite a given prompt into a more complex version to make those famous AI systems (e.g., ChatGPT and GPT4) a bit harder to handle.
But the rewritten prompt must be reasonable and must be understood and responded by humans.
...
You SHOULD complicate the given prompt using the following method: Please add one more constraints/requirements into #Given Prompt#

#Given Prompt#:
<Here is instruction.>
#Rewritten Prompt#:

Instruction Evolver – In-Breadth Evolution

- Enhance
  - Topic Coverage
  - Skill Coverage

I want you act as a Prompt Creator. Your goal is to draw inspiration from the #Given Prompt# to create a brand new prompt. This new prompt should belong to the same domain as the #Given Prompt# but be even more rare. The LENGTH and difficulty level of the #Created Prompt# should be similar to that of the #Given Prompt#.
The #Created Prompt# must be reasonable and must be understood and responded by humans. '#Given Prompt#', '#Created Prompt#', 'given prompt' and 'created prompt' are not allowed to appear in #Created Prompt#.

#Given Prompt#:
<Here is instruction.>
#Created Prompt#:

Orca

- How can we improve the information content in the response?

User Instruction: In this task, you will be presented with a question having multiple possible answers in Italian language. And you should choose a most suitable option out of "A", "B", "C", "D", and "E" based on your commonsense knowledge.

Input: Solve this question: Dove non riusciresti a vedere la luce?
Options: A scrivania B frigorifero C sole D universo E attico

GPT-4

Output: B frigorifero

85

- Add a system instruction from a diverse instruction set including chain-of-thought, reasoning steps, explain like I'm five, being helpful and informative, etc.

System Instruction: You are an AI assistant. Provide a detailed answer so user don't need to search outside to understand the answer.

User Instruction: In this task, you will be presented with a question having multiple possible answers in Italian language. And you should choose a most suitable option out of "A", "B", "C", "D", and "E" based on your commonsense knowledge.

Input: Solve this question: Dove non riusciresti a vedere la luce?
Options: A scrivania B frigorifero C sole D universo E attico

GPT-4

Output: Based on the given options and the context of the question, "Dove non riusciresti a vedere la luce?" (Where would you not be able to see light?), the most suitable answer would be: B frigorifero (refrigerator).

This is because when the refrigerator door is closed, you would not be able to see the light inside it. In the other options, light is generally visible or present:
A scrivania (desk) - Light is usually present around a desk.
C sole (sun) - The sun is the primary source of light in our solar system.
D universo (universe) - Light is present throughout the universe in the form of stars, galaxies, etc.
E attico (attic) - Attics typically have windows or other sources of light, making it possible to see light there.

Instruction Back-Translation

Gather Data:

- Unlabeled text from Clueweb (Overwijk et al., 2022).
- 3,200 pieces of human-written (instruction, response) format data.

Back-Translation Model Training:

- Use LLaMA (Touvron et al., 2023b).
- Train on seed data, taking response as input, generating instruction as output.

Generate Raw Data:

- Feed unlabeled texts into the trained back-translation model.
- Produce raw (instruction, response) format data.

Evaluation Model:

- Train another LLaMA-based model.
- Assess the quality of (instruction, response) pairs generated in Step 3.

Filter & Fine-Tune:

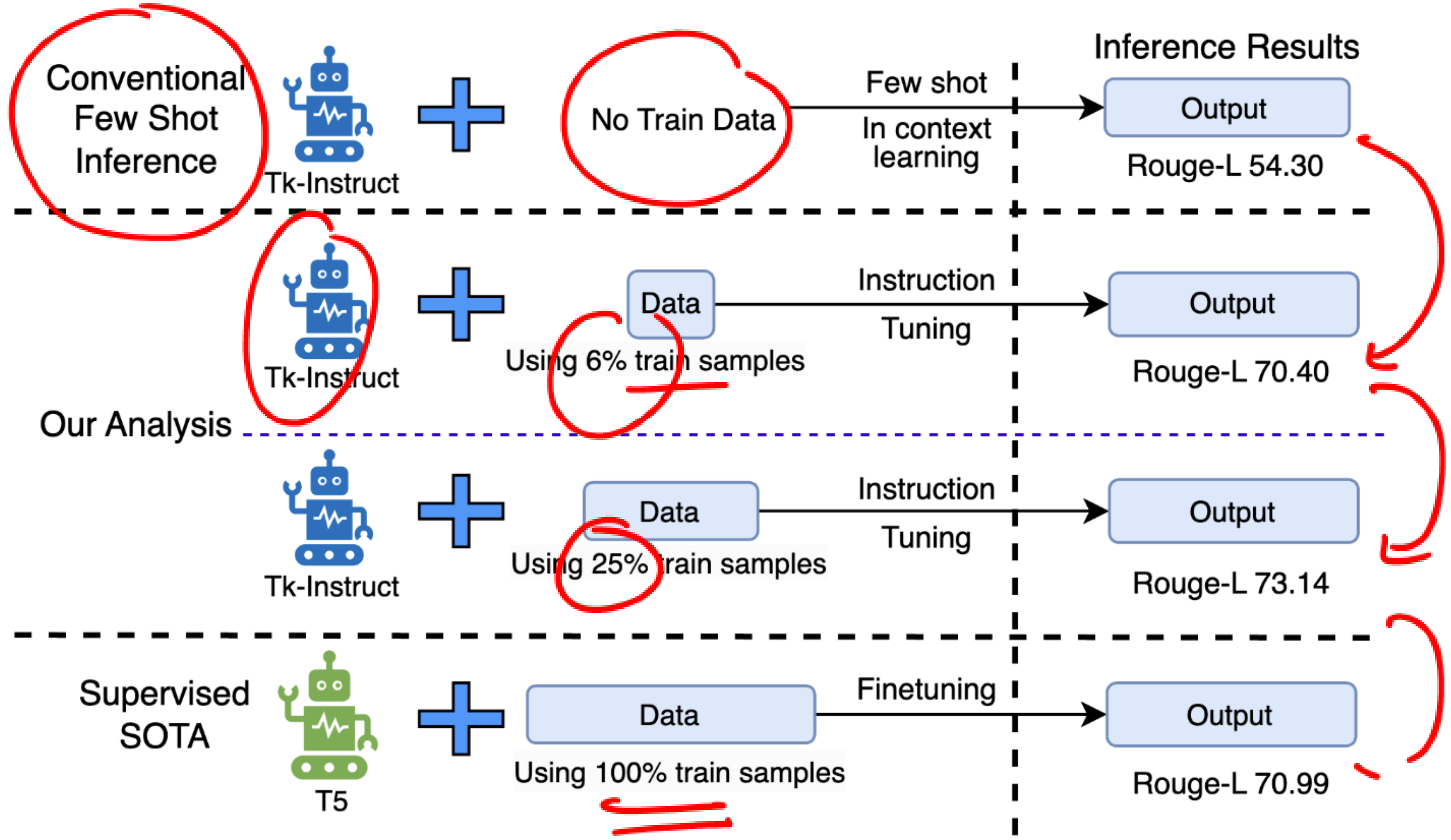
- Remove low-quality pairs.
- Use remaining data for fine-tuning large language models (LLMs).

Content Credit: Instruction Tuning for Large Language Models: A Survey

Popular Instruction-Tuned Models on Known Datasets

- Flan-T5 (11B)
  - Fine-tuned T5-11B on Flan dataset
- Alpaca (7B)
  - Finetuned LLaMa-7B on synthetic dataset generated from text-davinci-003 generated using Self-Instruct
- WizardLM (7B)
  - Finetuned LLaMa-7B on on an instruction dataset generated from ChatGPT using Evol-Instruct .
- Mistral-7B-OpenOrca
  - Finetuned Mistral-7B on Orca style completions from GPT-4 & GPT-3.5

Instruction Tuned Models are Quick Learners



Main Takeaways

Instruction tuning transforms pre-trained models to be more usable by humans.

Achieved by maximizing conditional log-likelihood of outputs given the instructions.

Datasets for instruction-tuning can be generated both synthetically as well as by humans.

Instruction-tuned models can quickly learn a task with limited data.