UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

Pedro Schindler Freire Brasil Ribeiro

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

PEDRO SCHINDLER FREIRE BRASIL RIBEIRO

Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): PEDRO SCHINDLER FREIRE BRASIL RIBEIRO

Título do trabalho: Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Pedro Schindler Freire Brasil Ribeiro**, **Discente**, em 04/02/2026, às 16:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 13/03/2026, às 11:44, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5957159** e o código CRC **E2E67F15**.

Referência: Processo nº 23070.005560/2026-70 SEI nº 5957159

PEDRO SCHINDLER FREIRE BRASIL RIBEIRO

Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

RIBEIRO, PEDRO SCHINDLER FREIRE BRASIL
Aprendizado por Reforço para Decomposição de Prompts [manuscrito]: Treinamento e Avaliação de Modelo Planejador / PEDRO SCHINDLER FREIRE BRASIL RIBEIRO. - 2025.
51 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Large Language Models. 3. Aprendizado por Reforço.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004

PEDRO SCHINDLER FREIRE BRASIL RIBEIRO

Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 09 de dezembro de 2025.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Dr. Sávio Salvarino Teles de Oliveira
(INF-UFG)

PEDRO SCHINDLER FREIRE BRASIL RIBEIRO

Aprendizado por Reforço para Decomposição de Prompts

Treinamento e Avaliação de Modelo Planejador

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Aprendizado por reforço aplicado a LLMs**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; Large language models; Aprendizado por reforço.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Reinforcement learning applied to LLMs**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

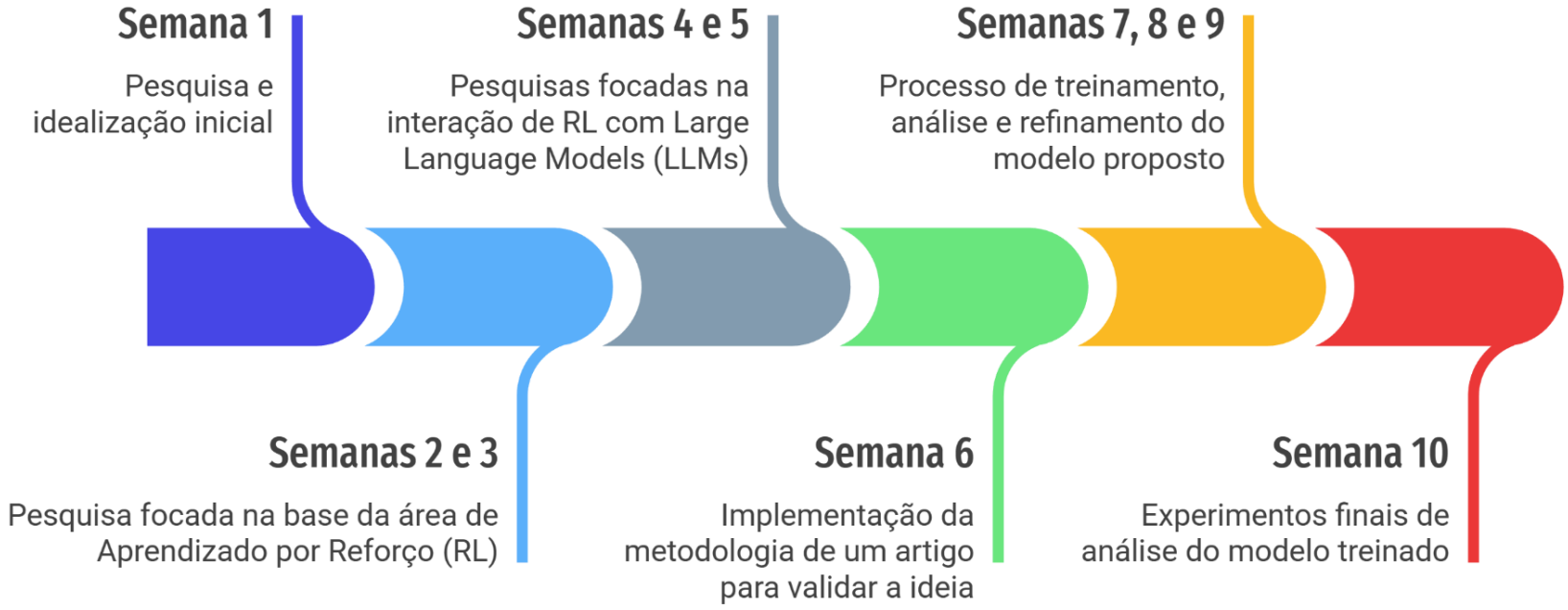
Keywords: Artificial intelligence; Large language models; Reinforcement learning.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada



Pedro Schindler Freire Brasil Ribeiro

Especialista em: Aprendizado por Reforço aplicado a Large Language Models

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: **Pedro Schindler Freire Brasil Ribeiro**

Especialidade: **Aprendizado por Reforço aplicado a Large Language Models**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada começou na **Semana 1** com a definição estratégica do tema que guiaria toda a Residência: a decomposição de prompts complexos. A partir da idealização do design de um "LLM Decompositor" e do estudo inicial, estabeleci o objetivo de treinar, usando Aprendizado por Reforço (RL), um modelo LLM (*Large Language Model*) capaz de quebrar instruções em sub-tarefas intermediárias. As definições iniciais e o planejamento macro desta fase podem ser consultados em detalhes no material disponibilizado no **Apêndice 1**. Foi neste momento que percebi o potencial de utilizar modelos menores para orquestrar tarefas complexas, lançando a base para o que viria a ser o foco em *Prompt Chaining*.

A partir da definição do tema, dediquei as **Semanas 2** e **3** para construir uma base sólida em RL. A leitura aprofundada da obra de Sutton e Barto, especificamente os capítulos sobre métodos de política e aproximação de função, foi fundamental. Paralelamente, realizei um levantamento de *surveys* recentes para conectar a teoria clássica às aplicações

[1] Dez Semanas, entre setembro de 2025 e dezembro de 2025.

modernas em LLMs, culminando na criação de uma linha do tempo evolutiva da área. No **Apêndice 2**, é possível encontrar a síntese destas leituras e a estruturação teórica que suportou as decisões técnicas subsequentes.

Durante o período das **Semanas 4** e **5**, a pesquisa se afunilou na interação entre RL e LLMs. Foi possível classificar a literatura em duas vertentes: (1) alinhamento e (2) aprimoramento de capacidades fundamentais. Optei por focar no aprimoramento e, especificamente, em utilizar o RL para “elevar” a inteligência do modelo (*Reasoning*). Neste período, defini o domínio de *Creative Writing* e escolhi o algoritmo GRPO (*Grouped Ranking Policy Optimization*) para a implementação, conforme detalhado nos estudos do **Apêndice 3**. Esta fase foi crucial para transformar o conceito em um plano de implementação.

A **Semana 6** foi um marco de validação na minha jornada. Antes de iniciar o treinamento pesado, implementei a metodologia de um artigo de referência (*Sun et al., 2024*) para testar a hipótese do *Prompt Chaining* na tarefa de sumarização. Os resultados foram expressivos, demonstrando uma superioridade de 78% da técnica de encadeamento sobre o *prompt* único (*stepwise*). Além disso, validei a compatibilidade dessa técnica com o *Reasoning*, o que demonstrou uma melhora menor, mas ainda assim expressiva - conforme os dados de validação apresentados no **Apêndice 4**. Esse resultado me confirmou a potencialidade deste método, melhorando resultados até em cima de modelos que já utilizam algum tipo de *Reasoning* e demonstrando a capacidade de melhorar ainda mais esses resultados, com um treinamento adequado para realização dessa tarefa.

O intenso período das **Semanas 7**, **8** e **9** foi dedicado ao ciclo de treinamento, análise e refinamento do modelo proposto. Enfrentei desafios significativos, como a estagnação da *loss* inicial e a dificuldade do modelo "Juiz" em fornecer sinais claros de melhora. A virada de chave ocorreu quando implementei um sistema de múltiplos juízes com *baselines* dinâmicas (uma fraca e uma forte), o que permitiu ao modelo treinado (Gemma-4B) superar o modelo base com uma taxa de vitória entre 70% e 80%. Os registros das *runs*, as curvas de aprendizado e as estratégias de recompensa adotadas para "elevar o teto" de performance estão documentados no **Apêndice 5**.

Na **Semana 10**, realizei os experimentos finais para analisar a transferibilidade e os limites do modelo treinado. Ao testar o *Prompt Chaining* com um modelo diferente (Qwen) para aprimorar o *Reasoning*, descobri que o modelo planejador é sensível ao executor utilizado durante o treinamento. Embora o encadeamento com *Reasoning* tenha superado o encadeamento simples, a descoberta do "descasamento" entre planejador e executor trouxe um *insight* valioso sobre a necessidade de co-treinamento. Os resultados finais e a análise deste fenômeno, que abre portas para trabalhos futuros, constam no **Apêndice 6**.

Em função de tudo que vivi nesta jornada, gostaria de deixar registrado que a exploração de RL para aprimorar capacidades fundamentais de LLMs se provou um caminho extremamente promissor. Ficou evidente o potencial da abordagem desenvolvida; e que outras técnicas de engenharia de *prompt* também podem ser internalizadas através de treinamento, estabelecendo-se como novas capacidades fundamentais. Além disso, acredito que o treinamento com o *Prompt Chaining* possui um vasto horizonte de expansão. Ao generalizar para novos domínios e escalar o processo, esta técnica tem o potencial de atingir a maturidade e o impacto que hoje observamos nos modelos que incorporam *Reasoning*.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 3 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante esta primeira Semana, foram realizadas as seguintes atividades:

-Definição do tema central que será abordado durante a Residência:
Treinamento de um LLM para decompor prompts em sub-prompts intermediários: Ideia

-Pesquisa e estudos relacionados ao tema: Pesquisa

-Idealização do design e treinamento desse LLM decompositor: Implementação

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

-Averiguar a viabilidade da ideia, em relação a questões práticas como disponibilidade de máquinas para treinamento, tempo de treinamento, etc.
-Demo em pequena escala, sem treinamento, para avaliar a eficácia da ideia.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

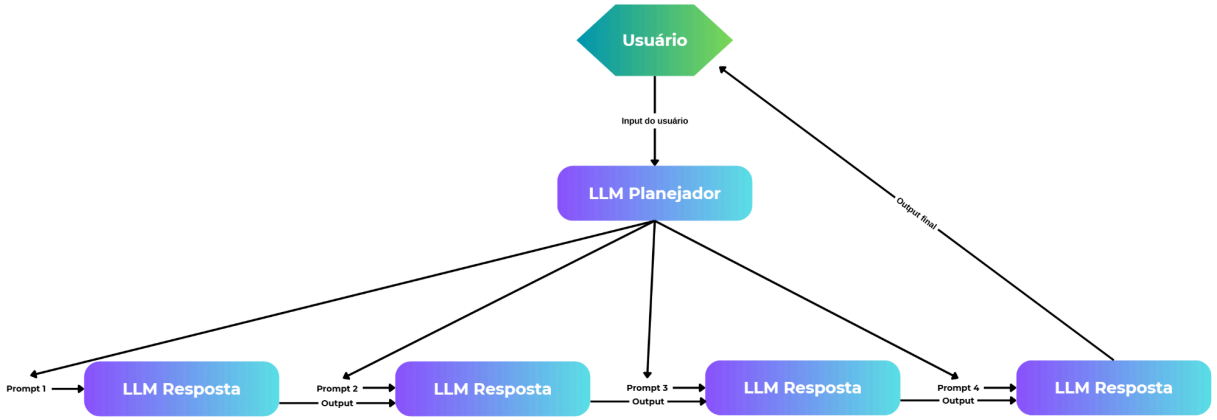
CEDRIC LUIZ DE CARVALHO: Go!

Documento “Ideia”

Ideia:

A ideia central é a criação de um "LLM Planejador": Em vez de tentar responder diretamente a uma consulta complexa do usuário, a sua principal função é decompor essa consulta em uma sequência de etapas menores e lógicas. Cada etapa é formulada como um prompt autônomo que pode ser executado por outro "LLM Resposta". Este processo imita a forma como um especialista humano aborda um grande projeto, construindo um alicerce sólido através de passos intermediários para garantir um resultado final de qualidade superior, com mais detalhes, coerência e precisão.

Para interações simples e diretas, como "Oi" ou "Qual a previsão do tempo para hoje?", o LLM Planejador reconhece a baixa complexidade e não realiza nenhuma divisão, respondendo de forma imediata. No entanto, ao receber uma tarefa complexa, ele realiza a divisão, onde a quantidade da decomposição é diretamente proporcional à complexidade da solicitação. Um pedido para escrever um conto pode ser dividido em quatro etapas, enquanto um pedido para criar um plano de negócios completo pode exigir dez ou mais etapas.



Exemplo:

Intuitivamente, é esperado que a resposta do exemplo 1B e 2B seja melhor que a resposta do 1A e 2A. O objetivo do "LLM Planejador" é receber como input o exemplo 1/2A e retornar como output, a lista de prompts dos exemplos 1/2B, para ser executado sequencialmente por outro LLM Resposta.

	Prompt 1	**Prompt 2**	**Prompt 3**	**Prompt 4**	**Prompt 5**
Exemplo 1A	Escreva uma história sobre o tema X.				

Exemplo 1B	Escreva uma história sobre o tema X. Começe escrevendo apenas o esboço da trama, dividindo em 3 atos.	Agora escreva os personagens que estarão envolvidos nessa trama, suas personalidades, falhas e jornadas durante a história.	Agora escreva os arcos de personagem que cada um vai ter durante a história, encaixando no esboço feito anteriormente.	Revise o que foi escrito até agora para que os temas e pontos da história sejam originais, imprevisíveis, e ressoem com o leitor.	Por fim, baseado em tudo que foi feito até agora, escreva a história em si sobre o tema X.
Exemplo 2A	Nesse meu código Y, teve esse erro X, resolva.				
Exemplo 2B	Analise meu código Y	Ele está dando um erro X, liste 5 possíveis e prováveis causas para esse erro.	Agora pense em soluções para esse erro, pense fora da caixa.	Agora tente modificar o código para resolver esse erro.	

Documento “Implementação”

Treinamento

Treinar com GRPO (group relative policy optimization) um LLM Planejador para decidir quando dividir um pedido do usuário em etapas e como compor a melhor sequência de prompts intermediários, sempre com no máximo 10 etapas e idealmente não dividindo tarefas triviais. Buscando superar a resposta direta em qualidade, precisão e aderência aos requisitos, mantendo custo e latência sob controle. Este plano representa uma proposta inicial, sujeita a ajustes e validações futuras.

Usar um Qwen-3/4B como Planejador e um Qwen-3/4B congelado como Executor (responsável por executar cada etapa e também por responder diretamente sem plano). Talvez testar incluir um Planejador congelado adicional para servir como baseline multi-etapas. Manter todos os modelos de execução e baseline sem atualização de pesos; atualizar apenas o Planejador principal. Um componente Avaliador calcula métricas objetivas (código, matemática, etc.) ou, em domínios subjetivos, usa um LLM juiz leve com rúbrica que julga o output.

Para cada prompt, primeiro geramos um pequeno conjunto de respostas diretas com o LLM Resposta congelado (por exemplo M=2 a 3 amostras). Em paralelo, geramos um pequeno conjunto de planos do Planejador treinável (por exemplo K=2 a 4 amostras). Talvez, também gerar 1 ou 2 planos usando um Planejador congelado e executar com o mesmo LLM Resposta, obtendo uma segunda linha de base. Em seguida, executamos cada plano do Planejador treinável passo a passo: o LLM Resposta recebe o contexto do passo, o output do passo anterior e produz o próximo artefato; ao final, segue a instrução inicial e produz a resposta final. Avaliamos todas as saídas e calculamos a recompensa de cada plano do Planejador.

A recompensa é construída para só ser positiva quando a divisão do LLM Planejador fizer diferença real. Primeiro definimos sucesso: Para tarefas objetivas, sucesso é acerto exato ou testes passando; para tarefas criativas, sucesso é a avaliação subjetiva do LLM juiz (Podendo ser de algum benchmark como Creative Bench, ou feito do zero). Se o output da pipeline do planejador estiver errado ou pior que a baseline, a recompensa não é positiva. O princípio é: só premiar a divisão quando ela muda o resultado para melhor. Assim, definimos um “núcleo” de recompensa relativo às linhas de base. Se o plano do Planejador alcança sucesso e a melhor baseline (direta e, se usada, multi-etapas congelada) falha, o núcleo recebe valor positivo. Se o plano falha e alguma baseline tem sucesso, o núcleo recebe um valor negativo. Se ambos acertam ou ambos falham, o núcleo é zero. Opcionalmente somamos um termo pequeno de diferença de qualidade (score do plano menos o melhor score da baseline) apenas como modelagem suave, sem inverter a lógica principal. As baselines (respostas diretas e, talvez, as multi-etapas congelada) não entram no grupo do GRPO; elas apenas servem de comparação para calcular a recompensa.

Além do núcleo, aplicamos supervisão de processo para avaliar a utilidade dos passos intermediários. Por exemplo, passos claros, corretos e não redundantes somam pontos; passos vagos, repetidos ou que induzem a resposta errada, reduzem pontos. Em paralelo, aplicamos penalidades estruturais: bônus por JSON válido; penalidade crescente conforme o número de passos; penalidade forte ao ultrapassar o limite de 10 passos; penalidade extra quando prompts triviais são indevidamente divididos.

Com as recompensas finais calculadas para cada um dos K planos no grupo, o algoritmo GRPO realiza a atualização: O GRPO calcula a recompensa média do grupo (g). Para cada plano no grupo, calcula-se a "vantagem", que é a recompensa daquele plano menos a recompensa média do grupo. Os pesos do LLM Planejador são atualizados via gradiente. O objetivo é induzir planos curtos, enxutos e eficazes.

Documento “Pesquisa”

[2505.09388] Qwen3 Technical Report (LLM)
[2412.16720] OpenAI o1 System Card (Reasoning)
[2501.12948] DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (Reasoning with GRPO)
[2201.11903] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Chain of thought prompting technique)
Chain complex prompts for stronger performance - Anthropic (Prompt chaining prompting technique)
[2406.00507] Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization (Prompt chaining prompting technique)
[2402.03300] DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models (Introduces GRPO)
[2312.06281] EQ-Bench: An Emotional Intelligence Benchmark for Large Language Models (Benchmark, Judge-LLM)

A leitura e análise dessas fontes foram cruciais para a formulação da proposta, que se posiciona na intersecção de técnicas de prompting, otimização por reforço e avaliação de modelos. A viabilidade técnica do projeto se apoia em modelos de base robustos, como os detalhados no relatório técnico do Qwen. Para que o treinamento com GRPO seja bem-sucedido, é indispensável uma função de recompensa precisa, e a metodologia de utilizar um LLM-juiz para avaliação de tarefas complexas, como explorado no EQ-Bench, oferece um caminho prático para gerar os sinais de aprendizado necessários para tarefas não objetivas.

A pesquisa se fundamenta na evolução de conceitos como o Chain-of-Thought (CoT), que começou como uma técnica de prompting para induzir o raciocínio passo a passo em LLMs, e que posteriormente foi transformado em uma capacidade intrínseca e treinável do modelo através de Reinforcement Learning (RL), como demonstrado pela DeepSeek com a introdução do GRPO no DeepSeekMath e sua aplicação no DeepSeek-R1, e do o1 da OpenAI. De forma análoga, o "prompt chaining" é hoje uma técnica de prompting manual para decompor tarefas complexas, com ganhos de eficácia e qualidade comprovados pelo paper 2406.00507 citado acima.

A hipótese central da minha ideia é aplicar um salto evolutivo similar: usar o treinamento com GRPO para transformar o prompt chaining de uma técnica manual em um processo de planejamento otimizado e aprendido, que inicialmente chamo de "chain-of-prompts".

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a segunda Semana, foram realizadas as seguintes atividades:

Em cima dos estudos base realizados na semana anterior, voltei meu foco para o Aprendizado por Reforço (RL) aplicado a LLMs.

*__Principal leitura:__ "Reinforcement Learning: An Introduction" por Sutton e Barto.
*Filtragem da leitura para os capítulos mais relevantes, selecionados: 1, 13, 15, 17.
*Leitura dos resumos de cada um desses capítulos (feitos usando a ferramenta do Gemini).

*__Levantamento de leitura adicional__:
*Para complementar a pesquisa da semana anterior sobre GRPO, busquei referências sobre outros algoritmos relevantes, como PPO, DPO, RLHF e RLAIF
*Também foram buscadas referências para facilitar o aprendizado, como surveys recentes sobre a área de RL e sua história, e artigos explicando a funcionalidade e diferença dos algoritmos relevantes.

* **Detalhes:** Pesquisa 2

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

-Ler por completo os capítulos selecionados do livro "Reinforcement Learning: An Introduction", produzindo uma síntese dos pontos principais identificados.

-Leitura das referências coletadas sobre os algoritmos relevantes de RL.

-Voltar o estudo para focar mais na conexão da área de RL com LLMs.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Documento “Pesquisa 2”

Conteúdo base de RL:
Principal leitura: "Reinforcement Learning: An Introduction" por Sutton e Barto
http://incompleteideas.net/book/RLbook2020.pdf

Principais capítulos identificados: 1 13 15 17
1.Introduction: Este capítulo foi selecionado para construir uma base sólida, compreendendo a definição fundamental do problema de aprendizado por reforço e seus elementos essenciais antes de avançar para os algoritmos complexos

13.Policy Gradient Methods: Este capítulo foi selecionado para eu entender a teoria central por trás de algoritmos modernos como PPO e GRPO, que são a base para frameworks como o RLHF e fortemente relacionados a LLMs. Ele explica como otimizar diretamente a política de um agente, que é exatamente a abordagem usada para ajustar o comportamento de grandes modelos de linguagem.

15.Neuroscience: Este capítulo foi selecionado porque eu busco uma compreensão mais profunda sobre as origens e a inspiração biológica dos algoritmos de RL. É um tema que me interessa fortemente, essa tentativa de replicar a biologia cerebral biológica digitalmente, e que certamente será muito útil ter esse conhecimento.

17.Frontiers: Este capítulo aborda os desafios atuais e das futuras direções da área. Por isso, julguei essencial para me ajudar a posicionar meu trabalho no contexto da pesquisa de ponta e a refletir sobre os problemas que ainda estão em aberto no campo do aprendizado por reforço.

Deep Reinforcement Learning: An Overview
https://arxiv.org/abs/1701.07274

A (Long) Peek into Reinforcement Learning
https://lilianweng.github.io/posts/2018-02-19-rl-overview/

Artigos que introduziram os principais algoritmos de reforço (Para a área de LLMs)
GRPO (Estudado na semana anterior = Papers Deepseek math e Deepseek R1)
PPO (https://arxiv.org/abs/1707.06347)
RLHF (https://arxiv.org/abs/2009.01325) ; https://epub.ub.uni-muenchen.de/125328/1/2312.14925v2.pdf
RLAIF (https://arxiv.org/abs/2212.08073)

Conteúdo focado em LLM:

Reinforcement Learning Enhanced LLMs: A Survey
https://arxiv.org/abs/2412.10400v3

A Technical Survey of Reinforcement Learning Techniques for Large Language Models
https://arxiv.org/pdf/2507.04136

A Deep Dive into RLfor LLM Reasoning
https://arxiv.org/pdf/2508.08221

Phi-4-reasoning Technical Report
https://arxiv.org/pdf/2504.21318

Training language models to follow instructions with human feedback
https://arxiv.org/abs/2203.02155

Why Reinforcement Learning Might be the Most Overlooked AI Breakthrough Yet
https://statistician-in-stilettos.medium.com/a-survey-of-advancements-in-gen-ai-with-reinforcement-learning-how-rlhf-and-reasoning-llms-are-cf9ad5935861

PPO, DPO & GRPO: Reinforcement Learning Techniques for Training LLMs
https://medium.com/@mandeep0405/ppo-dpo-grpo-reinforcement-learning-techniques-for-training-llms-193459ffc14e

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 17 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a terceira Semana, foram realizadas as seguintes atividades:

Prosseguimento dos estudos sobre RL para LLMs:

*__Principal leitura:__ "Reinforcement Learning: An Introduction" por Sutton e Barto.
*Leitura completa do capítulo 1. 22 Páginas. Síntese: Cap 1
*Leitura completa do capítulo 13. 18 Páginas. Síntese: Cap 13
*Leitura com LLM do capítulo 15. 44 Páginas. Síntese: Cap 15
*Leitura com LLM do capítulo 17. 22 Páginas. Síntese: Cap 17

Leitura com LLM (Gemini) foi feita no capítulo 15 devido ao tamanho longo e a baixa importância. E foi feito no capítulo 17 para que o Gemini indique as evoluções ocorridas desde a época da escrita do livro (2018). Prompts usados estão incluídos no docs de detalhes.

*__Leitura resumida de artigos mais recentes da área__:
*Reinforcement Learning Enhanced LLMs: A Survey (Wang et al, 2024)
*Comprehensive Survey of Reinforcement Learning: From Algorithms to Practical Challenges (Ghasemi et al, 2024)

*__Produção de uma linha do tempo, baseada no Cap 1 do livro do Sutton e Barto, e estendendo até os dias atuais, usando como fonte os surveys lidos citados acima.__

* __Detalhes e linha do tempo:__ Pesquisa 3

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

-Definir o problema e solução inicialmente proposta, dentro do contexto estudado durante as últimas duas Semanas.
-Após isso, em cima dessa definição, buscarei refinar meus estudos para soluções similares a problemas similares ao proposto, buscando identificar vantagens, desvantagens, inovações, falhas, etc.

Observação: **[caso precise fazer alguma observação, de qualquer "natureza"]**

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Documento “Pesquisa 3”

Linha do Tempo

Baseada no Capítulo 1 do Sutton e Barto

- **1911 - Lei do Efeito (Thorndike)** - Conceito psicológico de que ações com consequências satisfatórias são reforçadas.
- **1948 - Sistema de Prazer-Dor (Alan Turing)** - Descreveu um "sistema de prazer-dor" em um relatório, uma das primeiras ideias computacionais baseadas na Lei do Efeito.
- **1957 - Programação Dinâmica (Bellman)** - Abordagem de controle ótimo que introduz a equação de Bellman e funções de valor.
- **1959 - Jogador de Damas (Samuel)** - Primeiro programa famoso a implementar ideias de Aprendizado por Diferença Temporal (TD).
- **Anos 1980 - Ator-Crítico (Sutton & Barto)** - Arquitetura que separa o aprendizado da política (ator) e da função de valor (crítico).
- **1989 - Q-Learning (Watkins)** - Algoritmo model-free que unificou de forma elegante as principais correntes de pesquisa em RL.
- **1992 - TD-Gammon (Tesauro)** - Demonstrou que o RL com redes neurais poderia atingir nível sobre-humano em um jogo complexo.

Expansão para os Dias Atuais

- **2013 - Deep Q-Network (DQN)** - Combina Q-Learning com redes neurais profundas para jogar jogos de Atari em nível super-humano.
- **2016 - AlphaGo** - Vence o campeão mundial de Go usando redes neurais profundas e busca em árvore (MCTS).
- **2017 - Proximal Policy Optimization (PPO)** - Algoritmo de gradiente de política da OpenAI que se tornou um padrão pela sua estabilidade e eficiência.
- **2017 - AlphaGo Zero** - Aprende a jogar Go do zero, apenas por auto-play, superando todas as versões anteriores.
- **2020 - RLHF (Reinforcement Learning from Human Feedback)** - Técnica crucial para alinhar LLMs (como o GPT-3/4) com as intenções e preferências humanas.
- **2023 OpenAI O1** - Primeiro LLM usando RL para “reasoning”, melhorando a qualidade das respostas ao fazer o modelo “pensar” antes de responder.
- **2024 - Group Relative Policy Optimization (GRPO)** - Proposto pelo DeepSeek para treinar reasoning em LLMs, proposto como uma alternativa mais eficiente ao PPO.

Resumos:

Os seguintes links são resumos feitos dos capítulos relevantes do livro "Reinforcement Learning: An Introduction" por Sutton e Barto:

https://docs.google.com/document/d/1U7RST7uId0-zLYnAB28rkCbUDLThQkoly-8jz1RyITk/edit?usp=sharing
https://docs.google.com/document/d/1TR_MyBtBIzq7YxVsQcVhQoLG_0Kr0jxI66X0nSruo44/edit?usp=sharing
https://docs.google.com/document/d/1uuRbTXUS-Ywek17kJ-zfB0htszXJiMq4ZZjufFCLKkc/edit?usp=sharing
https://docs.google.com/document/d/10xe23I-vYJpzb608xQ1MWzOWCPNrviVdy8nL44zw2GU/edit?usp=sharing

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 24 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a quarta Semana, foram realizadas as seguintes atividades:

***Classificação feita sobre a área de RL aplicado a LLMs:**

1. Classificação por Metodologia de Treinamento

1.1 RL com Modelos de Recompensa Explícitos
Abordagem tradicional onde um modelo de recompensa é treinado primeiro para depois guiar o treinamento do LLM
Técnica: RLHF, RLAIF
Exemplos: GPT, Gemini, Claude(RLAIF)...

1.2 Otimização Direta de Preferências
Simplificar o processo, eliminando a necessidade de treinar um modelo de recompensa separado
Técnica: DPO
Exemplos: Llama 3 , Qwen2 , Phi-3…

2. Classificação por Objetivo da Aplicação

2.1 RL para Alinhamento de Comportamento:
Foca em tornar os LLMs mais "seguros", "úteis" e alinhados às expectativas humanas.
Técnica: RLHF, RLAIF
Exemplos: Todos os LLMs atuais.

2.2 RL para Aprimoramento de Capacidades Fundamentais:
Foca em melhorar a qualidade e a inteligência das respostas do modelo.
Técnica: Reasoning
Exemplos: OpenAI O1, Deepseek R1…

***Dado essa classificação, meu tema adiante será mais focado na classe 2.2:** "RL para Aprimoramento de Capacidades Fundamentais de LLMs"

***Qual problema esse tipo de técnica busca solucionar?**
Diminuir esforço manual e volatilidade ao usar técnica de prompt (No caso de Reasoning, Chain of Thought)

***Pesquisas feitas na semana, focado no ponto 2.2:**
A Survey of Reinforcement Learning for Large Reasoning Models (Zhang et al, 2025)
Part I: Tricks or Traps? A Deep Dive into RL for LLM Reasoning (Liu et al, 2025)

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

-Revisar planejamento feito na primeira Semana, agora com o conhecimento extra adquirido nas 3 últimas Semanas
-Depois disso, talvez começar a executar o planejamento refeito.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 2 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a quinta Semana, foram realizadas as seguintes atividades:

*__Tema:__ RL para Aprimoramento de Capacidades Fundamentais de LLMs

*__Objetivo:__ Explorar e desenvolver novas técnicas além de Reasoning.

Prompt technique	–(RL)→	New capacity
Chain of Thought	–(RL)→	Reasoning
Prompt Chaining	–(RL)→	?
(...)	–(RL)→	?

*__Planejamento da implementação revisada:__ Implementação Revisada

*__Colab implementando e testando GRPO para meu aprendizado:__ grpo.ipynb

Inicialmente começar com 1 domínio: creative writing.
Dataset identificado: Disya/eq-bench-creative-writing-v3,
Validar e depois expandir para outros domínios (Math, Code, etc)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

-Discutir com especialistas de RL sobre minha proposta e implementação, receber feedback
-Começar a codar o código de treinamento teste para essa fase inicial com apenas um domínio

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Documento “Implementação Revisada”

O treinamento do modelo planejador foi estruturado para utilizar Grouped Ranking Policy Optimization (GRPO), uma abordagem de aprendizado por reforço. O modelo base é `gemma-3-4b`, otimizado com LoRA através da biblioteca Unsloth para eficiência de memória e velocidade. O objetivo principal do modelo é atuar como um "Orquestrador de Narrativas", que recebe um prompt de escrita criativa do dataset `Disya/eq-bench-creative-writing-v3` e gera um plano estratégico em formato JSON. Este plano consiste em uma sequência de até seis passos, seguindo um fluxo de trabalho profissional que vai desde a criação do esqueleto da história e dos personagens até a reescrita do esboço integrado e a geração da prosa final. O processo de treinamento não avalia o plano diretamente, mas sim a qualidade do resultado final que ele produz.

Para calcular a recompensa, cada plano gerado pelo modelo em treinamento é executado por um modelo "Executor" externo, `google/gemma-3-4b-it`, acessado via API da OpenRouter. Esse Executor segue os passos do plano para escrever uma história completa. A qualidade desta história é então comparada com duas linhas de base para contextualizar sua performance. A primeira é uma "linha de base direta", onde a mesma história é gerada em uma única etapa, sem um plano. A segunda é uma "linha de base multi-passo", produzida por um modelo planejador de API (`google/gemma-3-4b-it`) que também cria um plano e o executa. Um modelo "Juiz" mais avançado, `gpt-4o-mini`, avalia as três histórias resultantes (a do modelo treinado, a direta e a multi-passo) e atribui uma pontuação de qualidade normalizada (0 a 1) para cada uma.

A função de recompensa é projetada com um currículo de aprendizado para guiar o modelo de forma progressiva. A recompensa final não é a pontuação absoluta da história, mas sim uma medida relativa à melhor das duas linhas de base. No início do treinamento, o modelo recebe um sinal binário simples de vitória ou derrota (positivo se superar a melhor linha de base, negativo caso contrário). À medida que o treinamento avança, o sinal de recompensa transita para um valor contínuo baseado na magnitude da diferença de qualidade, utilizando uma função `tanh` para escalar a recompensa. Isso incentiva o modelo a não apenas vencer, mas a gerar planos que resultem em histórias significativamente melhores. Adicionalmente, a recompensa total inclui pequenas penalidades estruturais para planos que produzem JSON inválido, excedem o limite de passos, ou são considerados "triviais" (não contêm palavras-chave de planejamento estratégico), garantindo que o modelo aprenda a gerar planos bem formados, concisos e úteis. O treinamento e o envio de checkpoints para o Hugging Face Hub são gerenciados através do `GRPOTrainer` da biblioteca TRL, com integração ao Weights & Biases para monitoramento.

Documento “grpo.ipynb”

O link do colab abaixo contém um código de treinamento simples feito para me familiarizar com o algoritmo de treinamento, entender como funciona, e como implementar da melhor forma:
https://colab.research.google.com/drive/17ytUyxioCt6_dSJCFSieiThQHBShy465?usp=sharing

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 9 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a sexta Semana, foram realizadas as seguintes atividades:

*__Tema:__ RL para Aprimoramento de Capacidades Fundamentais de LLMs

*__Revisita de um paper estudado na primeira Semana:__
*Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization (Sun et. al 2024)
*Paper explora e avalia o uso de Prompt Chaining Vs Um prompt só (Stepwise), para a tarefa de resumir texto.
*3 prompts: Draft, Critique, Refine. No stepwise, manda fazer tudo isso em um prompt apenas.
*Método de avaliação: LLM-as-a-judge comparando resultado com baseline prévia. Técnica trazida de outro paper "LLM Compare protocol (Liu et al., 2023)"
*Resultado: Prompt Chaining 45% melhor que um prompt só (gpt 4). 75% (gpt 3.5)

*__Replicação dos resultados:__
*GPT-4 muito caro, precisei trocar o modelo usado.
*Código feito para replicar resultados: Untitled26.ipynb
*Resultado: Prompt chaining 78% melhor que um prompt só.
*Testes adicionais: Reasoning é compatível com esse método? Traz ganhos adicionais?
*Resultado: Prompt chaining com reasoning 60% melhor que um prompt só com reasoning.
*Conclusão: Prompt chaining, na task de sumarização, melhorou os resultados tanto com reasoning desligado, quanto com reasoning ligado, demonstrando ser eficaz e compatível com essa técnica

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

-Depois de validar a técnica para o domínio de sumarização: Validar a ideia de gerar automaticamente o prompt chaining a partir do Stepwise prompt.
-Na prática: Codar e rodar o treinamento do LLM para fazer isso, no domínio inicialmente escolhido de creative writing.

Observação: **[caso precise fazer alguma observação, de qualquer "natureza"]**

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Documento “Untitled26.ipynb”

O link do colab abaixo contém o código de treinamento para replicar os resultados do paper de pesquisa citado no termo “Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization (Sun et. al 2024)”:
https://colab.research.google.com/drive/1Qei06ybt2je5fBP-rPdHzOhu7e-1Fk7I?usp=sharing

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 15 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a sétima Semana, foram realizadas as seguintes atividades:
*__Tema:__ RL para Aprimoramento de Capacidades Fundamentais de LLMs

*__Criação do código de treinamento inicial:__
- Um domínio: Creative writing
- Poucos steps (200) para testar apenas.
- Usei como base o código exemplo do Unsloth de GRPO para o Gemma: Gemma3_(1B)-GRPO.ipynb - Colab
- Código atual: treinamento-v1.ipynb
- Seguindo planejamento feito na quinta Semana.

*__Função de recompensa:__
- Segui mais ou menos o planejamento que tinha feito, mas com algumas modificações.
- Adição de recompensa por resposta em formato JSON válido e recompensa por menor número de steps.
- Reward = quality_reward (score_trained − max(score_direct, score_multistep)) + structural_reward + brevity_reward
- Normalizado para -1,1

*__Dificuldades:__
- Loss congelada em 0 usando o modelo Gemma.
- Decidir parâmetros ideais (Revisitei papers lidos anteriormente para checar. Deepseek, Phi)
- Fazer caber na 4090, por enquanto precisei mover cálculo das baselines e julgamentos para API.
- Escolha do modelo juiz para julgar e comparar as histórias. Importante pois esse sinal é o que guia o treinamento. Cheguei no gpt-4o-mini
- Fazer os prompts em si. Importante para o modelo começar o treinamento já sabendo mais ou menos o que fazer, pra não ficar perdido.

*__Resultado:__
- Consegui colocar para rodar terça de noite!
- Estimativa de tempo de treinamento: 30 horas

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Buscar alternativas para abaixar o tempo de treinamento, ver se consigo usar duas 4090 para tirar APIs e rodar tudo localmente, ou H100.
- Analisar resultado, entender o que der errado e o que melhorar.
- Tentar alinhar tudo para começar o treinamento principal do modelo antes da minha viagem, já que o treinamento será longo, e não saber se terei muito tempo para fazer isso durante a viagem (Ou se sequer terei acesso a internet lá).

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Documento “treinamento-v1.ipynb”

O link do colab abaixo contém a primeira versão do código de treinamento do modelo decompositor de prompts, explicado em mais detalhes no documento “Implementação Revisada” no Apêndice 3:
https://colab.research.google.com/drive/1anEsKGGprA4Dr3bGrUa5EXlLNeMg4x9O?usp=sharing

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 22 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a sétima Semana, foram realizadas as seguintes atividades:
***Tema:** RL para Aprimoramento de Capacidades Fundamentais de LLMs
***Objetivo:** Treinamento de um modelo usando RL para decomposição automática de prompts (Prompt Chaining)

***Resumo da Semana:**
-Muitas iterações, 7 dias, praticamente 7 runs de 24 horas, analisando e mudando parâmetros/recompensa/etc.
-Modelo começou pior, sem saber o que fazer, mas melhorou até paridade com baseline, mesmo em poucos steps (200).
-Número de divisões ótimo encontrado foi de 4. 1 prompt -> 4 sub-prompts.
-Impressão é que não precisa de muitos steps, 200 steps e 24h já faz muita coisa.

***Principal problema:**
-Modelo não conseguiu passar da paridade, devido ao método de recompensa utilizado.
-Quando o modelo ficou no mesmo nível da baseline, o LLM Juiz não consegue dar um sinal claro e distinguível de como melhorar.
-Literatura existente faz de um modo diferente, usando logits (probabilidade do modelo de retornar cada token). **RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback** (Lee et al 2024) introduz o método d-rlaif, mas não pode ser usado devido a necessidade de rodar o modelo localmente para pegar os logits.

***Solução:**
-Usar vários modelos diferentes para julgar, em vez de um só. Deixa o sinal mais forte.
-Em vez de usar como baseline o próprio modelo (Gemma 4b), usar dois modelos, um mais fraco (Llama 1b) e um mais forte (Kimi K2). Recompensa = Mover mais próximo da baseline forte, e mais longe da baseline fraca.
-Estou rodando com essa configuração e já consegui passar da paridade. Modelo treinado já está ficando melhor que o base.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Continuar nesse processo de teste, análise e iteração. Uma dessas runs será a oficial. (Já parece estar muito próximo, talvez a atual já seja)
- Caso a próxima run seja a que der certo, a partir daí, realizar experimentos com o modelo resultante.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: **6 de nov. de 2025**

Participantes da Entrega **[matriculados em Residência em IA]**:

Pedro Schindler Freire Brasil Ribeiro

Entrega: **[descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]**

Durante a Semana, foram realizadas as seguintes atividades:
***Tema:** RL para Aprimoramento de Capacidades Fundamentais de LLMs
***Objetivo:** Treinamento de um modelo usando RL para decomposição automática de prompts (Prompt Chaining)

***Resumo da Semana:**
-Depois de treinar o modelo com a configuração descrita na semana passada (Duas baselines, uma fraca e uma forte), peguei esse modelo forte e coloquei ele contra o modelo original do gemma. Resultados foram muito bons, com um win_rate em torno de 70 a 80%.
-Link da imagem dos gráficos de treinamento das principais runs: https://photos.app.goo.gl/zvZnXStkijwgQjDo6

-Ainda assim, não fiquei satisfeito com os resultados e busquei idealizar alternativas pra melhorar o treinamento:
-Teste 1: Fixar o score da baseline do modelo original em 0, para o juiz saber que essa é a base e usar a diferença de scores como recompensa pro modelo treinado, pra maximizar essa diferença diretamente.
-Teste 2: Já que o teto de treinamento até o momento foi a baseline forte, e já estou usando praticamente o melhor modelo como baseline forte, a ideia aqui é usar o plano gerado pelo modelo treinado para gerar a baseline forte também (usando como executor o modelo forte), assim, tanto a baseline forte quanto o modelo treinado vão sendo aprimorados durante o treinamento, continuamente elevando o teto.

***Idealização do experimento final:**
-Depois de realizar os testes acima, vou pegar o que deu melhor resultado entre todos os testes feitos nas semanas anteriores e realizar um experimento final pra comprovar que os ganhos de qualidade vem de uma nova fronteira de treinamento, diferente de treinar o modelo diretamente para gerar melhores histórias, com reasoning ou etc.
-A ideia é treinar o gemma diretamente para gerar melhores histórias, treinando-o para fazer reasoning, e em uma chamada de API apenas, gerar uma melhor história (O modo normal atualmente de se treinar o modelo). Depois disso, comparar com um juiz: 1. Histórias geradas com o plano do modelo treinado, usando como executor o gemma original. 2. Histórias geradas com o plano do modelo treinado, usando como executor o gemma treinado com reasoning. 3. Histórias geradas diretamente com o gemma treinado com reasoning.
-Se o 2 tiver resultados melhores que o 1 e o 3, então os ganhos advindos do meu treinamento são ganhos novos e adicionais, indo além do que é possível hoje.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

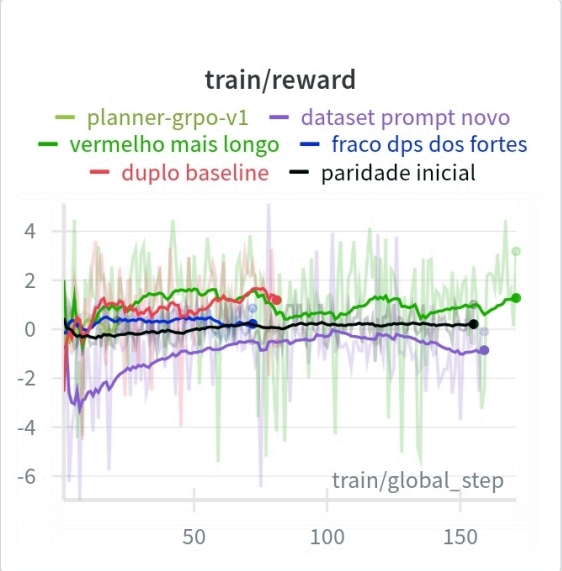
- Rodar testes 1 e 2
- Rodar experimento final, usando como modelo o melhor dos testes feitos nas Semanas.

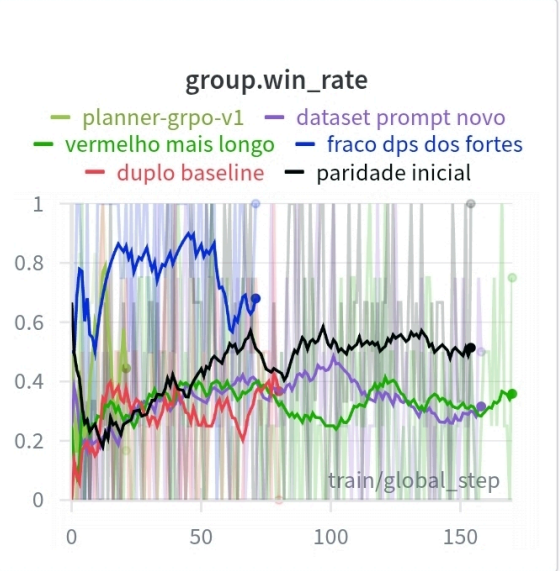
Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

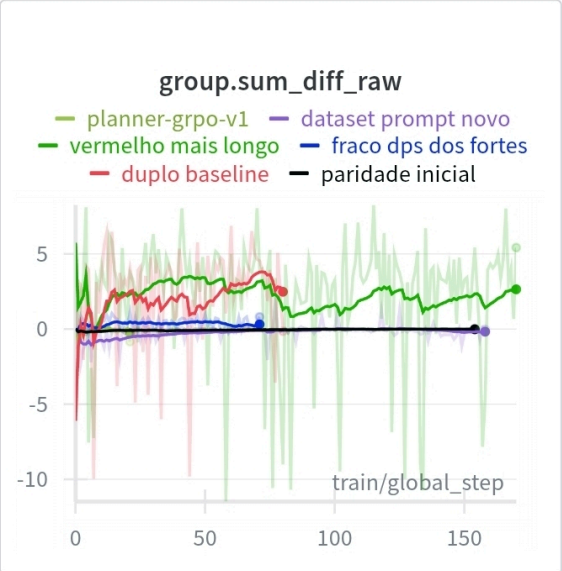
ACEITE DA ENTREGA:

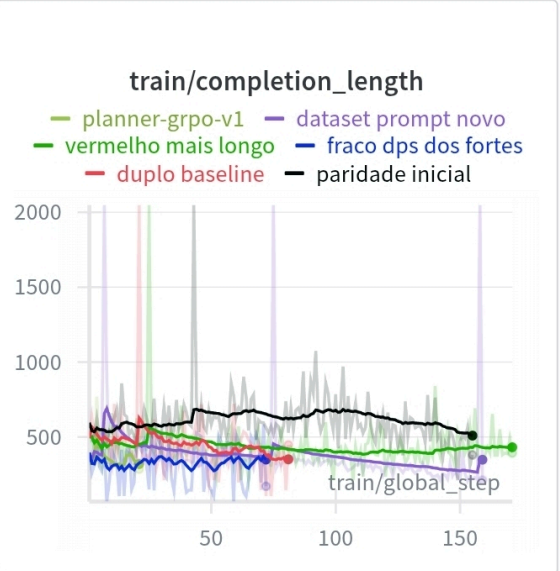
CEDRIC LUIZ DE CARVALHO: Go!

Imagem presente no link colocado no termo









APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 13 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Schindler Freire Brasil Ribeiro

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante a última Semana, foram realizadas as seguintes atividades:
***Tema:** RL para Aprimoramento de Capacidades Fundamentais de LLMs
***Objetivo:** Treinamento de um modelo usando RL para decomposição automática de prompts (Prompt Chaining)

***Resumo da Semana Anterior (Não pude apresentar):**
-Consegui resultado positivo em uma das runs. Modelo treinado gera histórias melhores do que a do modelo original na maioria dos casos, e tem paridade com o kimi-k2 (Um dos melhores modelos de creative writing atualmente)
-Não fiquei totalmente satisfeito e fui atrás de idealizar mudanças e testes pra obter resultados melhores.
-Também idealizei o experimento final pra testar se os ganhos obtidos com o treinamento/Prompt chaining, são novos e únicos, ou seja, indicando uma nova fronteira para o treinamento de modelos melhores. Esse experimento compara o prompt chaining com e sem reasoning, e chamada direta com e sem reasoning. Se o prompt chaining com reasoning for melhor que o prompt chaining sem reasoning, e melhor que chamada direta sucesso.

***Resumo dessa Semana:**
-Realizei o experimento final descrito acima.
-A ideia original seria usar o gemma 4B mesmo, pois foi o modelo que eu usei para treinamento durante a residência, porém, como esse modelo não tem modo reasoning, eu teria que treinar isso, e acabou não dando tempo. Por isso usei o qwen 3, que tem ambos os modos.
-Resultado: https://photos.app.goo.gl/YxmMZgcQjPxQ3UoV6
-Em resumo: Prompt chaining com reasoning teve resultado melhor que o prompt chaining sem reasoning (Marginalmente). Chamada direta com reasoning teve resultado melhor que chamada direta sem reasoning (Marginalmente). Porém, ambas as chamadas diretas foram melhores que o prompt chaining, contrariando o resultado obtido na semana anterior.
-Conclusão:
1. Sucesso, no sentido em que o prompt chaining com reasoning foi melhor do que o prompt chaining sem reasoning. Isso indica que o método pode trazer ganhos novos e adicionais em cima de técnicas atuais como reasoning.
2. Chamadas diretas foram melhores do que o prompt chaining, independente de reasoning. Isso é surpreendente aqui pois é o contrário do resultado obtido e reportado na semana anterior, mas isso tem um

motivo claro: O modelo pra criar os prompt chaining foi treinado com outro modelo sendo o executor, o gemma, mas alterando o modelo, o resultado foi perdido. Isso traz uma boa descoberta: Modelos treinados para criar o chaining são sensíveis e se adaptam ao modelo usado para executar os prompts, mudar o modelo necessita de retreinamento do modelo planejador de prompt chaining.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Future work:

- Refazer o experimento descrito acima, mas agora usando o mesmo modelo usado de executor no treinamento, e realizar o treinamento do reasoning do zero para ter uma comparação mais justa.
- Expandir para todos os domínios relevantes: Código, matemática, lógica, etc.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Estou sem voz porque peguei intoxicação alimentar e alergia a algo que comi durante minha viagem. E piorado pela temperatura congelante de lá. Minha garganta quase fechou, estou melhor, mas ainda sem voz.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Imagem presente no link colocado no termo