

TRAINING VÀ FINETUNING LANGUAGE MODEL VỚI GPT-2

Nhóm 2

- Trần Xuân Bảo - 23020332
- Hà Xuân Huy - 23020375
- Phan Hoàng Dũng - 23020346

CÁC PHẦN CHÍNH

01 GIỚI THIỆU

02 CƠ SỞ LÝ THUYẾT

03 XÂY DỰNG THÍ NGHIỆM

04 KẾT QUẢ

05 HƯỚNG PHÁT TRIỂN

06 KẾT LUẬN

GIỚI THIỆU

Intro

GIỚI THIỆU

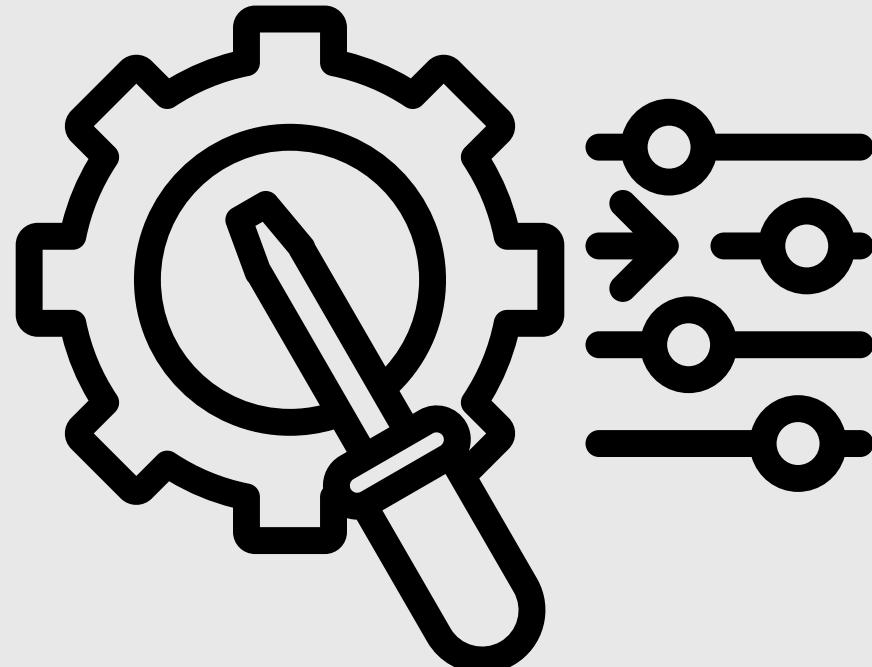
Sự phát triển của mô hình ngôn ngữ lớn (LLM)

- Mô hình như GPT-2 có khả năng sinh văn bản mượt mà, hiểu ngữ cảnh tốt, hỗ trợ nhiều ứng dụng NLP:
 - Trả lời câu hỏi
 - Tóm tắt văn bản
 - Viết sáng tạo, dịch tự động, v.v.
- Tuy nhiên, GPT-2 gốc được huấn luyện trên dữ liệu tổng quát, không chuyên biệt cho các lĩnh vực cụ thể như pháp luật.



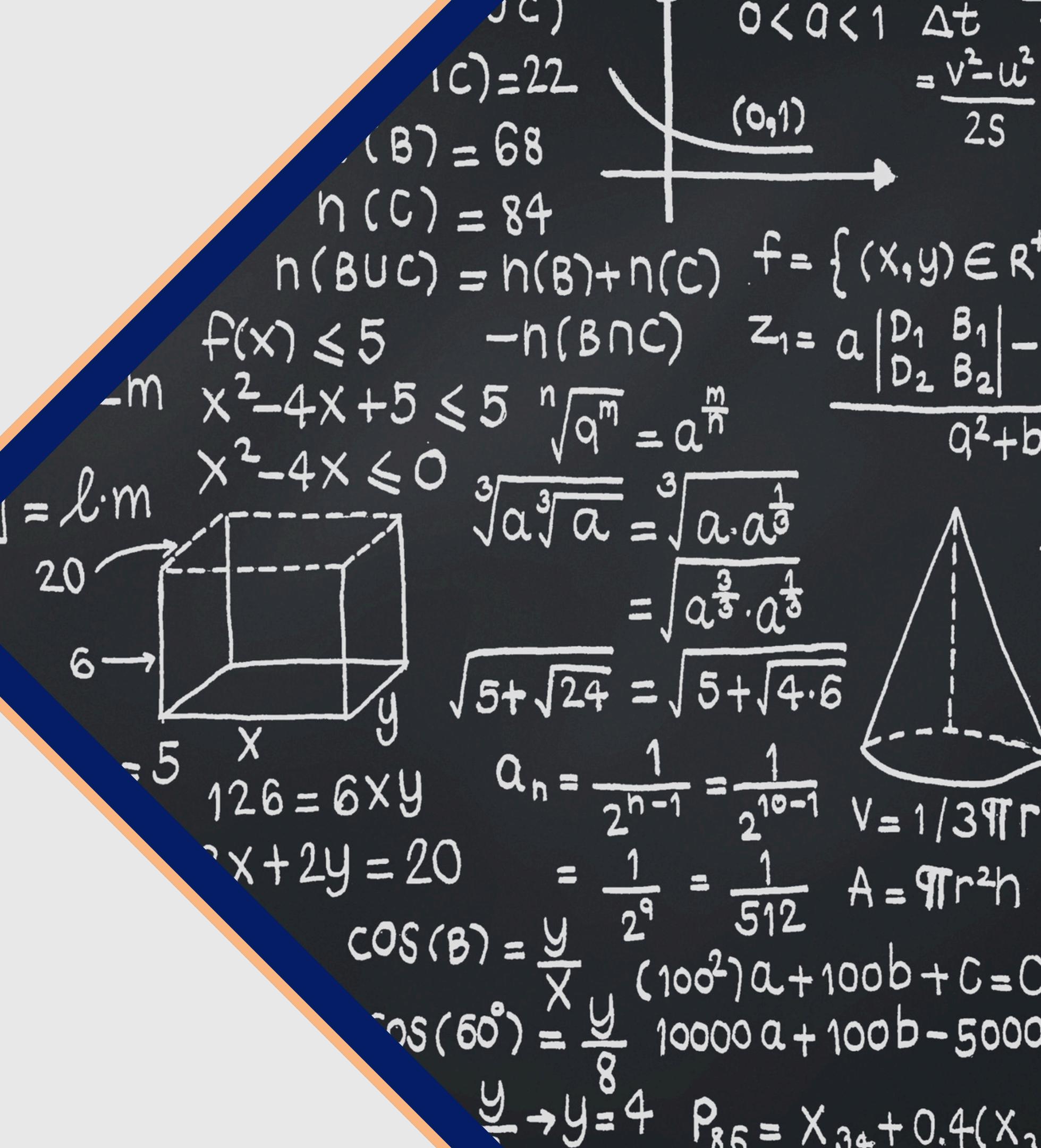
GIỚI THIỆU

- Những giới hạn khi dùng GPT-2 trực tiếp
 - Thiếu kiến thức chuyên ngành pháp lý
 - Không hiểu ngữ cảnh địa phương
 - Dễ bị lan man hoặc đưa ra thông tin không chính xác



- Cân chỉnh chỉnh (fine-tuning) để khắc phục
 - Cập nhật mô hình với dữ liệu hỏi đáp pháp lý thực tế
 - Giúp mô hình hiểu cách diễn đạt, logic và cấu trúc câu trả lời đúng luật

CƠ SỞ LÝ THUYẾT



CƠ SỞ LÝ THUYẾT

Bài báo “Improving Language Understanding by Generative Pre-Training”

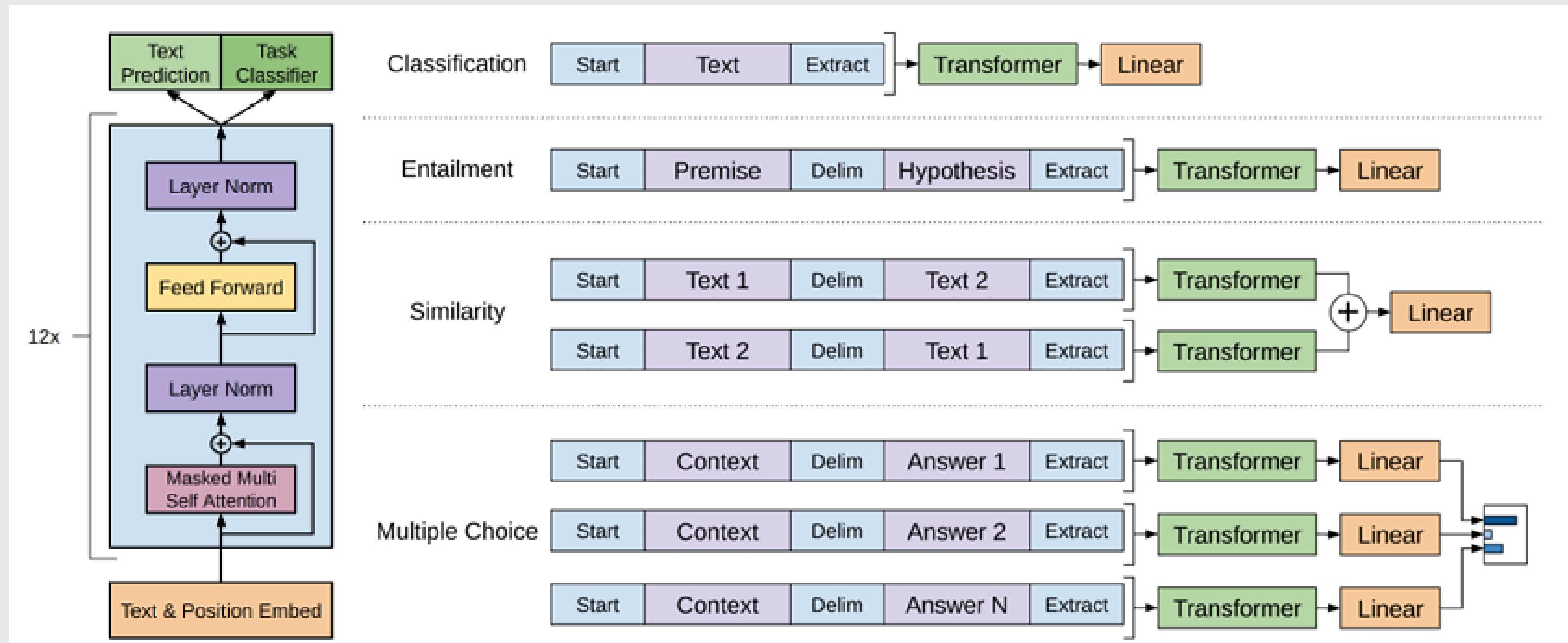
- GPT: Generative Pretraining Transformer
 - GPT là mô hình đầu tiên trong dòng GPT của OpenAI
 - Kết hợp huấn luyện không giám sát với fine-tuning có giám sát
 - Không cần thay đổi nhiều về kiến trúc giữa các tác vụ
- Mục tiêu chính
 - Tạo một mô hình ngôn ngữ tổng quát có thể:
 - Học từ dữ liệu không gán nhãn
 - Giải quyết các bài toán như NLI (Natural language inference), QA (Question Answering), Sentence similarity, classification với cùng một mô hình.
- Các task và dataset sử dụng

Table 1: A list of the different tasks and datasets used in our experiments.

Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

CƠ SỞ LÝ THUYẾT

Chuyển đổi đầu vào cho các tác vụ khác nhau



Minh họa kiến trúc Transformer (trái) và cách chuyển đổi đầu vào cho các tác vụ fine-tuning khác nhau (phải), bằng cách biến mọi dạng input có cấu trúc thành chuỗi token để xử lý thống nhất bởi mô hình tiền huấn luyện.

CƠ SỞ LÝ THUYẾT

Quy trình huấn luyện 2 giai đoạn

- Giai đoạn 1 - Pre-training (Unsupervised)

- Nhiệm vụ: Huấn luyện mô hình ngôn ngữ để dự đoán từ tiếp theo dựa vào ngữ cảnh trước đó trong chuỗi token không gán nhãn.
- Dữ liệu đầu vào: Tập văn bản chưa gán nhãn: $U=\{u_1, u_2, \dots, u_n\}$
- Hàm mục tiêu

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

- Với k : kích thước cửa sổ ngữ cảnh
- Θ : tham số của mô hình, được tối ưu bằng SGD

CƠ SỞ LÝ THUYẾT

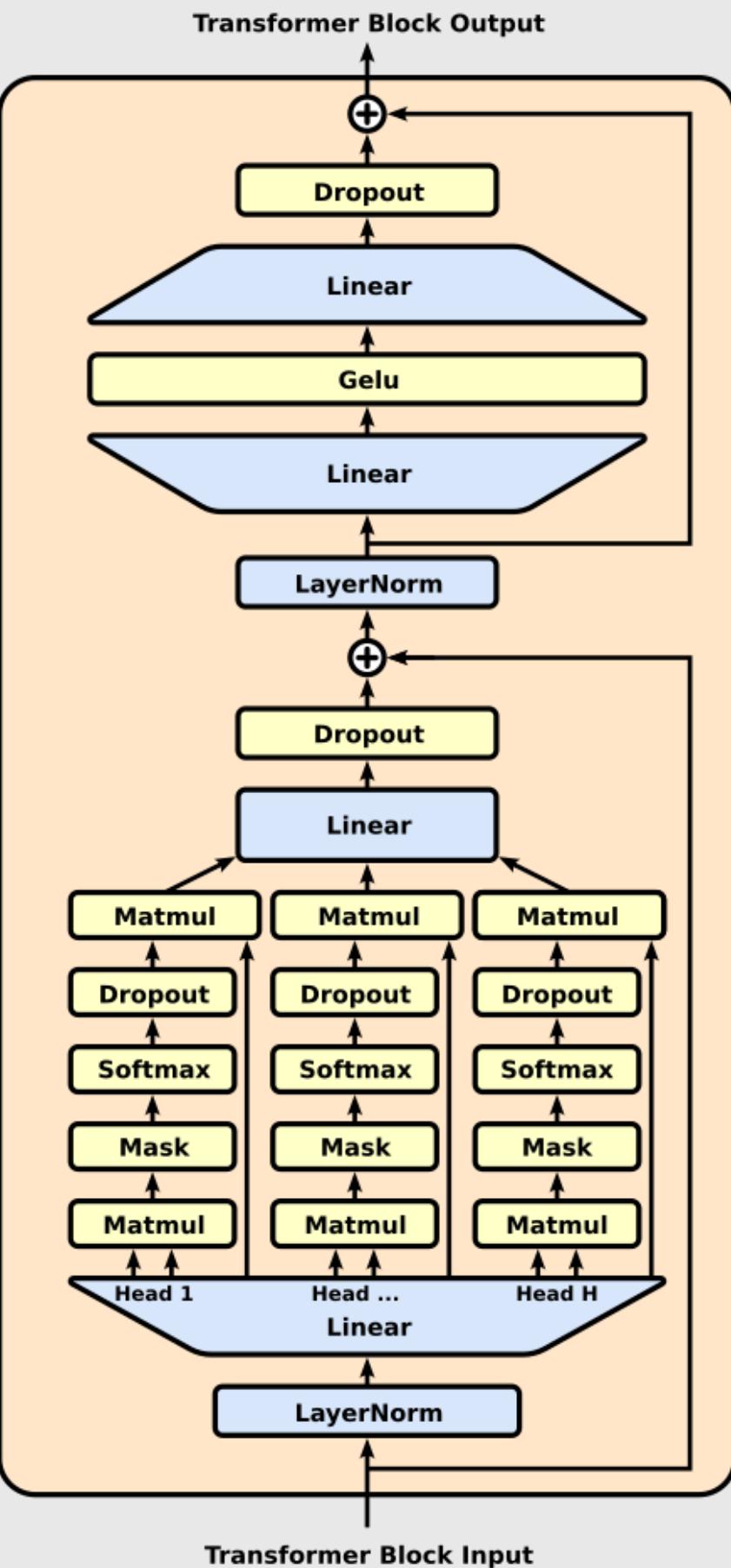
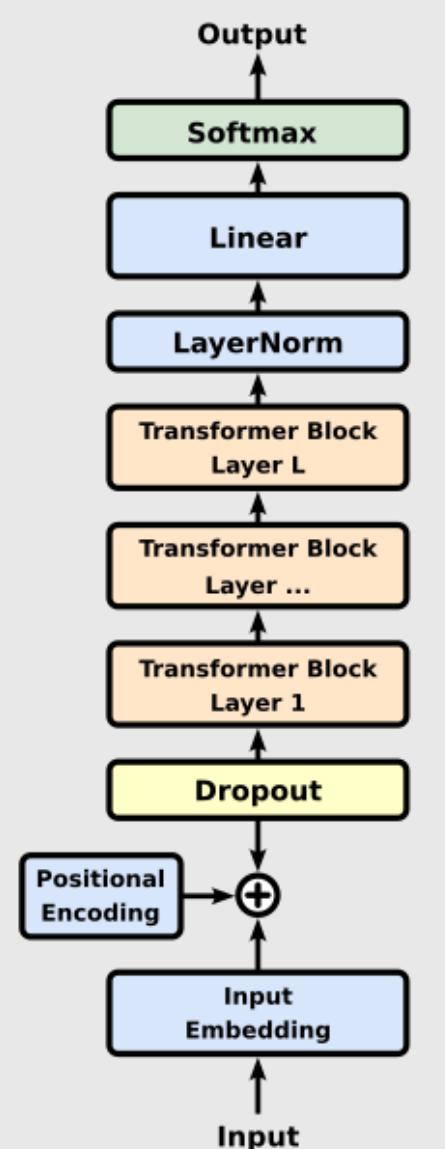
Quy trình huấn luyện 2 giai đoạn

- Giai đoạn 2 - Supervised Fine-tuning
 - Nhiệm vụ: Tinh chỉnh mô hình đã pre-train trên một tập dữ liệu có gán nhãn để thực hiện các tác vụ cụ thể
 - Dữ liệu: Tập dữ liệu có nhãn: $C=\{(x,y)\}$
 - Đầu ra tầng cuối $h^m(m_l)$ được đưa qua lớp tuyến tính W_y : $P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y)$.
 - Hàm mất mát chính:
$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m).$$
 - Auxiliary Objective:
 - Kết hợp thêm mục tiêu language modeling để cải thiện khả năng tổng quát ,tăng tốc độ hội tụ

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

CƠ SỞ LÝ THUYẾT

- Kiến trúc mô hình GPT-1
 - Transformer Decoder (12 layers)
 - Hidden size: 768
 - Attention heads: 12
 - Dropout: 0.1
 - Regularization: L2 ($\lambda = 0.01$)
 - Vocabulary: BPE 40k tokens
 - Optimizer: Adam + Learning Rate warmup + Cosine decay



CƠ SỞ LÝ THUYẾT

Kết quả huấn luyện

- Kết quả thực nghiệm trên các tác vụ suy diễn ngôn ngữ tự nhiên (natural language inference)
- Kết quả trên các tác vụ trả lời câu hỏi và lý luận dựa trên kiến thức thông thường (commonsense reasoning)

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	<u>89.3</u>	-	-	-
CAFE [58] (5x)	80.2	79.0	<u>89.3</u>	-	-	-
Stochastic Answer Network [35] (3x)	<u>80.6</u>	<u>80.1</u>	-	-	-	-
CAFE [58]	78.7	77.9	88.5	<u>83.3</u>	-	-
GenSen [64]	71.4	71.3	-	-	<u>82.3</u>	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	<u>82.1</u>	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	<u>77.6</u>	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	<u>60.2</u>	<u>50.3</u>	<u>53.3</u>
Finetuned Transformer LM (ours)	86.5	62.9	57.4	59.0

CƠ SỞ LÝ THUYẾT

Kết quả huấn luyện

- Kết quả về độ tương đồng ngữ nghĩa và phân loại
- Tất cả các tác vụ được đánh giá bằng bộ chuẩn GLUE benchmark.

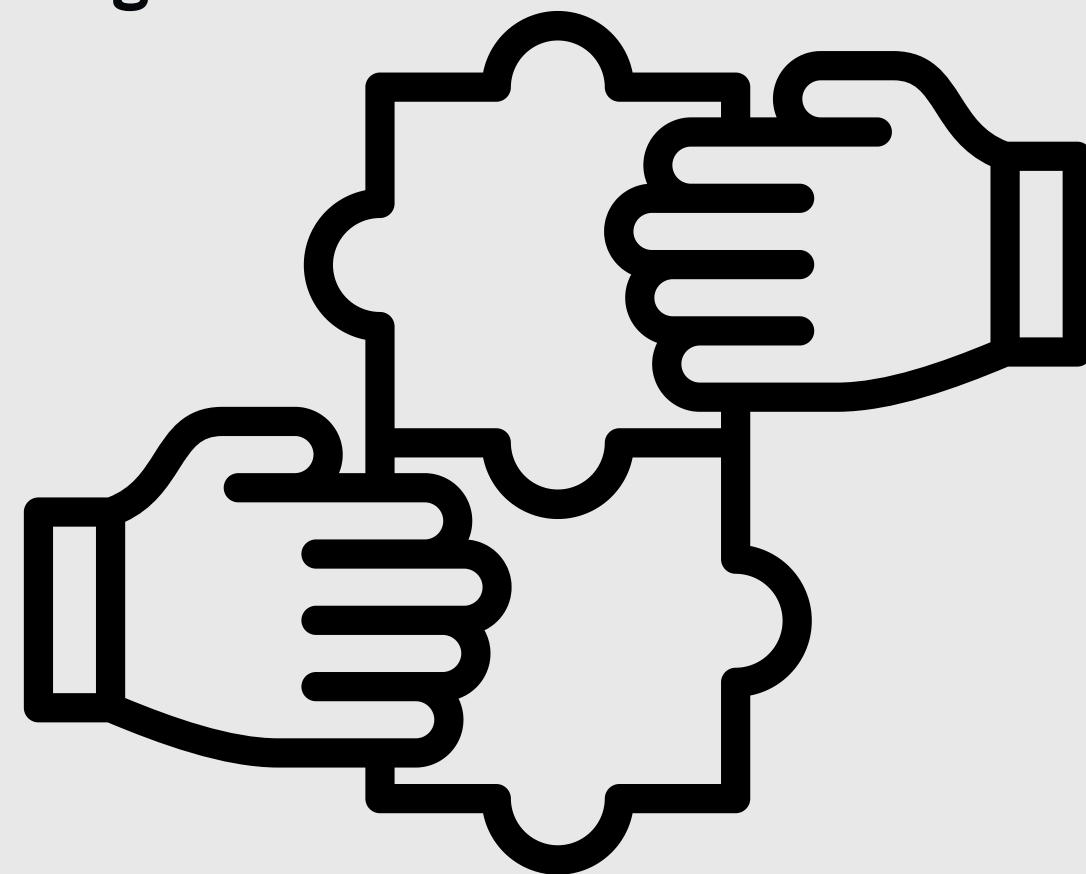
- Phân tích các biến thể mô hình trên nhiều tác vụ.
- Avg. score là trung bình không trọng số của tất cả các kết quả.

Method	Classification		Semantic Similarity		GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STS-B (pc)	
Sparse byte mLSTM [16]	-	93.2	-	-	-
TF-KLD [23]	-	-	86.0	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3
Finetuned Transformer LM (ours)	45.4	91.3	82.3	82.0	70.3
					72.8

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STS-B (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	70.3	81.8	88.1	56.0
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	75.0	47.9	92.0	84.9	83.2	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

CƠ SỞ LÝ THUYẾT

- Đóng góp chính
 - Đề xuất phương pháp unsupervised pre-training + supervised fine-tuning
 - Kiến trúc đơn giản, dễ chuyển đổi giữa các tác vụ
 - Thông nhất cách biểu diễn đầu vào cho đa nhiệm NLP
- Tác động lâu dài
 - Đặt nền móng cho GPT-2, GPT-3, v.v.
 - Thiết lập paradigm "pre-train → fine-tune"
 - Chứng minh hiệu quả của dữ liệu không gán nhãn



CƠ SỞ LÝ THUYẾT

Bài báo “Language Models are Unsupervised Multitask Learners”

- Mục tiêu chính:
 - Chứng minh language model không giám sát có thể thực hiện zero-shot learning trên nhiều tác vụ NLP.
 - Huấn luyện GPT-2 - mô hình Transformer lớn - trên dữ liệu văn bản WebText.
- Phương pháp:
 - Kiến trúc: Transformer Decoder (1 chiều, giống GPT-1).
 - Huấn luyện bằng next-token prediction (dự đoán token kế tiếp).
 - Không fine-tune, chỉ đánh giá bằng prompt thiết kế sẵn cho từng task.

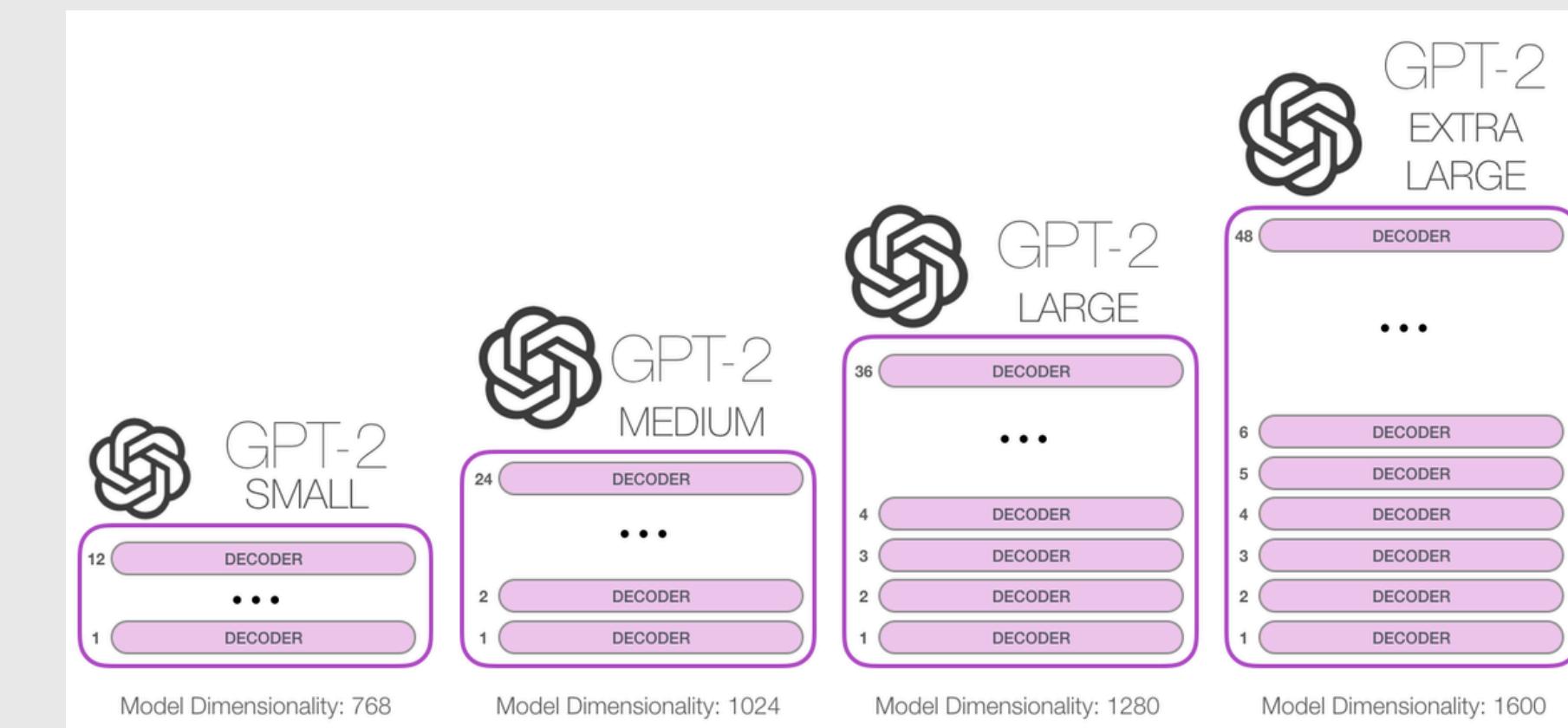


CƠ SỞ LÝ THUYẾT

Dữ liệu & Kiến trúc GPT-2

- Dữ liệu WebText:
 - Thu thập từ 45 triệu URL Reddit (≥ 3 upvotes).
 - Sau lọc còn ~8 triệu văn bản (40GB).
 - Không chứa Wikipedia để tránh rò rỉ tập test.
- Tokenization:
 - Sử dụng Byte Pair Encoding (BPE) ở mức byte → hỗ trợ mọi chuỗi Unicode.
- Kiến trúc GPT-2

Mô hình	Lớp	Hidden size	Attention heads	Số tham số
GPT-2 Small	12	768	12	117M
GPT-2 Medium	24	1024	16	345M
GPT-2 Large	36	1280	20	762M
GPT-2 XL	48	1600	25	1542M



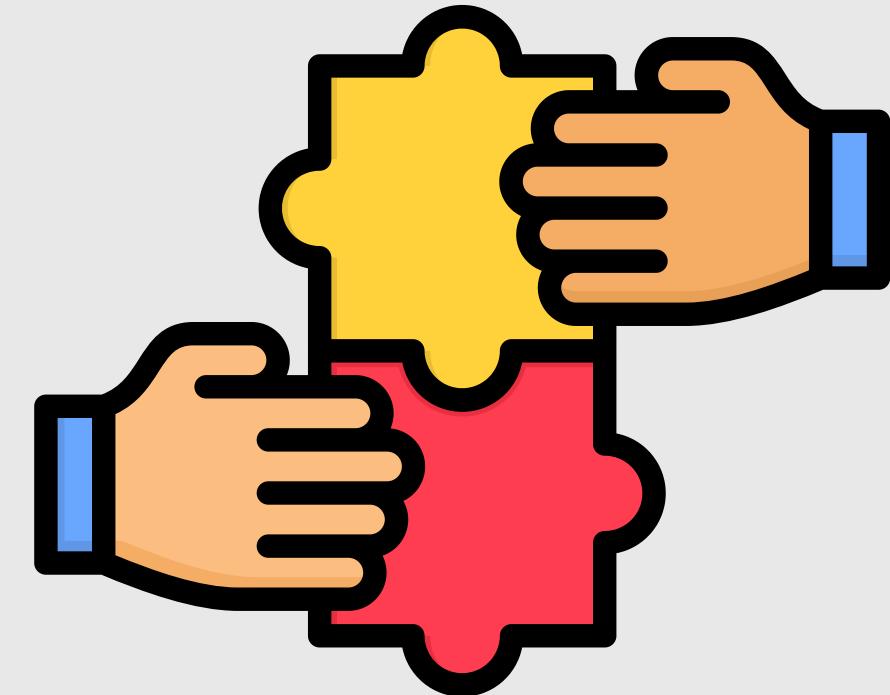
CƠ SỞ LÝ THUYẾT

Kết quả huấn luyện nổi bật trên các tác vụ khác nhau của GPT-2

Tác vụ	Hiệu suất nổi bật
Language Modeling	State-of-the-art trên 7/8 tập benchmark
Children's Book Test (CBT)	93.3% chính xác cho danh từ, gần bằng con người
LAMBADA (dài hạn)	Cải thiện perplexity từ 99.8 → 8.6 và nâng độ chính xác trên bài test này từ 19% lên 52.66%
Commonsense (Winograd)	Tăng accuracy từ 63% → 70.7%
Reading Comprehension (CoQA)	55 F1 - ngang ngửa hệ thống có huấn luyện trên 127K mẫu
Summarization	Có thể tóm tắt với prompt như TL;DR: nhưng còn thua baseline
Translation (Fr-En)	BLEU ~11.5 không huấn luyện song ngữ
Q&A (Natural Questions)	4.1% EM (so với random baseline ~1%)

CƠ SỞ LÝ THUYẾT

- Đóng góp chính của bài báo
 - Cho thấy mô hình ngôn ngữ đủ lớn có thể tự học các tác vụ phức tạp mà không cần huấn luyện đặc thù.
 - Khả năng zero-shot transfer tăng mạnh theo kích thước mô hình (hiệu ứng log-linear).
 - Gợi ý rằng tăng dung lượng mô hình + dữ liệu đủ đa dạng có thể dẫn tới hệ thống tổng quát hoá tốt mà không cần gán nhãn cụ thể.



CƠ SỞ LÝ THUYẾT

Bài báo “Training language models to follow instructions with human feedback” - 2022

- Vấn đề: Các mô hình ngôn ngữ lớn như GPT-3 không tự động hiểu hoặc tuân theo ý định người dùng, có thể tạo ra nội dung sai lệch, độc hại hoặc không hữu ích.
- Giải pháp: Đề xuất InstructGPT - mô hình GPT-3 được fine-tune bằng phản hồi từ con người (human feedback) để làm theo chỉ dẫn tốt hơn.

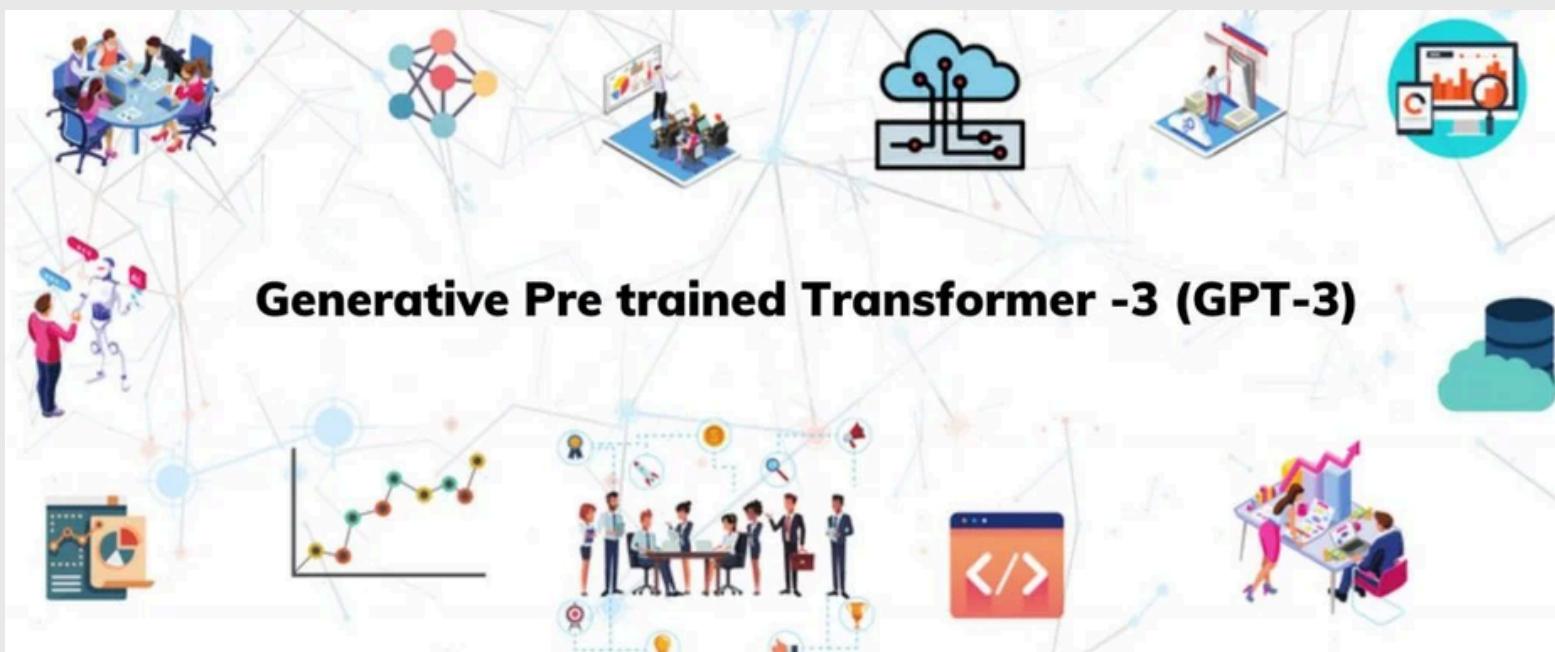


CƠ SỞ LÝ THUYẾT

Các bước chính của RLHF

1. Supervised Fine-Tuning (SFT): Huấn luyện GPT từ mô hình gốc để nó bắt đầu biết làm theo chỉ dẫn người dùng.

- Prompt từ API hoặc do labelers viết.
- Labelers cung cấp câu trả lời lý tưởng cho từng prompt.
- Khởi đầu từ GPT-3 đã pretrain.
- Huấn luyện lại bằng CrossEntropy loss trên tập prompt-completion.



Step 1

Collect demonstration data,
and train a supervised policy.

A prompt is
sampled from our
prompt dataset.



Explain the moon
landing to a 6 year old

A labeler
demonstrates the
desired output
behavior.



Some people went
to the moon...

This data is used
to fine-tune GPT-3
with supervised
learning.



SFT
Some people went
to the moon...

CƠ SỞ LÝ THUYẾT

Các bước chính của RLHF

2. Reward Model Training (RM): Xây dựng một mô hình đánh giá (reward model) để xếp hạng đâu ra tốt/xấu theo con người.

- Với mỗi prompt sinh ra K (4–9) đáp án khác nhau từ mô hình SFT.
- Con người xếp hạng: đáp án nào tốt nhất, tệ nhất.
- Tạo các cặp so sánh (e.g. (resp_1, resp_2)).
- Dùng loss hàm sigmoid + cross-entropy.
- Mỗi prompt cho $K \rightarrow K(K-1)/2$ cặp để huấn luyện hiệu quả.

Step 2

Collect comparison data, and train a reward model.

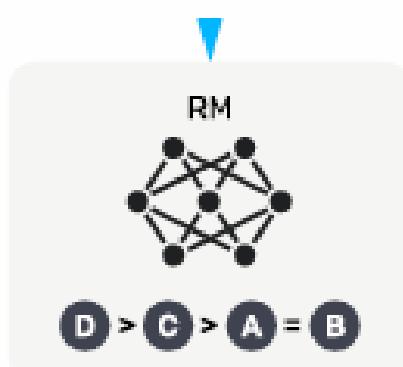
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



CƠ SỞ LÝ THUYẾT

Các bước chính của RLHF

3. Reinforcement Learning (PPO): Sử dụng PPO (Proximal Policy Optimization) để tinh chỉnh mô hình GPT sao cho nó tối đa hóa reward do RM đánh giá.

- Prompt từ người dùng (PPO dataset).
- Reward được tính từ RM.
- Policy model sinh output: GPT-3 đã qua SFT.
- Đưa prompt và output qua reward model để lấy điểm. Dùng PPO để cập nhật policy nhằm tối ưu hóa reward: Thêm penalty KL divergence để giữ hành vi gần với mô hình SFT.
- (Tùy chọn) Mix với pretraining loss để giảm mất mát hiệu suất trên task truyền thống (PPO-ptx).

Step 3

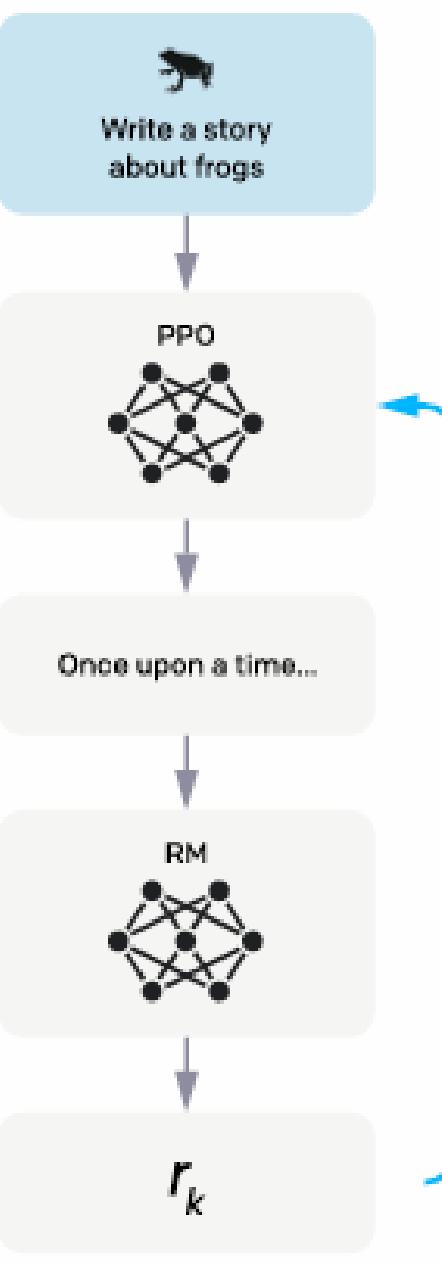
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



CƠ SỞ LÝ THUYẾT

Kết quả:

- InstructGPT (1.3B) được đánh giá tốt hơn GPT-3 (175B) trong 85% trường hợp dù nhỏ hơn 100 lần về kích thước. Các đánh giá được thực hiện bằng cả con người và tự động.
- Cải thiện đáng kể về tính đúng đắn.



CƠ SỞ LÝ THUYẾT

Bài báo “LEARNING DYNAMICS OF LLM FINETUNING” - 2025

- Mục tiêu: Hiểu rõ cách các ví dụ huấn luyện ảnh hưởng đến dự đoán của mô hình thông qua khái niệm learning dynamics - tức là quá trình ảnh hưởng giữa các ví dụ huấn luyện và đầu ra mô hình.
- Đóng góp chính:
 - Xây dựng một framework phân tích động lực học học tập trong quá trình fine-tune các mô hình ngôn ngữ lớn (LLM).
 - Áp dụng framework này để giải thích nhiều hiện tượng trong instruction tuning và preference tuning.

CƠ SỞ LÝ THUYẾT

Kỹ thuật DPO (Direct Preference Optimization):

- Chuẩn bị dữ liệu
 - Thu thập các cặp phản hồi tốt và kém ứng với cùng một prompt.
- Tính xác suất sinh phản hồi
 - Tính log-probability (logp) mà mô hình gán cho cả phản hồi tốt và phản hồi kém.
- Tối ưu hóa loss DPO

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x_u, y_u^+, y_u^-) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta^t}(y_u^+ | \chi_u^+)}{\pi_{\text{ref}}(y_u^+ | \chi_u^+)} - \beta \log \frac{\pi_{\theta^t}(y_u^- | \chi_u^-)}{\pi_{\text{ref}}(y_u^- | \chi_u^-)} \right) \right]$$

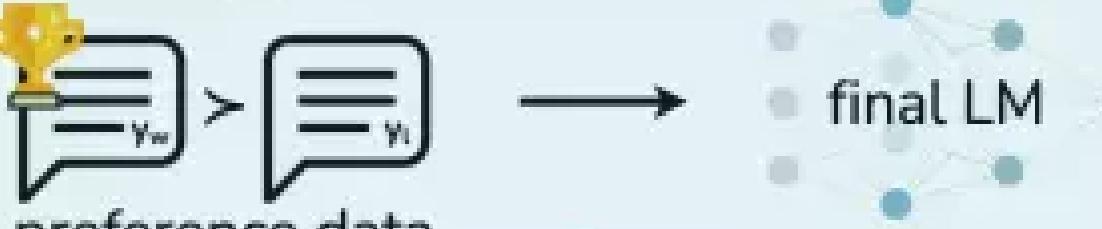
- Fine-tune mô hình
 - Cập nhật tham số của mô hình trực tiếp từ loss DPO để cải thiện chất lượng phản hồi.

Direct Preference Optimization (DPO)

x: "write me a poem about
the history of jazz"



maximum likelihood



XÂY DỰNG THÍ NGHIỆM



XÂY DỰNG THÍ NGHIỆM

Thu thập dữ liệu

- Mục tiêu: Xây dựng tập dữ liệu hỏi - đáp pháp lý chất lượng cao từ trang Law StackExchange để huấn luyện hoặc fine-tune mô hình ngôn ngữ.
- Công cụ sử dụng:
 - Selenium để tải trang và tương tác trình duyệt
 - BeautifulSoup để phân tích HTML và trích xuất dữ liệu

Stack**Exchange**

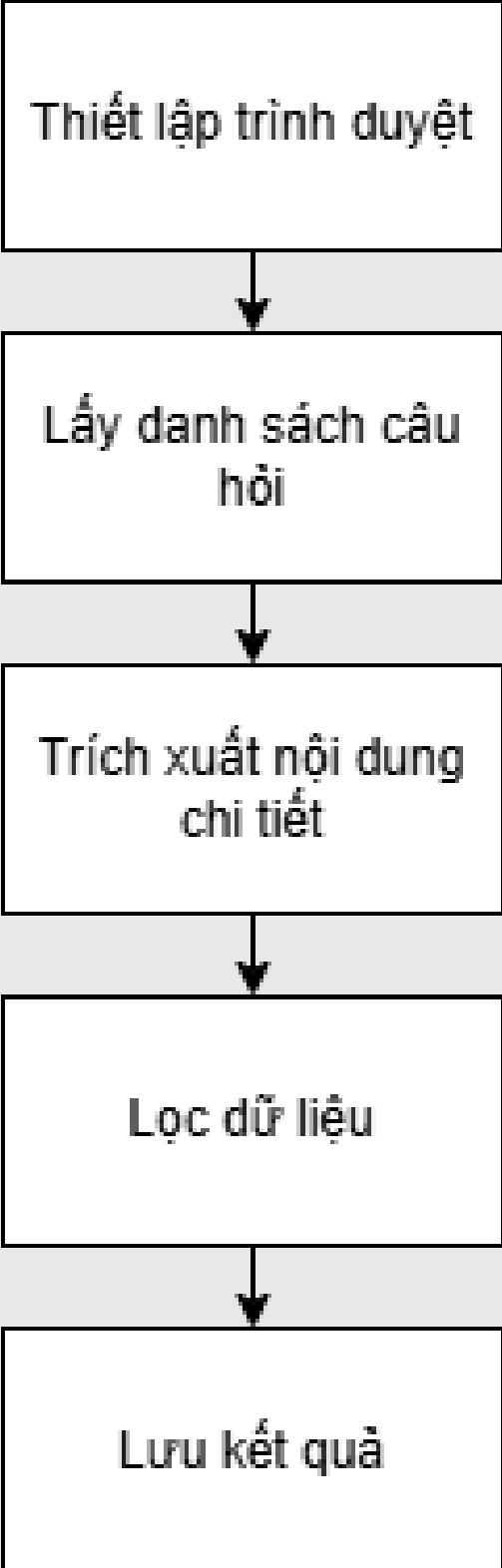


Thu thập dữ liệu

Các bước

- Thiết lập trình duyệt: Dùng Selenium + BeautifulSoup để chạy Chrome ở chế độ headless.
- Lấy danh sách câu hỏi: Truy cập từng trang câu hỏi theo thứ tự vote cao và thu thập link câu hỏi.
- Trích xuất nội dung chi tiết: Với từng link câu hỏi, lấy tiêu đề (prompt) và câu trả lời có vote cao nhất (response).
- Lọc dữ liệu:
 - Chỉ giữ lại câu hỏi kết thúc bằng dấu '?'.
 - Loại bỏ các câu trả lời ngắn hoặc thiếu nội dung.
- Lưu kết quả: Ghi dữ liệu dạng JSONL với cặp "prompt" và "response" đã được làm sạch.

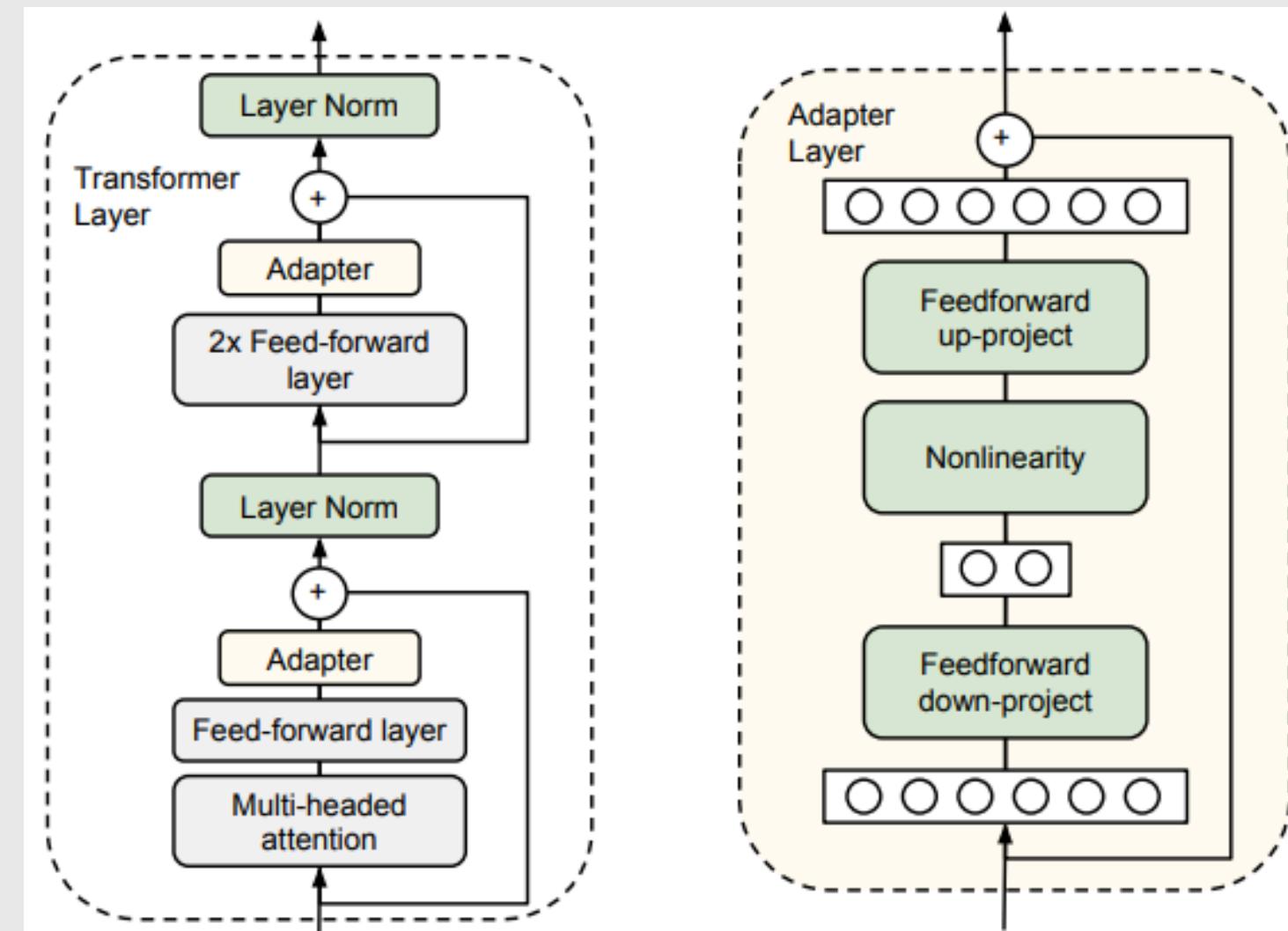
XÂY DỰNG THÍ NGHIỆM



XÂY DỰNG THÍ NGHIỆM

Sử dụng LoRa+Adapters cho Fine-tuning

- Mục đích:
 - Giảm chi phí và tài nguyên khi fine-tuning mô hình lớn.
 - Tránh phải cập nhật toàn bộ mô hình gốc (cố định backbone).
- Cách triển khai:
 - Adapter: chèn thêm các tầng nhỏ (bottleneck) giữa các layer transformer.
 - LoRA (Low-Rank Adaptation): chỉ cập nhật ma trận hạng thấp trong attention layer → giảm số tham số cần huấn luyện.
- Ngoài ra, nhóm còn thử nghiệm prompt tuning nhưng kết quả thí nghiệm trên mô hình nhỏ cho thấy LoRA + Adapter hiệu quả hơn prompt tuning rõ rệt



lớp Adapters được chèn vào mỗi khối Transformer

XÂY DỰNG THÍ NGHIỆM

Xây dựng mô hình

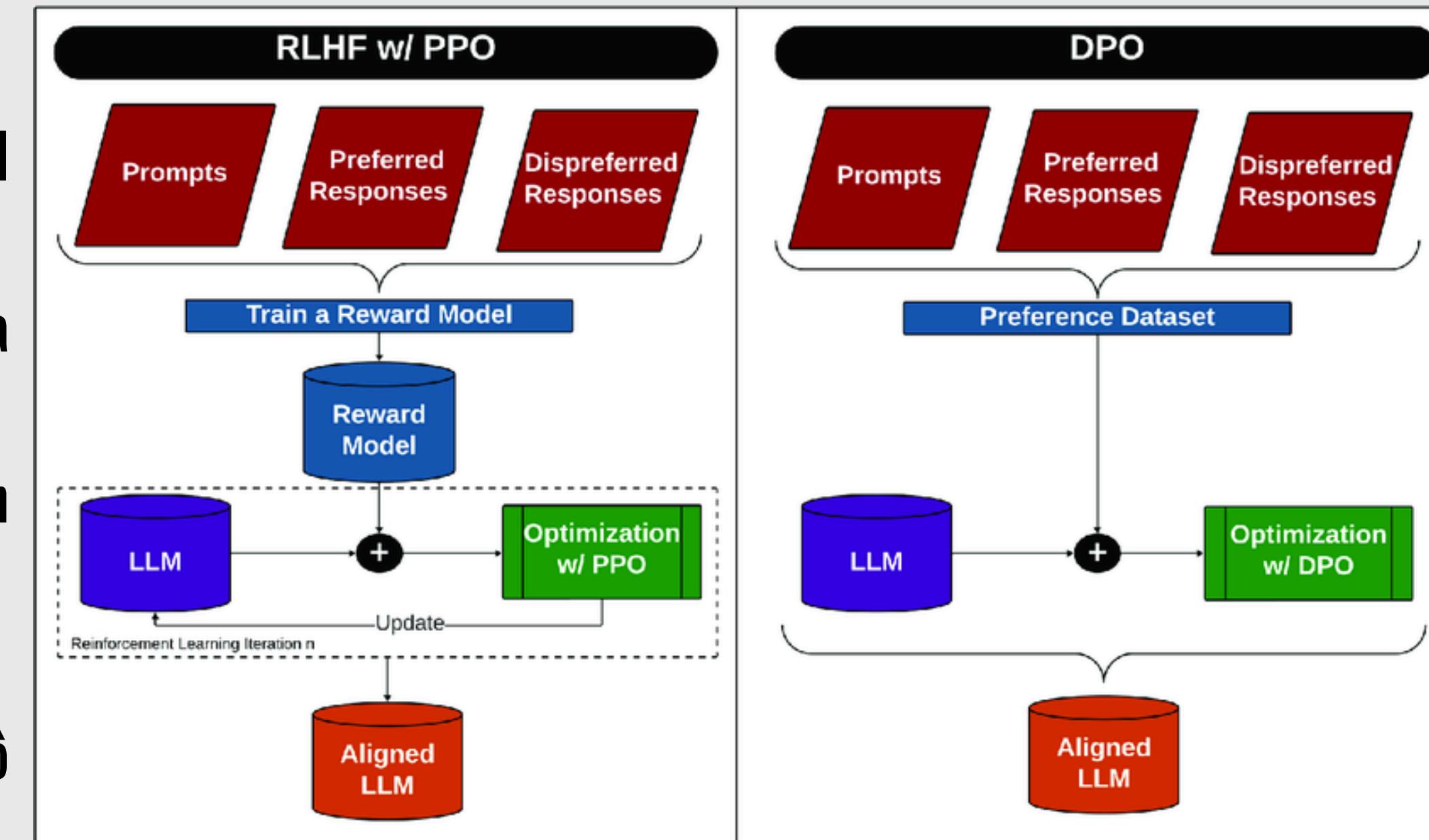
- Mô hình nền tảng:
 - Sử dụng GPT-2 Large (774M tham số) làm mô hình cơ sở.
- Fine-tuning bước 1: SFT (Supervised Fine-Tuning)
 - Dữ liệu: tập câu hỏi - câu trả lời đã thu được từ Law StackExchange.
 - Mục tiêu: huấn luyện mô hình sinh câu trả lời sát theo ví dụ gốc, dạng (prompt → response).

Mô hình	Lớp	Hidden size	Attention heads	Số tham số
GPT-2 Large	36	1280	20	762M

XÂY DỰNG THÍ NGHIỆM

Xây dựng mô hình

- Fine-tuning bước 2: RLHF: PPO & DPO
 - PPO (Proximal Policy Optimization)
 - Dùng mô hình đánh giá (Reward Model) để chấm điểm phản hồi.
 - Tối ưu hóa GPT-2 theo hướng tối đa hóa reward.
 - Có dùng thêm loss tiền huấn luyện (ptx loss) để giữ ổn định.
 - DPO (Direct Preference Optimization)
 - Dùng cặp phản hồi tốt/xấu để mô hình học trực tiếp từ so sánh.
 - Đơn giản và ổn định hơn PPO, không cần bước huấn luyện RM.



KẾT QUẢ

Results

KẾT QUẢ

Mô hình GPT-2 Large base và sau khi áp dụng SFT

Mô hình	Mean Normalized Score	Std Deviation	Win Rate
GPT-2 Base	0,7343	0,3494	31,50%
SFT (Fine-tuned)	0,8354	0,2937	68,50%

KẾT QUẢ

Mô hình sau khi áp dụng SFT và sau khi áp dụng DPO

- Với 1000 mẫu đánh giá

Mô hình	Mean Normalized Score	Std Deviation	Win Rate
SFT Model	0,8354	0,2937	60%
DPO Model	0,7314	0,3591	40%

- Với 3000 mẫu đánh giá

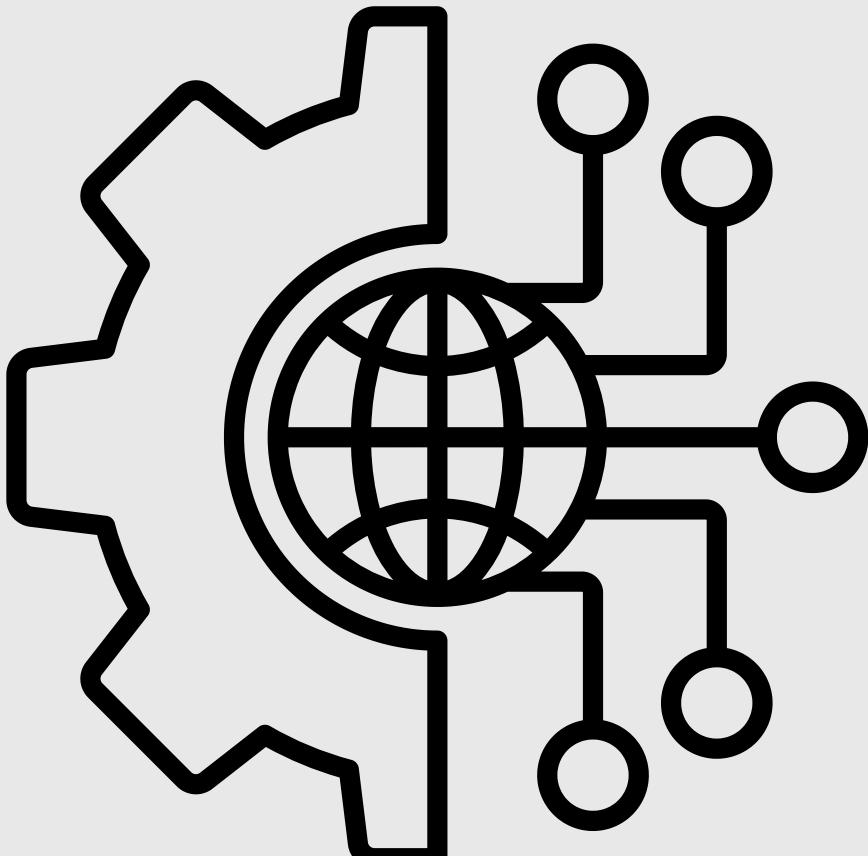
Mô hình	Mean Normalized Score	Std Deviation	Win Rate
SFT Model	0,8466	0,2854	61,27%
DPO Model	0,7397	0,3532	38,73%

HƯỚNG PHÁT TRIỂN



HƯỚNG PHÁT TRIỂN

- Mở rộng tập dữ liệu: Thu thập thêm dữ liệu pháp lý từ nhiều nguồn (Vietnamese Law, case studies...).
- Triển khai API và ứng dụng thực tế: Tích hợp chatbot pháp lý trên web/app, hỗ trợ tra cứu pháp luật tự động.



- Sử dụng Retrieval-Augmented Fine-Tuning (RAFT): Kết hợp truy xuất tri thức trong quá trình fine-tune, giúp mô hình trả lời sát ngữ cảnh và đầy đủ hơn.
- Multi-Task Learning: Fine-tune đồng thời nhiều nhiệm vụ như hỏi đáp, phân loại luật, trích xuất điều khoản để mô hình hiểu pháp lý sâu hơn.



KẾT LUẬN



KẾT LUẬN

- Mô hình ngôn ngữ lớn có thể được điều chỉnh hiệu quả để giải quyết các bài toán chuyên biệt như hỏi đáp pháp lý.
- Việc kết hợp giữa dữ liệu phù hợp và kỹ thuật huấn luyện hiện đại giúp cải thiện chất lượng đầu ra rõ rệt.



- Các phương pháp như SFT, PPO, DPO và LoRA giúp tinh chỉnh mô hình hiệu quả, tiết kiệm chi phí mà vẫn giữ hiệu suất cao.
- Hướng tiếp cận này mở ra nhiều tiềm năng ứng dụng trong các lĩnh vực có yêu cầu hiểu biết chuyên sâu như y tế, luật, tài chính.



THANK YOU FOR YOUR ATTENTION

