

UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

# Alinhamento de LLMs via Aprendizado por Reforço

Avaliação de Métodos de Preferência Humana

Artur Matos A. Novais



**UFG**

UNIVERSIDADE  
FEDERAL DE GOIÁS

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

ARTUR MATOS ANDRADE NOVAIS

## **Alinhamento de LLMs via Aprendizado por Reforço**

Avaliação de Métodos de Preferência Humana

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): ARTUR MATOS ANDRADE NOVAIS

Título do trabalho: Alinhamento de LLMs via Aprendizado por Reforço

Avaliação de Métodos de Preferência Humana

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

[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 **Artur Matos Andrade Novais, Discente**, em 04/02/2026, às 16:19, 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 11:20, 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](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **5956288** e o código CRC **9FE0A910**.

---

Referência: Processo nº 23070.005479/2026-90

SEI nº 5956288

ARTUR MATOS ANDRADE NOVAIS

**Alinhamento de LLMs via Aprendizado por Reforço**  
Avaliação de Métodos de Preferência Humana

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.

NOVAIS, ARTUR MATOS ANDRADE  
Alinhamento de LLMs via Aprendizado por Reforço [manuscrito]:  
Avaliação de Métodos de Preferência Humana / ARTUR MATOS ANDRADE  
NOVAIS. - 2025.  
100 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. Modelos de Linguagem. 3. Aprendizado por  
Reforço.

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

CDU 004

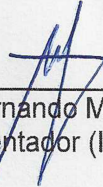
ARTUR MATOS ANDRADE NOVAIS

## Alinhamento de LLMs via Aprendizado por Reforço

Avaliação de Métodos de Preferência Humana

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)



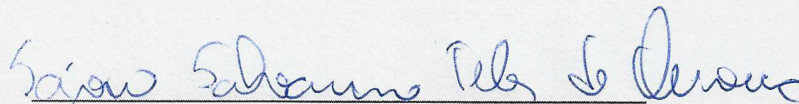
---

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)

ARTUR MATOS ANDRADE NOVAIS

## **Alinhamento de LLMs via Aprendizado por Reforço**

Avaliação de Métodos de Preferência Humana

### **RESUMO**

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Alinhamento de 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; Modelos de linguagem; Aprendizado por reforço.

### **ABSTRACT**

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

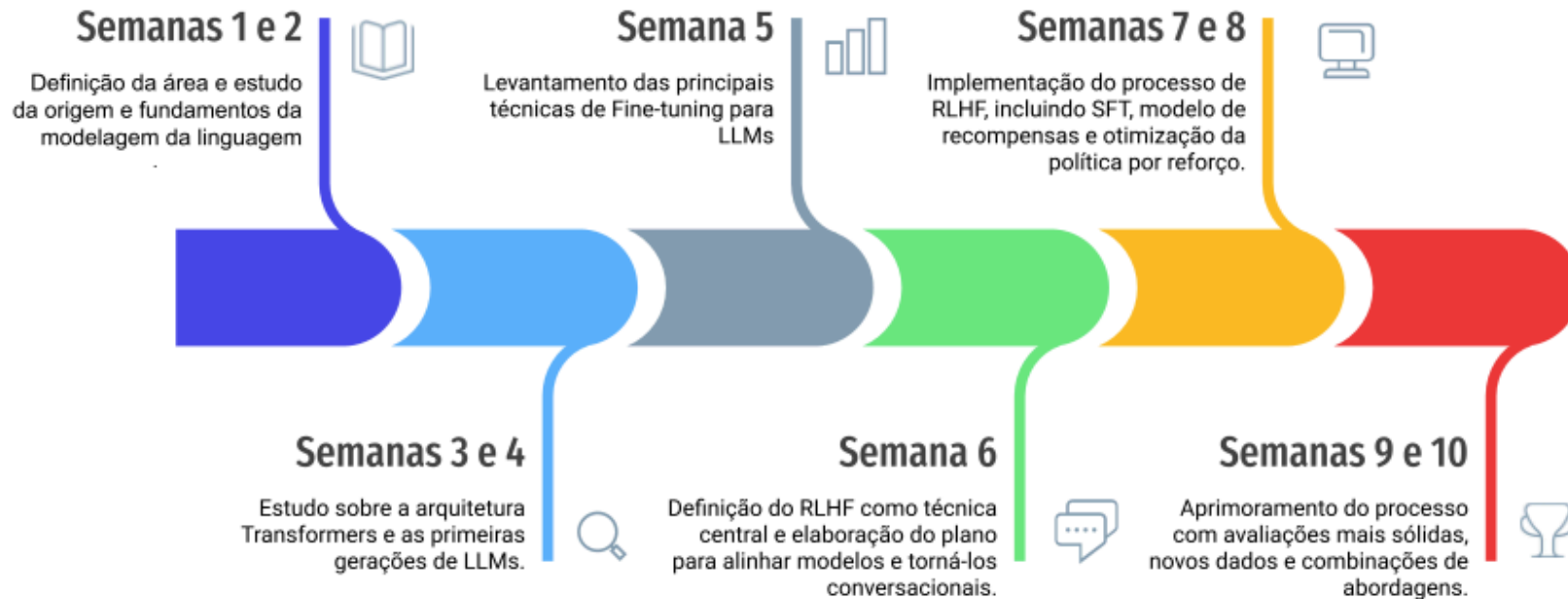
Keywords: Artificial intelligence; Language models; Reinforcement learning.

Goiânia  
2025

# Minha Jornada

Artur Matos Andrade Novais

Especialista em: Alinhamento de LLMs



---

## MINHA JORNADA

**Nome:** Artur Matos Andrade Novais

**Especialidade:** Alinhamento de LLMs

### Objetivo deste documento

Durante o processo da disciplina Residência em IA<sup>1</sup>, 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**, quando realizei um estudo sistemático das áreas de aplicação da Inteligência Artificial a partir da classificação do CSCE 2025 e, após essa exploração, escolhi inicialmente **MLOps** como área de interesse, estudando sua história e surgimento. No entanto, reflexões anteriores sobre linguagem e representação do mundo acabaram me levando a redefinir o foco para **Modelagem de Linguagem**, reconstruindo, com base em referências descritas no **Apêndice 1**, a trajetória inicial da área, das primeiras abordagens probabilísticas, como também, o papel de Markov, Shannon e Chomsky. Na **Semana 2**, levei essa base teórica para a prática ao implementar um experimento inspirado nos estudos de Shannon com textos em inglês e, em seguida, estudei os primeiros modelos neurais de linguagem e arquiteturas recorrentes, o que ampliou minha compreensão sobre contexto, memória e transformação de sequências ao longo do tempo. As análises históricas, a motivação pela área, o experimento implementado e as revisões produzidas ao longo dessas **Semanas 1 e 2** estão organizados no **Apêndice 1**.

---

<sup>1</sup> Dez Semanas, entre setembro de 2025 e dezembro de 2025.

Após entender um pouco mais sobre a história da área e o avanço inicial dos modelos neurais nas **Semanas 1 e 2**, durante a **Semana 3**, aprofundi os estudos em representação semântica da linguagem, explorando trabalhos que consolidam a ideia de representar palavras em espaços vetoriais contínuos e mostram como relações de similaridade e composição emergem de grandes volumes de texto não rotulados; em seguida, retomei o eixo da geração de texto com redes neurais, estudando arquiteturas *encoder-decoder* com mecanismos de atenção e usando essa base para analisar a arquitetura *Transformer*, em especial o funcionamento da autoatenção, a possibilidade de paralelização e o papel das codificações posicionais. Já na **Semana 4**, foquei nos primeiros modelos de linguagem baseados em Transformers, como o *BERT*, voltado à construção de representações bidirecionais reutilizáveis em diferentes domínios, e a linha *GPT* (*GPT-1*, *GPT-2* e *GPT-3*), que explora o pré-treinamento em larga escala e evidencia capacidades zero-shot e few-shot. A revisão dessa sequência de trabalhos, junto à atualização do diagrama cronológico da área, tornou mais nítido o caminho que leva dos modelos estatísticos aos *LLMs* atuais e, sobretudo, reforçou meu fascínio pelos modelos conversacionais capazes de seguir instruções e sustentar diálogos mais naturais, o que levou à definição de adaptação e alinhamento de modelos de linguagem baseados em *Transformer* como próxima trilha de estudo, apoiada em referências recentes que, juntamente com os materiais trabalhados nessas **Semanas 3 e 4**, estão organizadas no **Apêndice 2**.

Na **Semana 5**, a partir da trilha de adaptação e alinhamento de modelos de linguagem definida nas **Semanas 3 e 4**, aprofundi o estudo das técnicas que sustentam essa trajetória, explorando especificamente a etapa de pós-treinamento de *LLMs* com a leitura do survey “*A Survey on Post-Training of Large Language Models*”, que sintetiza os principais paradigmas de ajuste fino, alinhamento, raciocínio, eficiência e adaptação até 2025. Com isso, pude consolidar uma visão mais estruturada sobre o papel do pós-treinamento na transição de modelos “apenas pré-treinados” para sistemas realmente conversacionais e capazes de incorporar preferências humanas, com destaque para métodos baseados em aprendizado por reforço, como *RLHF* e abordagens de otimização direta de preferências. Ao final dessa **Semana 5**, meu interesse se voltou de forma mais clara ao eixo de alinhamento por reforço, não por uma inclinação prévia específica à área, mas pelo reconhecimento de

que grande parte das abordagens de alinhamento atuais se apoia nesse tipo de técnica. O trabalho realizado nesta **Semana 5** está melhor descrito no **Apêndice 3**.

Na **Semana 6**, aprofundi a trilha de adaptação e alinhamento iniciada nas **Semanas** anteriores ao estudar em detalhe o artigo *“Training Language Models to Follow Instructions with Human Feedback”*, que popularizou o *RLHF* e mostrou, de forma concreta, como modelos relativamente pequenos podem superar arquiteturas muito maiores quando são bem alinhados às instruções humanas. A partir dessa leitura, defini como objetivo prático replicar o processo descrito no artigo, escolhendo o modelo *Gemma 3 1B base* como ponto de partida e realizando testes iniciais que evidenciaram sua dificuldade em seguir instruções antes de qualquer forma de pós-treinamento. Isso me levou a aprofundar o estudo dos métodos de aprendizado por reforço aplicados ao alinhamento, com foco em abordagens como *PPO* e *DPO* e em materiais que discutem boas práticas para aprendizado a partir de feedback de preferência, bem como a mapear frameworks capazes de suportar esse tipo de experimento. Ao comparar alternativas como *TRL*, *OpenRLHF* e *NeMo-RL*, optei por adotar inicialmente o *TRL*, tanto pela integração ao ecossistema Hugging Face quanto pela disponibilidade de documentação e cursos, consolidando ao final desta **Semana 6** um plano mais concreto para implementação do *RLHF* sobre o modelo escolhido. O trabalho realizado nesta **Semana 6** está descrito em maior detalhe no **Apêndice 4**.

Nas **Semanas 7 e 8**, avancei na replicação do fluxo de *RLHF* descrito em *“Training Language Models to Follow Instructions with Human Feedback”*, com o objetivo de percorrer de forma prática as três etapas do processo: treinamento supervisionado da política, treinamento do modelo de recompensa e otimização da política por reforço. Na **Semana 7**, realizei o *SFT* do modelo *Gemma 3 1B base* para trazer a capacidade de responder instruções ao modelo e também para servir como política inicial, treinei um modelo de recompensa e implementei o script que utiliza esse modelo para avaliar automaticamente os checkpoints da política e selecionar aquele com a maior recompensa média em um conjunto de dados estabelecido. Todas as implementações e etapas, incluindo a criação do conjunto de dados utilizado, foram conduzidas com o objetivo de replicar da forma mais fiel possível o cenário descrito no artigo. Já na **Semana 8**, foquei em integrar esses componentes em um

fluxo único de otimização por reforço, conectando política supervisionada e modelo de recompensa em um script de treino com a técnica de *GRPO*, incluindo a configuração da *penalização por divergência de KL*. Nesse mesmo período, iniciei também um processo de avaliação externa utilizando *benchmarks* padronizados, que evidenciou o desafio de definir uma metodologia realmente consistente para analisar os resultados produzidos em cada uma das etapas. O trabalho desenvolvido nessas **Semanas 7 e 8** está descrito em maior detalhe no **Apêndice 5**.

Nas **Semanas 9 e 10**, concentrei-me em revisar criticamente o experimento de alinhamento e em estruturar um sistema de avaliação coerente com o descrito no artigo *“Training Language Models to Follow Instructions with Human Feedback”*. Na **Semana 9**, diante da dificuldade em interpretar os resultados anteriores, reli o artigo com atenção voltada especificamente à forma como os autores avaliam seus modelos e, a partir disso, reorganizei meu esquema de avaliação para dar prioridade à capacidade de seguir instruções, à veracidade e à toxicidade, combinando benchmarks como *TruthfulQA* e *IFEval* com avaliações automáticas e uma etapa de avaliação cega com anotadores humanos. Também construí um novo conjunto de dados mais diverso e próximo da distribuição relatada no trabalho original e refiz o fluxo de *RLHF* com essa base. Na **Semana 10**, com a avaliação mais bem definida e após corrigir um erro de implementação na etapa de reforço, passei a testar sistematicamente diferentes combinações de técnicas e dados, comparando configurações como apenas *SFT*, *SFT* seguido de *GRPO*, *GRPO* combinado com *DPO* e variações no uso dos conjuntos de preferência. A melhor configuração obtida ao final desse ciclo combinou otimização por reforço e otimização direta de preferências, resultando em uma melhora de aproximadamente **47%** na capacidade de seguir instruções, **29%** em veracidade e redução adicional na toxicidade, mesmo sem instruções explícitas para isso, superando de forma consistente o modelo apenas com *SFT*. O trabalho desenvolvido nessas **Semanas 9 e 10** está descrito em maior detalhe no **Apêndice 6**.

Em função de tudo que vivi nesta Jornada, percebo que esse percurso foi muito além de um recorte técnico em Modelagem de Linguagem: ele me mostrou a importância de voltar à base histórica e conceitual, de enxergar que em cada nível de conhecimento sempre há

novas camadas a explorar e de me aproximar, de fato, do processo científico. Comecei sem um caminho claramente definido, apenas com a vontade de entender o que significa modelar a linguagem, com o tempo, esse interesse se transformou no desejo de fazer uma máquina conversar e seguir instruções, o que me levou ao aprendizado por reforço, área em que eu tinha conhecimento limitado e até certo “medo”. Hoje, fico genuinamente feliz por ter enfrentado essas barreiras, por conseguir navegar com mais segurança pelos conceitos e técnicas da área e, principalmente, por ter desenvolvido um interesse real e duradouro pelo tema, algo que considero um dos principais resultados pessoais dessa Jornada.

Por fim, gostaria de agradecer imensamente aos professores que acompanharam este processo, em especial o Professor Fernando Federson, o Professor Cedric Luiz e o Professor Leonardo Alves. Sou profundamente grato por ter sido aluno de vocês tanto no início quanto na conclusão da graduação, certamente não sou mais a mesma pessoa que entrou na universidade, e cada um de vocês contribuiu para essa transformação. Em especial, agradeço ao Professor Fernando Federson por todo o carinho, pela dedicação e por todos os ensinamentos ao longo dessa Jornada. É alguém por quem tenho enorme respeito e admiração, e em quem me espelho também como pessoa. Gostaria também de deixar um agradecimento especial ao Professor Sávio Teles que, além de tudo o que me ensinou, foi decisivo para despertar, em um de seus grupos de estudos, o meu interesse pela área que hoje escolhi estudar com mais profundidade, tendo um papel muito importante não apenas na definição do meu foco de estudo, mas também no meu desenvolvimento pessoal ao longo desse percurso. Agradeço ainda à minha família, que apoiou minhas escolhas, me incentivou a mudar de cidade para cursar a graduação e esteve ao meu lado em cada decisão importante. Aos meus amigos, que considero parte da minha família, deixo um agradecimento especial por terem sido apoio e companhia em noites mal dormidas, prazos apertados e desafios que enfrentamos juntos, no fim, sinto que crescemos juntos, e isso torna esta Jornada ainda mais significativa para mim.

## 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:** 2 de set. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Para a primeira Semana do processo da disciplina Residência em IA foi feito:

- Estudo de interesse sobre as áreas de aplicação segundo CSCE 2025:
  - Seleção de todos os temas de interesse
  - Busca e entendimento dos temas “desconhecidos”
  - Agrupamento de tópicos em grandes áreas
- Definição de uma grande área a ser trabalhado durante o Processo da Disciplina:
  - MLOps (Machine Learning Operations)
    - Estudo da história e surgimento da área: [História e surgimento MLOps](#)
- Redefinição da área a ser trabalhada durante o Processo da disciplina
  - Language Modeling (representar matematicamente a organização da linguagem natural)
  - [Motivação para escolha da área](#)
- Investigação sobre a história e o surgimento da área, com foco na concepção inicial da ideia.
  - Definição e levantamento de artigos que abordam uma visão histórica da área
  - Estudo das obras de Claude Shannon: informação, entropia e experimentos de previsão.
  - [Documento que detalha minha jornada no estudo da história e surgimento da área](#)

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

**Implementar o experimento de Shannon com n-grams estatísticos e comparar com os resultados originais.**

**Ler e analisar o artigo “A Neural Probabilistic Language Model” com mais de 12 mil citações e inaugural no uso modelos neurais para language modeling.**

---

Elaborar uma linha do tempo com marcos históricos e a evolução do *language modeling*.

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:** 10 de set. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante a Semana 1, foi definido o Tema de Language Modeling, onde após explorar um pouco da história e surgimento da ideia foram realizados os seguintes passos na Semana 2:

Criação de um código para replicar o experimento de Shannon

- Foram utilizados livros em inglês para o experimento e o resultado foi extremamente próximo do relatado. [Link para o repositório com o código](#)

Avanço e aprofundamento nas **técnicas neurais** de Language Modeling:

- Leitura do artigo [A Neural Probabilistic Language Model](#) e organização de uma revisão a partir das minhas anotações. [Link para acessar a revisão](#)
  - Nesse processo de leitura descobri que se trata de um dos primeiros artigos a abordar diretamente a ideia de **representações densas vindas das redes neurais**
  - Ao pesquisar referências sobre o artigo, encontrei um [post no Medium](#) com o modelo e o código; a partir dele, adaptei e montei um treino usando os mesmos dados do experimento de Shannon. [Link para o repositório com o código](#)

Estudo de técnicas utilizando **redes recorrentes** que antecedem os mecanismos de atenção e arquitetura transformers

- Leitura do artigo [Recurrent neural network based language model](#) e organização de uma revisão a partir das minhas anotações. [Link para acessar a revisão](#)
- Leitura do artigo [Sequence to Sequence Learning with Neural Networks](#) e organização de uma revisão a partir das minhas anotações. [Link para acessar a revisão](#)

Início de um mapa com linha do tempo contendo os marcos históricos e a evolução do language modeling: [link do mapa](#).

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

Explorar o desenvolvimento da representação de linguagem a partir do artigo [A Neural Probabilistic Language Model](#), um dos primeiros a destacar representações densas vindas das redes neurais.

Explorar e estudar os mecanismos de atenção que antecedem a arquitetura transformers.

Leitura do Artigo "[attention is all you need](#)", artigo mais citado da área

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

---

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

## Motivação para escolha da área

No livro “O andar do bêbado”, o autor discute que embora os gregos possuíam uma forte tradição matemática e refletiram sobre jogos de azar, mas não desenvolveram a teoria da probabilidade de forma estruturada. O livro destaca que a teoria da probabilidade só começou a ganhar forma por volta do século XVII, principalmente com Pascal, Fermat e Huygens. Como um dos fatores que contribuíram para isso, o livro traz a **notação decimal** trazida da Índia/árabes para a Europa, que simplificou enormemente os cálculos. Antes disso, com números romanos, era quase inviável manipular grandes quantidades de forma eficiente.

Com isso, podemos trazer a seguinte discussão: “O fenômeno que chamamos de probabilidade se modificou ao longo do tempo ou apenas modificamos a interface (linguagem) a qual representamos o problema?”

Desse modo, entendo que a linguagem é “apenas” uma interface que criamos para representar fenômenos do mundo que são invariantes a mesma. Ainda assim, para mim sempre foi muito interessante pensar o quanto a linguagem consegue representar o mundo, lembro que desde muito novo, quando ganhei do meu pai o livro “O homem que calculava”, onde me apaixonei pela matemática em si e comecei a pensar na matemática como uma linguagem e em todas as questões citadas acima.

Mais recentemente com o advento dos LLMs e minha exposição à área de inteligência artificial comecei a pensar nesse conceito de linguagem e o quanto disso conseguimos replicar de maneira matemática com a computação. Essa reflexão me leva diretamente à minha escolha em me aprofundar na área de language modeling, a qual tenho um grande interesse de estudar sob a ótica descrita acima.

## História e surgimento da área de Language Modeling

**Motivação:** Acredito que para me tornar um especialista no tema tenho que dominar história, fundamentos e ferramentas da área. Nesse percurso, o primeiro passo é mergulhar na história, entendendo como surgiu a concepção da ideia e em que momento começaram as discussões que até hoje sustentam e orientam esse campo.

**Metodologia:** Inicialmente pesquisei alguma referência que destacasse a história da área de uma maneira “linear”, não para servir como fonte acadêmica definitiva, mas como um ponto de partida para entender a trajetória e expandir a pesquisa direcionando-a para os termos e termos encontrados.

**Referência inicial:** A referência escolhida foi [Language Models: Past, Present, and Future - A language modeling overview](#) que é um artigo publicado em 2022 da revista *Communications of the*

ACM uma das influentes organizações internacionais em computação. O autor, Hang Li, é Head de pesquisa na Byte Dance, empresa conhecida mundialmente por produtos como Tik Tok. Além disso, o autor é ACM Fellow, um reconhecimento concedido pela ACM a profissionais com contribuições excepcionais e também Fellow da ACL (Association for Computational Linguistics) e da IEEE, o que reforça alguma “autoridade” sobre o tema de pesquisa.

**Definição:** Antes de tudo, a fonte define que “Language models amount to important machinery for modeling natural language texts based on probability theory, statistics, information theory, and machine learning”, o que pode ser interpretado no contexto de language modeling como a formulação matemática voltada a capturar regularidades e estruturas da linguagem natural.

**Conteúdo:**

O artigo começa destacando que existem duas abordagens principais para modelar linguagens humanas usando meios matemáticos: uma baseada na **teoria da probabilidade** e a outra na **teoria formal da linguagem**. Juntamente com essa classificação são citados 3 nomes, dois relacionados a teoria de probabilidade (Markov e Shannon) e um relacionado a teoria formal da linguagem (Chomsky)

**Teoria da probabilidade:** Markov (1913) - Claude Shannon (1948)

**Teoria formal da linguagem:** Chomsky (1956)

**Markov:**

O artigo começa destacando o trabalho de Markov e trazendo-o como possivelmente o primeiro cientista a estudar modelos de linguagem (o termo não existia na época).

Segue uma análise de Markov no contexto do artigo e minha visão sobre: [Markov no contexto de Language Modeling](#)

**Claude Shannon:**

Embora o artigo destaque Markov como possivelmente o primeiro cientista a estudar modelos de linguagem, em meus estudos ficou claro o salto que tivemos com Claude Shannon, que observou e estudou o tema sobre uma ótica diferente, em vez de “apenas” mostrar que existiam dependências entre símbolos, Shannon quis medir o quão estruturada a linguagem realmente é.

## Claude Shannon no contexto de Language Modeling

**Shannon no artigo Language Models: Past, Present, and Future**

Quando vamos para Shannon, vemos o trabalho intitulado de “[The Mathematical Theory of Communication](#)”, um nome bem mais atraente com a área de estudo e que fundou toda uma área

chamada teoria da informação. Shannon deu um passo além de Markov que “apenas” mostrou que existem dependências, ao propor uma forma de quantificar a incerteza da linguagem. Além disso, ele trouxe o conceito de entropia cruzada, que avalia o quanto um modelo de linguagem se aproxima da distribuição “real” dos dados. Além disso, ele trouxe um método matemático para medir qualidade de modelos: quanto menor a entropia cruzada, melhor a capacidade de previsão. Esse conceito foi trazido por Shannon em 1948, e o perceptron surge após 1950, ou seja, novamente o conceito que conhecemos vem antes do meio normalmente encontramos associado a esse conceito.

### **Minha visão:**

Em algum momento Shannon introduz e descreve melhor os conceitos de entropia e em um de seus teoremas ele diz que a “*entropia verdadeira da língua é sempre o limite inferior do quanto conseguimos modelar essa língua com qualquer modelo aproximado*”, Isso significa que nenhum modelo será melhor que a própria distribuição verdadeira da linguagem que ele aproxima, ou seja, o modelo está limitado por si só a maneira a qual a língua em si funciona, o que parece “óbvio”, mas com isso posto, podemos agora observar todo o processo de “busca por inteligência” sob outra ótica, por que se o modelo está limitado a própria língua, a que a língua está limitada? Qual o limite de conhecimento possível que podemos representar usando a linguagem? Existe algum fenômeno que não é representável através da linguagem? Isso me parece entrar na discussão de se quem veio primeiro foi o ovo ou a galinha, pois por mais que de alguma forma aquele conhecimento, observação... exista independente da linguagem, o quão externa a linguagem é em relação ao mesmo? Me parece a mesma questão quando olhamos para a matemática, que de alguma forma pode ser vista como uma linguagem, o fenômeno existe antes da linguagem, mas de quantas maneiras distintas podemos representá-lo? Será que existem representações melhores do que as outras? Por exemplo, a própria teoria de probabilidade foi se desenvolvendo melhor com a mudança da linguagem matemática para o sistema de algarismos atual, visto que o sistema de algarismos romanos não favorecia operações aritméticas, que são extremamente necessárias na teoria de probabilidade. Me parece que constantemente olhamos para a linguagem e aquilo que ela representa como uma coisa só, quando na verdade talvez existam fenômenos distintos e a linguagem seja o meio usado para alcançar o “fenômeno final”, onde para toda essa área precisamos de alguma forma nos “limitar” a esse meio.

Em algum ponto da leitura sobre Shannon, buscando entender um pouco mais sobre a sua história e contribuição, me deparei com um texto onde são descritos alguns dos experimentos de Shannon que me chamaram atenção.

Contudo, o texto que encontrei referenciava um ensaio publicado na revista The Hedgehog Review intitulado de “Language Machinery Who will attend to the machines’ writing?” Decidi então ler esse texto e confesso que foi uma das referências que mais fez sentido para mim nesse entendimento do surgimento da área dado a ótica de observação do autor sobre o tema. O interessante é que o texto está relacionado com os temas propostos no artigo inicial, citando Markov, Shannon e Chomsky, porém com um maior nível de aprofundamento. Segue uma análise sobre o trabalho:

[Análise de Language Machinery Who will attend to the machines’ writing.](#)

No fim, onde gastei mais tempo foi na análise do trabalho de Shannon, que é onde considero que de fato surgiu a mentalidade descrita da área de language modeling. O ensaio traz **Shannon como o primeiro modelador estatístico de linguagem**, onde demonstrou que o uso da linguagem podia ser descrito por estatísticas e imitado por elas, ponto que concordo, principalmente apoiado em sua experimentação. Para o estudo sobre Shannon foram consultados (não necessariamente lidos integralmente), seus dois artigos:

[A Mathematical Theory of Communication](#)  
[Prediction and Entropy of Printed English](#)

### **Chomsky e a teoria formal da linguagem**

Chomsky assume um paradigma diferente, na ideia que os modelos de linguagem baseados em n-gramas são limitados, porque funcionam como gramáticas de estados finitos incapazes de capturar a complexidade infinita das línguas naturais. Chomsky ainda ficou conhecido por organizar a **teoria formal da linguagem**.

Quando Chomsky “critica” os modelos baseados apenas em n-gramas, o que ele está dizendo, no fundo, é que a linguagem humana não é só uma sequência de palavras encaixadas estatisticamente. Ela tem estrutura, hierarquia e, por consequência, está ligada ao significado. Uma frase como “se chover, então...” carrega uma relação lógica que pode se repetir e se encaixar infinitamente. Para Chomsky, isso mostrava que a linguagem não é só forma, mas também a capacidade de expressar relações de sentido que ultrapassam qualquer sequência de estados finitos.

Na visão de Chomsky então a linguagem, por sua própria natureza, precisa de uma estrutura capaz de preservar as condições que dão origem ao significado.

### **Avanço da área entre 1970 a 1980**

A referência principal [Language Models: Past, Present, and Future](#) pula direto para modelos neurais por volta de 2001, contudo, em pesquisa, vi que por volta dos anos 1970 surgem os primeiros modelos estatísticos relevantes que usam n-gram, alinhando a ideia de Markov e Shannon onde um deles teve destaque em 1980 proposto pela IBM, ainda utilizando modelos n-gram. Até então n-grams puramente estatísticos (a partir de contagem) foram padrão na área onde ao longo dos anos o avanço estava mais centrado em dados e poder computacional do que nas técnicas em si.

## **Estudo dos primeiros modelos neurais de linguagem**

### **A Neural Probabilistic Language Model**

**Motivação:** Uma das coisas percebidas durante a implementação de um código para replicar o experimento de Shannon sobre os n-grams é que a medida que aumentamos o n, o tempo aumenta de

maneira não linear, por que está disposto conforme a combinação do vocabulário e o valor de  $n$ . Isso é citado pelo artigo como um motivador para o trabalho, citando as redes neurais como uma solução para esse problema.

**Conteúdo:** Um dos pontos trazidos pelo artigo inicialmente é sobre a limitação dos modelos  $n$ -gram estatísticos:

“Quando você usa  $n$ -grams, cada posição pode ser qualquer palavra do vocabulário. Então, ao aumentar o  $n$ , você multiplica o número de sequências possíveis por todo o tamanho do vocabulário mais uma vez”

Além disso, por exemplo, com um vocabulário de cinquenta mil palavras, o total de combinações quando você passa de 3gram para 4gram vai de algo na ordem de bilhões para algo na ordem de centenas de trilhões. Fora que uma palavra modificada na sequência pode alterar literalmente todo contexto.

**Minha visão:**

*Em minha visão então, temos a hipótese de que a linguagem pode ser modelada de forma matemática, mas embora seja “possível” ter um modelo de baixa entropia com a modelagem anterior ao surgimento das redes neurais isso seria impossível de maneira prática. Isso porque, na língua portuguesa temos cerca de 370 mil palavras, e seria praticamente impossível ter um grande exemplo de sequências longas e registrar isso de alguma forma. Fora que na modelagem puramente estatística baseada em contagem, um único sinônimo pode alterar toda ordem e ser vista como algo novo, da mesma forma que um antônimo seria visto. Ou seja, até o momento, pouco se pensava no significado das coisas.*

Um ponto interessante que o artigo trouxe e que nunca havia visto: De alguma maneira, quanto mais próximas as palavras em um texto mais dependentes elas são para os modelos anteriores.

O artigo traz DOIS problemas que sugere avançar:

1. Levar em consideração a similaridade entre as palavras
2. Contexto extremamente curto para bom funcionamento

**Levar em consideração a similaridade entre as palavras:**

Nesse ponto o artigo introduz a ideia de mapeamento onde cada palavra tem um **vetor de características**. Com isso, temos provavelmente o surgimento da ideia das incorporações de palavras, algo que foi extremamente relevante para o desenvolvimento da área surgindo nesse artigo. O artigo cita cada palavra como um ponto em um espaço vetorial.

Assim, resolvemos a dependência de “representar” cada palavra em função do vocabulário completo e usando um valor  $m$ , sendo  $m$ , a quantidade de características usadas no vetor para cada palavra.

**Minha visão:**

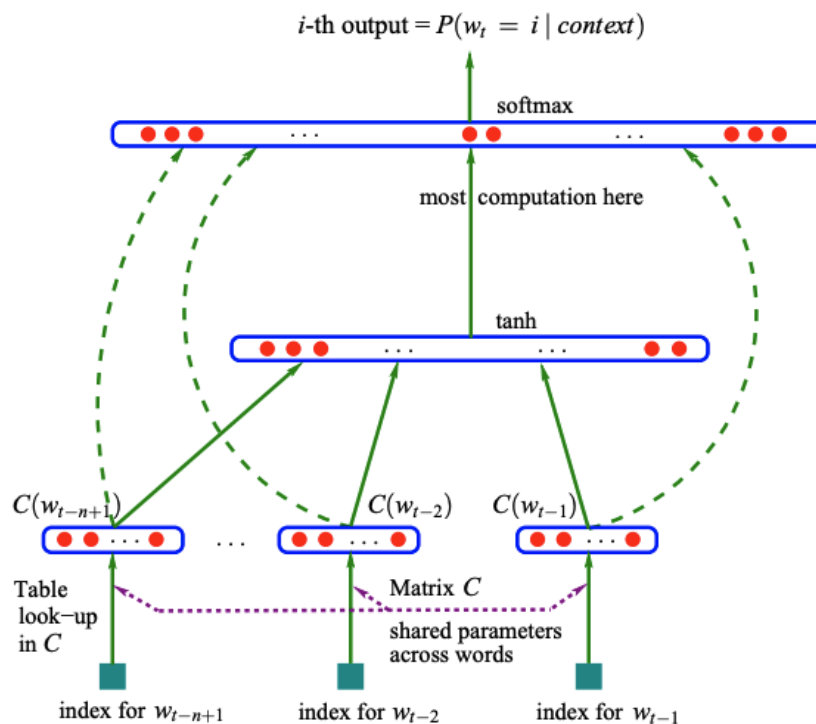
*Na minha visão a mudança não é “apenas” um incremento técnico, mas também uma mudança de paradigma, pois a partir desse ponto, quando assumimos um espaço vetorial organizado, assumimos também que as palavras terão significados relativos às demais, isso porque esses vetores possuem distâncias uns dos outros, trazendo a possibilidade de representar sinônimos, antônimos e trazer de alguma forma o significado a tona na modelagem de linguagem. Na minha visão, o trabalho é “genial” não por utilizar redes neurais e sim pela mudança de paradigma proposta.*

Ao longo do artigo descobri que a ideia de usar redes neurais na modelagem de linguagem não é nova, mas que foi a primeira vez feita dessa forma, onde o vetor de características é usado com a ideia de contextos similares e com alta escala.

É citada também que a ideia de vetores de características foi bem explorada na área de recuperação de informação. Contudo, de novo como diferencial o artigo cita a técnica usada até então e que não funcionou no caso da geração.

Aqui no artigo, temos claramente duas ramificações da modelagem de linguagem:

1. Modelagem para representação da informação
2. Modelagem para encontrar um modelo ideal onde  $P(w_t/w_{t-1}, w_{t-2} \dots)$ , ou seja, modelar a probabilidade da próxima palavra dado as  $n$  palavras anteriores, paradigma de geração esse que surge em Markov e perdura até os dias de hoje



No fim das contas o que temos é um vocabulário de tamanho  $x$  onde cada palavra será mapeada a um vetor de tamanho  $y$ , e a entrada da rede sendo  $c$  vetores  $y$  correspondente as palavras, com o alvo de predição da rede sendo a próxima palavra, para corrigir a rede faremos uma softmax no final gerando uma ideia de probabilidade para cada vetor do vocabulário, onde idealmente o vetor de maior probabilidade seria associado a próxima palavra.

Em seguida, o artigo destaca vários termos técnicos relacionados ao treinamento de redes neurais, mas não necessariamente a modelagem de linguagem em si.

Eles experimentaram o modelo proposto em mais de 14 milhões de palavras na arquitetura proposta e também nos modelos ngram estado até a época do artigo.

A forma de medição é a perplexidade onde tiveram um resultado 24% melhor que o estado da arte

Em próximos trabalhos é citado a questão de palavras fora do vocabulário e como essa abordagem poderia fazer isso, inicializando esse vetor com os vetores de palavras que poderiam ocorrer no mesmo contexto.

Outro ponto interessante trazido é sobre a interpretabilidade desses vetores.

***Minha visão:***

*O trabalho inaugura uma forma de pensar diferente, onde se observarmos bem a arquitetura proposta, veremos uma certa semelhança com o que usamos hoje em dia inclusive, visto que a entrada e saída do modelo se aproximam bastante, ou seja a forma de modelar usando redes neurais é extremamente significativa, extrapolando a arquitetura usada.*

## Recurrent neural network based language model

**Conteúdo:** Logo na introdução nos deparamos com a seguinte frase: *“The goal of statistical language modeling is to predict the next word in textual data given context; thus we are dealing with sequential data prediction problem when constructing language models”* Já nesse momento vemos a ideia de sequência no tipo de dado textual, o que de fato faz sentido e se alinha com a visão já estabelecida. Contudo, o “ouro” está no questionamento que o artigo faz ao indagar se realmente houveram avanços reais desde o surgimento dos modelos de maior prática na época), onde cita:

*“In fact, most of the proposed advanced language modeling techniques provide only tiny improvements over simple baselines, and are rarely used in practice.”*

Logo em seguida é citado o estudo anterior dos neural LM's como uma abordagem extremamente bem sucedida, reafirmando de alguma forma o quão importante foi o estudo proposto. Contudo, como

grande problema citado temos o de que as redes feed forward utilizadas necessitam de um **contexto fixo, cujo tamanho deve ser especificado antes do treino**, citando como número de 5 a 10 palavras vistas nas redes da época, citando também a capacidade humana em lidar com contextos maiores. É citado também a “temporalidade” do tipo de dado textual.

A partir disso o artigo passa a comentar sobre as redes neurais recorrentes, mas o importante para a área nesse ponto é o seguinte:

- Temos agora uma ideia de “memória”, visto que as redes neurais recorrentes um estado que é propagado a cada “passada” do modelo. Junto com isso, agora processamos o contexto de maneira dinâmica, resolvendo o problema que foi citado dos primeiros modelos neurais para modelagem de linguagem.

Dessa forma, podemos ver o método proposto modelando linguagem como um processo sequencial com memória (estado recorrente).

Como resultado é citado que em alguns experimentos, o desempenho quase “dobrou” trocando as feedforwards simples pelas redes recorrentes.

O contexto que a maioria dos testes é feito se insere na tarefa de reconhecimento de fala.

Cita-se também que os resultados foram melhores em todos os casos de teste, inclusive quando os modelos concorrentes são treinados com mais dados.

## Sequence to Sequence Learning with Neural Networks

Esse artigo foi proposto pela Google em 2014, e logo em seu início, cita os problemas do uso de redes tradicionais, assim como seu antecessor [Recurrent neural network based language model](#). O ponto é que existem tarefas que são fortemente beneficiadas pelo uso de sequências. Um exemplo que me chama atenção é a tarefa de Question and answering citada no artigo:

*“Likewise, question answering can also be seen as mapping a sequence of words representing the question to a 1 sequence of words representing the answer”*

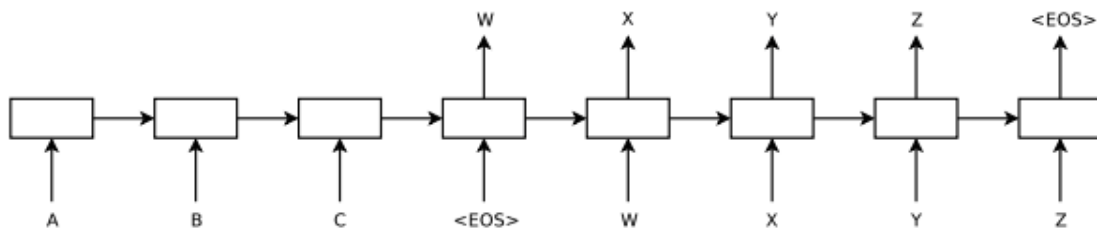
Essa visão me chama a atenção pois de certa forma essa ideia de mapeamento faz sentido, visto que a compreensão de uma pergunta é essencial para geração de uma resposta e é um processo que facilmente imagino ocorrendo com os humanos, mas nunca havia visto por essa ótica.

De cara é citado como foi atingido o melhor resultado já visto até o momento na tarefa de tradução

*“This is by far the best result achieved by direct translation with large neural networks.”*

Durante o artigo é bastante destacado questões relacionadas ao uso de LSTM em si, mas isso é “ferramental” em minha visão, sendo o importante de fato a maneira de pensar e propor uma nova arquitetura.

O que vem de “diferente” no artigo é que ao invés de usar as redes recorrentes como vimos em [Recurrent neural network based language model](#), passamos a utilizar duas redes recorrentes, que chamaremos de encoder decoder, o que parece natural para trabalharmos com um problema sequence-to-sequence onde uma das redes (encoder) vai processar a sequência de entrada e a outra rede (decoder) irá gerar a saída esperada. Naturalmente nesse tipo de caso pensamos em tradução, mas até mesmo expandido para fora do processamento de linguagem, poderíamos utilizar a arquitetura para mapeamento de maneira geral.



Um outro ponto em que parece fazer sentido é que ter a sequência completamente processada faz com que se alguma forma tenhamos isso mais “ativo” na memória da rede recorrente.

O artigo também destaca que agora tanto sequência de entrada quanto de saída podem ter tamanhos dinâmicos e utilizar de um “token” especial <EOS> para finalizar a geração, encarregando o próprio modelo de ditar o tamanho da sequência. Pensando no contexto da tradução isso faz sentido, pois o mapeamento não é 1 para 1.

Uma outra sacada inteligente do artigo e que trouxe bons resultados é inverter a ordem apenas do trecho de entrada, sob a hipótese de que o início da geração do decoder é uma espécie de “arranque” sendo bastante importante acertar para dar continuidade de maneira correta na sequência, então a palavra inicial do encoder fica mais próxima da inicial do decoder, mesmo que a distância média entre as palavras não mude. Por mais “antinatural” que pareça, a abordagem gerou bons resultados.

## 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:** 18 de set. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também os primeiros **modelos neurais** até o uso das **redes recorrentes**.

Ainda assim, na última Semana, durante a leitura do artigo “[A Neural Probabilistic Language Model](#)”, inovador ao abordar representações densas vindas das redes neurais, entendi que o artigo inspira duas frentes de trabalho:

1. **Representação semântica de texto e redes neurais**
2. **Geração de texto e redes neurais**

Nessa Semana, aprofundi o estudo em representação semântica gerando anotações sobre o tema disponíveis em [Estudo sobre representação da linguagem](#) contendo a leitura dos seguintes artigos:

- [Natural Language Processing \(Almost\) from Scratch](#)
- [Efficient Estimation of Word Representations in Vector Space](#)
- [Distributed Representations of Words and Phrases and their Compositionality](#)


Após explorar e entender melhor a área de representação semântica, voltei os estudos a Geração de texto com redes neurais focando no surgimento do conceito de atenção, produzindo um documento com minhas anotações sobre o tema: [Atenção pré transformers](#) contendo a leitura de:

- [Neural machine translation by jointly learning to align and translate](#)

Posto isso, busquei entender um pouco mais sobre o surgimento da arquitetura Transformers realizando a leitura e anotação dos seguintes trabalhos:

- [Attention is all you need](#) - [Revisão Attention Is All you Need](#)
- [The Illustrated Transformer](#) - [Revisão The Illustrated Transformer](#)

Ao longo do trabalho ficou nítido que todos os artigos lidos até o momento estavam extremamente relacionados, formando uma espécie de trilha, onde um artigo expandia a área a partir do anterior . Com isso, criei um diagrama que conecta e expõe os trabalhos ao longo do tempo.

 Diagrama Language Modeling

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

Continuar explorando a linha de representação semântica, mas usando Transformers com o artigo:  
**“BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”**

Explorar a subárea de adaptação e alinhamento de modelos de linguagem baseados em transformers, iniciando pelos artigos

- **Improving Language Understanding by Generative Pre-Training**
- **Language Models are Unsupervised Multitask Learners**

### **Observação:** [caso precise fazer alguma observação, de qualquer “natureza”]

Uma das coisas que “gostei” de perceber foi a forma que as ideias surgem e como são construídas ao longo do tempo. Conhecia um pouco sobre o word2vec, mas não conhecia a história até seu surgimento. Não sabia que se falava de utilizar redes neurais para formar um espaço vetorial que captura relações semânticas em 2003, não sabia que em 2011 tentaram gerar uma rede que representa língua sem otimizar para um propósito específico, utilizando aprendizado não supervisionado, não sabia que a ideia de modelar semântica surge por volta de 1950 com Pierce inspirado no trabalho de Shannon. Antes desse estudo, achava que toda ideia que conhecia surgia com o trabalho do word2vec, foi muito interessante ver esse conhecimento sendo construído ao longo do tempo. Percebi isso mais vezes, mas trouxe esse caso como exemplo.

## **ACEITE DA ENTREGA:**

**CEDRIC LUIZ DE CARVALHO:** 

## 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]:

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais** e finalizando na introdução da **arquitetura transformers**.

**Nesta Semana**, o foco foi aprofundar o entendimento dos **primeiros e principais** modelos de linguagem baseados na arquitetura **Transformer**, com **estudo e anotações** sobre cada um deles. Para isso foram lidos os seguintes artigos:

- [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#)  
Anotações BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- [Improving Language Understanding by Generative Pre-Training \(GPT-1\)](#)  
Anotações Improving Language Understanding by Generative Pre-Training (GPT-1)
- [Language Models are Unsupervised Multitask Learners \(GPT-2\)](#)  
Anotações Language Models are Unsupervised Multitask Learners
- [Language Models are Few-Shot Learners \(GPT-3\)](#)  
Anotações Language Models are Few-Shot Learners

Com isso foi possível atualizar o [diagrama sobre a área de interesse](#) que tem sido feito ao longo das semanas

Depois da leitura, fiquei intrigado em saber **como evoluímos desse estágio para os modelos realmente conversacionais**, capazes de manter diálogos naturais e compreender instruções. Descobri então o Artigo do [Instruct GPT](#) com a ideia de **pós-treinamento**.

A partir disso, decidi então **definir como trilha de estudo** o tema de **adaptação e alinhamento de modelos de linguagem baseados em transformers**.

Com isso, **busquei artigos do estilo Survey** que tratem explicitamente sobre o tema que fossem posteriores ao ano de 2024, onde encontrei:

[A SURVEY ON POST-TRAINING OF LARGE LANGUAGE MODELS](#)

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

Ler o survey: [A SURVEY ON POST-TRAINING OF LARGE LANGUAGE MODELS](#) (90 páginas)

Iniciar um glossário que tem como objetivo reunir os principais termos, conceitos e técnicas apresentados no survey acima.

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

**ACEITE DA ENTREGA:**

CEDRIC LUIZ DE CARVALHO: Go! ▾

## Estudo sobre representação semântica da linguagem

Durante a leitura do artigo [A Neural Probabilistic Language Model](#), o qual foi extremamente importante para o início da utilização de redes neurais em modelagem de linguagem, inaugurando o conceito de um espaço vetorial organizado onde poderíamos modelar relações semânticas nesse espaço. Com essa leitura, percebi que esse conceito abre portas para duas utilizações possíveis:

1. Geração de texto, onde levamos em consideração o significado “mais que a estatística”
2. Representação da linguagem

Durante o processo, vim explorando um pouco mais a primeira vertente, contudo, nesse momento, será explorado um pouco mais sobre a representação da linguagem, onde a ideia principal está em responder perguntas como:

- Como representar uma palavra de maneira matemática?
- Como medir o quão “próximo” uma palavra está da outra?

### Busca de referências

Para buscar as referências foi utilizado o site [Research Rabbit](#) onde inserimos o artigo [A Neural Probabilistic Language Model](#) e conseguimos ver os trabalhos similares em um grafo, onde temos os principais trabalhos relacionados exibidos de acordo com número de citações e ano de publicação. Foram então selecionados artigos posteriores ao artigo de busca e que tratassem de alguma forma sobre representação semântica. Com esses critérios, foi possível destacar:

- [Natural Language Processing \(Almost\) from Scratch](#) - 2011- 5540 citações
- [Improving Word Representations via Global Context and Multiple Word Prototypes](#) - 2012 - 972 citações
- [Word representations: A simple and general method for semi-supervised learning](#) - 2010 - 1711 citações

Os 3 artigos acima são os 3 mais citados relacionados ao [A Neural Probabilistic Language Model](#) após sua publicação. Como um deles é extremamente mais citado, será objeto de estudo.

### Natural Language Processing (Almost) from Scratch

Segue um link com minhas revisões sobre o artigo: [Anotações Natural Language Processing \(Almost\) from Scratch](#)

Um resumo da minha visão sobre o artigo seria de que é “revolucionário” ao destrinchar melhor a ideia dos embeddings com um treinamento não supervisionado, observar o espaço vetorial formado e pensar na representação da língua de maneira geral, usando tarefas intermediárias.

Após isso, foi replicado o mesmo processo de busca utilizando agora como base o artigo citado, onde foi possível encontrar outros dois artigos, ambos em 2013 feitos pela Google e com mais de 15 mil citações cada, o que mostra a relevância de ambos na área:

- [Efficient Estimation of Word Representations in Vector Space](#) - 2013 - 17256 citações
- [Distributed Representations of Words and Phrases and their Compositionality](#) - 2013 - 20799 citações

Dado a relevância de ambos artigos serão lidos e geradas revisões sobre os dois.

### **Efficient Estimation of Word Representations in Vector Space**

Segue um link com minhas revisões sobre o artigo: [Anotações Efficient Estimation of Word Representations in Vector Space](#)

### **Distributed Representations of Words and Phrases and their Compositionality**

Segue um link com minhas revisões sobre o artigo: [Anotações Distributed Representations of Words and Phrases and their Compositionality](#)

Os dois artigos são altamente correlacionados e são praticamente “sequências” onde um adiciona algumas melhorias ao outro.

Na minha opinião, esses artigos são revolucionários. Eles mostram um alto grau de sofisticação ao construir um espaço vetorial em que as propriedades das palavras não são definidas manualmente. Partindo de números inicializados aleatoriamente e de dados não rotulados, o modelo aprende, de forma automática, representações que capturam relações complexas entre palavras e que no fim fazem sentido para nós.

Uma das coisas que “gostei” de perceber foi a forma que as ideias surgem e como são construídas ao longo do tempo. Conhecia a ideia que originou o word2vec, mas não conhecia a história até seu surgimento. Não sabia que se falava de utilizar redes neurais para formar um espaço vetorial que captura relações semânticas em 2003, não sabia que em 2011 tentaram gerar uma rede que representa língua sem otimizar para um propósito específico, utilizando aprendizado não supervisionado, não

sabia que a ideia de modelar semântica surge por volta de 1950 com Pierce inspirado no trabalho de Shannon.

Por mais que eu tenha tido algum contato com a área de language modeling anteriormente, ver os trabalhos, motivações e suas ligações tem sido uma experiência muito positiva para mim.

## Estudo sobre atenção pré Transformers

É de conhecimento prévio a importância do artigo “attention is all you need” na modelagem de linguagem. A intenção inicial então é entender um pouco mais sobre como chegamos nessa ideia e como foi desenvolvida até o famoso artigo.

### Busca de referências

Para buscar as referências foi utilizado o site [Research Rabbit](#) onde inserimos o artigo “[attention is all you need](#)” e buscamos os artigos com maior número de citações do ano de 2013 em diante e que tratassem dos mecanismos de atenção de alguma forma. O resultado encontrado foi:

1. NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE - 2014 - 17266 citações
2. Effective Approaches to Attention-based Neural Machine Translation - 2015 - 6864 citações
3. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention - 2015 - 2077 citações.

O primeiro artigo encontrado é também o mais citado, o que nos indica um pioneirismo na área, sendo então nosso primeiro artigo de estudo. Além disso, o segundo artigo se propõe como uma forma de otimização e teste de formas de atenção, onde por mais que agregue não é o foco no momento.

### Anotações Neural machine translation by jointly learning to align and translate

Logo no início do artigo vemos que estamos tratando de tradução automática, tema muito citado no artigo [Sequence to Sequence Learning with Neural Networks](#), lido na Semana anterior.

Uma outra citação interessante é o destaque do vetor que passamos do encoder para o decoder como um “gargalo”.

Eles trazem também que existem estudos comprovando que havia uma grande dificuldade na tradução automática ao lidar com traduções maiores que aquelas vistas durante o treinamento do modelo.

Em seguida o artigo destaca o seguinte:

*“In order to address this issue, we introduce an extension to the encoder–decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated.”*

Sendo muito interessante perceber que eles destacam a ideia inicial como uma busca que é feita nas palavras de entrada.

Uma das seções do artigo que vem a seguir descreve a arquitetura do artigo [Sequence to Sequence Learning with Neural Networks](#), a qual será usada como base para melhorias futuras, indicando a relevância da abordagem seq2seq vista.

Logo em seguida o artigo nos traz a seguinte fórmula:

$$p(y_i | y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

where  $s_i$  is an RNN hidden state for time  $i$ , computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

Por mais “feio” que pareça a principal diferença que temos para a abordagem anterior seria o  $c_i$ , que o artigo chama de “vetor de contexto”

Logo em seguida o artigo nos mostra que o  $c_i$  seria “nada mais” que a soma ponderada dos estados intermediários da RNN no encoder que o artigo chama de  $h_j$  ou anotações.

**The context vector  $c_i$  is, then, computed as a weighted sum of these annotations  $h_i$ :**

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \quad (5)$$

uma rede feed-forward (chamada de alinhador no artigo) então, vai produzir um número a partir do estado atual do decoder e cada estado intermediário do encoder, onde o  $\alpha_{ij}$  é a softmax de todos esses números.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

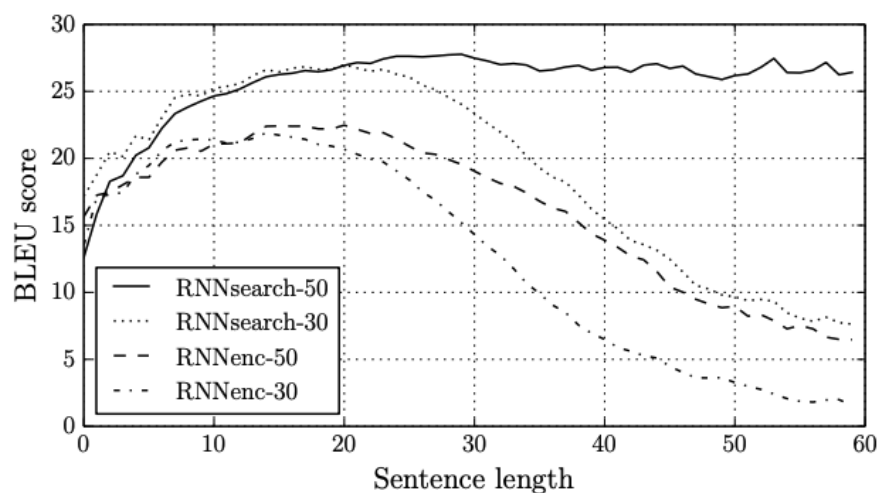
No fim, o fluxo é como se tivéssemos a rede encoder e a rede decoder.

Cada uma das redes vai a cada passo gerar um estado intermediário.

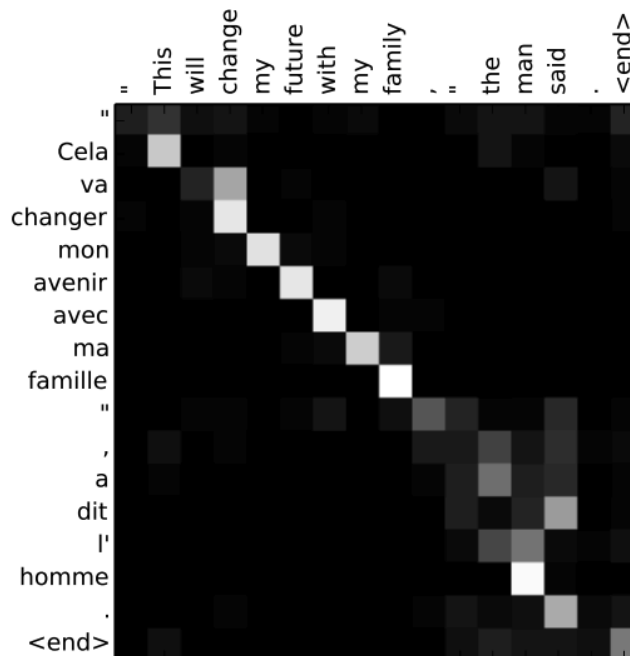
No decoder, a cada novo passo de geração nós vamos pegar o estado intermediário daquele passo pegar cada estado intermediário de cada passo do encoder e passar par a par em uma rede neural. Essa rede neural vai nos dar um número que podemos enxergar como um “score” não normalizado. Esses scores serão então normalizados com uma softmax, gerando “uma distribuição de probabilidade”, onde todos os scores estão entre 0 e 1 e somadas valem 1. Esses scores são então usados para ponderar os estados intermediários do encoder que serão passados em cada etapa do decoder.

Podemos imaginar agora que a cada palavra gerada estamos prestando mais ou menos atenção em partes específicas do texto que está no encoder. Onde o artigo descreve como um mecanismo de atenção onde temos uma recuperação seletiva de informação.

A partir daí, o artigo fala sobre avaliação do sistema usando a tarefa de tradução do inglês para o francês, trazendo então resultados bastante superiores ao uso tradicional das rnns, com enfoque na manutenção desse resultado conforme crescemos o tamanho de sequência, objetivo esse que era de interesse do artigo.



Além do resultado, o artigo destaca também um pouco sobre as famosas matrizes de atenção, onde podemos inclusive ter alguma interpretabilidade sobre o sistema, fato que será um tópico do artigo.



Outro ponto interessante é ver como o artigo também cita outros artigos que nos levaram a diversos outros trabalhos:

*“Since Bengio et al. (2003) introduced a neural probabilistic language model which uses a neural network to model the conditional probability of a word given a fixed number of the preceding words, neural networks have widely been used in machine translation”*

Em minha visão o artigo foi revolucionário do ponto de vista arquitetural por enxergar um problema claro que tínhamos nas redes seq2sq, tanto que um dos métodos que o artigo que inaugura a abordagem seq2seq usa na tradução automática é de inverter a frase no encoder para ter o início dela mais próxima do início da geração. Com isso, nos parece claro a necessidade de “selecionar” melhor o que será passado do encoder para o decoder. Foi exatamente isso que foi feito nesse artigo.

## Estudo sobre a arquitetura Transformers

### Revisão Attention Is All you Need

Logo no início do artigo temos como citação que os modelos dominantes para transdução de sequências até então são baseadas em redes neurais recorrentes com a ideia de encoder e decoder, onde segundo o artigo:

*“The best performing models also connect the encoder and decoder through an attention mechanism”*

Mostrando estar bem conectado com aquilo que foi desenvolvido até então.

A proposta do artigo é uma nova arquitetura “simples” chamada de transformer, baseada unicamente em mecanismos de atenção, eliminando completamente a recorrência.

Uma primeira crítica às redes neurais que justificaria o trabalho é a natureza sequencial das redes recorrentes, ou seja, o estado  $h(1)$  depende do estado  $h(0)$ ... Ou seja, o estado atual depende do estado anterior, que depende do anterior... Com isso, temos uma grande complexidade computacional, pois não é possível paralelizar esse tipo de rede.

Um trecho interessante é citado onde:

*“To the best of our knowledge, however, the Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequencealigned RNNs or convolution”*

Logo após isso o artigo entra na arquitetura do modelo, citando que os modelos competitivos de transdução de sequência utilizam uma arquitetura encoder-decoder, o que foi mantido nessa arquitetura, onde o encoder pega uma série de símbolos e mapeia isso para uma representação contínua, onde o decoder recebe essa sequência contínua e mapeia para outra sequência de símbolos.

Além disso, em cada passo, o modelo é auto-regressivo, consumindo os símbolos gerados anteriormente como entrada adicional ao gerar o próximo.

O artigo nos deixa também a famosa foto da arquitetura transformers:

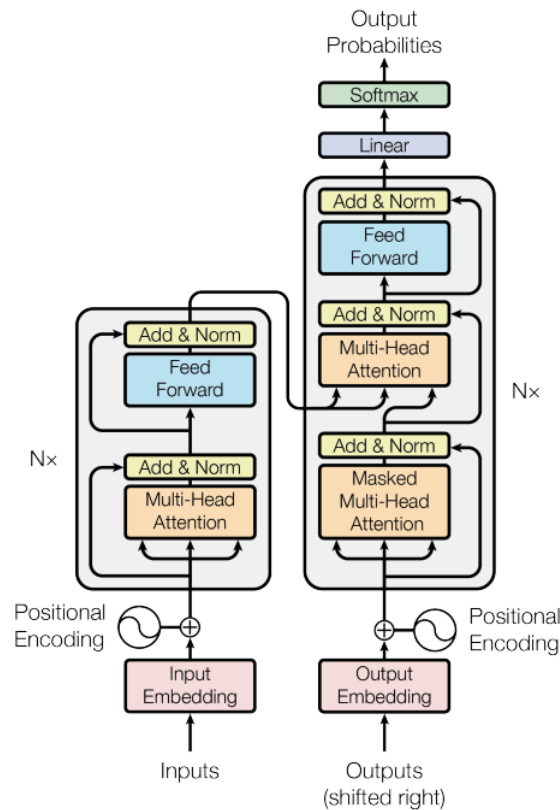
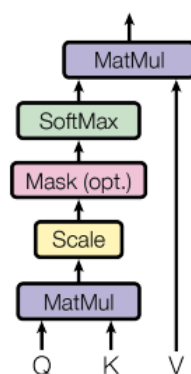


Figure 1: The Transformer - model architecture.

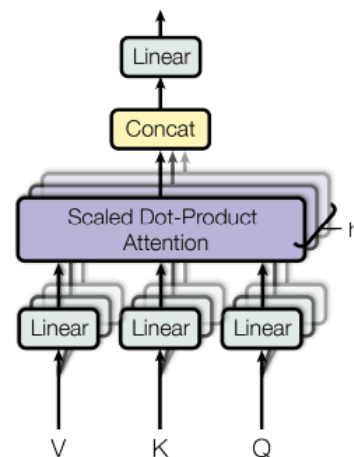
O artigo ainda cita que temos várias camadas de encoder e decoder empilhadas, mas que vou desconsiderar inicialmente para facilitar o entendimento.

Diferente da atenção proposta por Bahdanau, a atenção proposta nos Transformers segue a ideia de utilizar Q,K,V, sendo calculado da seguinte forma:

#### Scaled Dot-Product Attention



#### Multi-Head Attention



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

O artigo ainda cita que usa uma atenção multiplicativa, diferente da que vimos com Bahdanau. No fim, concatenamos o resultado das diversas “cabeças de atenção” e projetamos isso em uma camada linear.

Agora em teoria temos como montar o significado do token de entrada a partir de todos os outros tokens de entrada, usando a atenção, seguindo uma ideia de “compartilhar” significado, parecido com o que vimos em Bahdanau.

O artigo também cita o uso das redes feedforward na arquitetura, assim como a ideia dos “embeddings” aprendidos que surgem em 2003.

Um ponto bem interessante que o artigo traz é que a matriz de pesos que gera os embeddings também é compartilhada com a última matriz antes de passarmos a softmax para obtermos o próximo token, o que é bem interessante.

O artigo também discute algo muito válido, quando tratamos de redes recorrentes, a noção temporal está implícita no funcionamento da rede, visto que os estados possuem dependências. Ou seja, o modelo recorrente é capaz de diferenciar as seguintes frases:

- João emprestou o livro para Maria
- Maria emprestou o livro para João

Como o transformers proposto não é recorrente, essa noção de posição nas frases não é natural. O que o artigo faz quanto a isso é propor adicionar uma representação temporal nos embeddings das palavras, onde essa representação depende de uma série de funções matemáticas trigonométricas que formam noções de temporalidade local e global.

O artigo também justifica o uso da atenção para resolver problemas de dependências de longo prazo, justificando esse funcionamento por um outro viés, o que pode ser verificado no seguinte trecho:

*“One key factor affecting the ability to learn such dependencies is the length of the paths forward and backward signals have to traverse in the network. The shorter these paths between any combination of positions in the input and output sequences, the easier it is to learn long-range dependencies”*

Ou seja, a atenção é vista por um viés de conexão da rede, onde se em uma rede recorrente eu preciso passar por  $n$  steps para encontrar uma palavra, nos transforme isso é constante, visto que cada palavra se liga diretamente com todas as outras.

Além disso, o artigo também traz a questão da interpretação da atenção.

O artigo então detalha alguns passos de treino e parte para os resultados analisando a tarefa de tradução do inglês para o alemão e francês, virando estado da arte nas duas tarefas.

Além disso o artigo nos traz uma tabela interessante mostrando os resultados da variação dos parâmetros do modelo e seu impacto

	$N$	$d_{\text{model}}$	$d_{\text{ff}}$	$h$	$d_k$	$d_v$	$P_{\text{drop}}$	$\epsilon_{\text{ls}}$	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096						4.75	26.2	90	
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	<b>4.33</b>	<b>26.4</b>	213

Além disso, foi questionado e testado se os transformers iriam conseguir generalizar bem no inglês, onde foi feito um teste com menos dados e sem muitas adaptações, em que o resultado foi extremamente satisfatório.

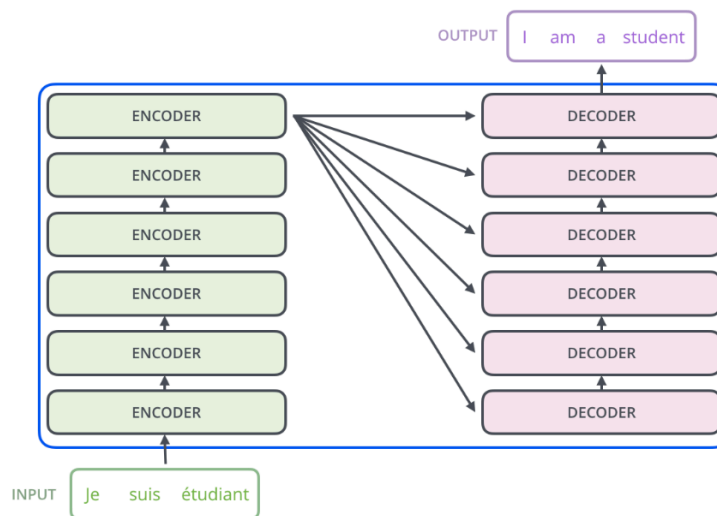
## Revisão The Illustrated Transformer

Após a leitura do artigo do transformers, percebi que “falta” discorrer e entender um pouco melhor o fluxo de informação que é estabelecido dentro da arquitetura, com isso, ao buscar trabalhos que me suportasse nesse sentido, encontrei um famoso blog chamado The Illustrated Transformer produzido pelo autor Jay Alammar.

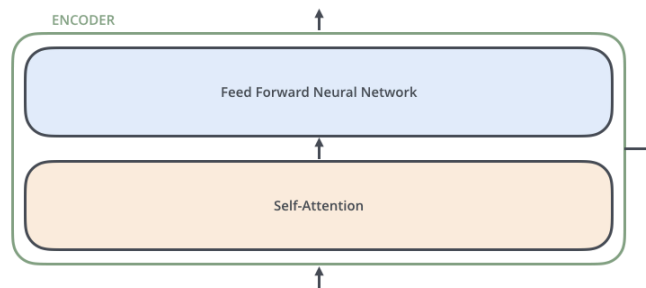
Logo no início do post é comentado sobre um trabalho chamado de The Annotated Transformer, que consiste em um guia comentando o artigo junto com um código de implementação, realizado por pesquisadores de um grupo de NLP de Harvard, o que pode ser um bom objeto de estudo.

Logo de início, ver a maneira como é estruturado os graus de abstração do transformers é bem interessante, onde vemos como uma caixa preta que pega uma sequência e devolve outra e vamos descendo o nível, vendo que dentro da caixa preta por exemplo existe encoder e decoder.

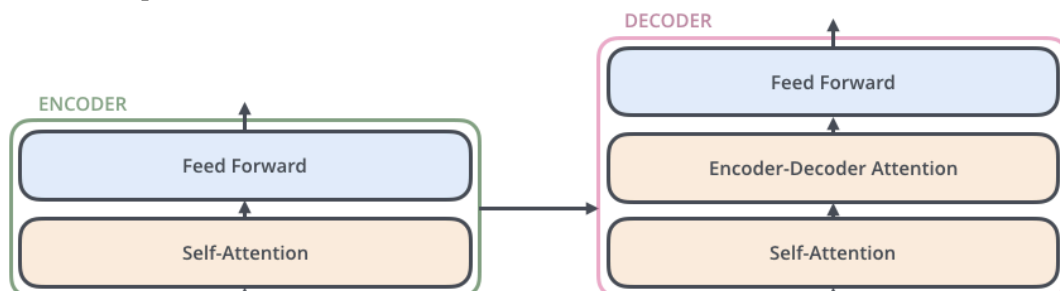
Abrindo um pouco mais vemos a imagem abaixo:



A seguir o post desce um pouco mais ainda e mostra que cada encoder é composto da seguinte forma:

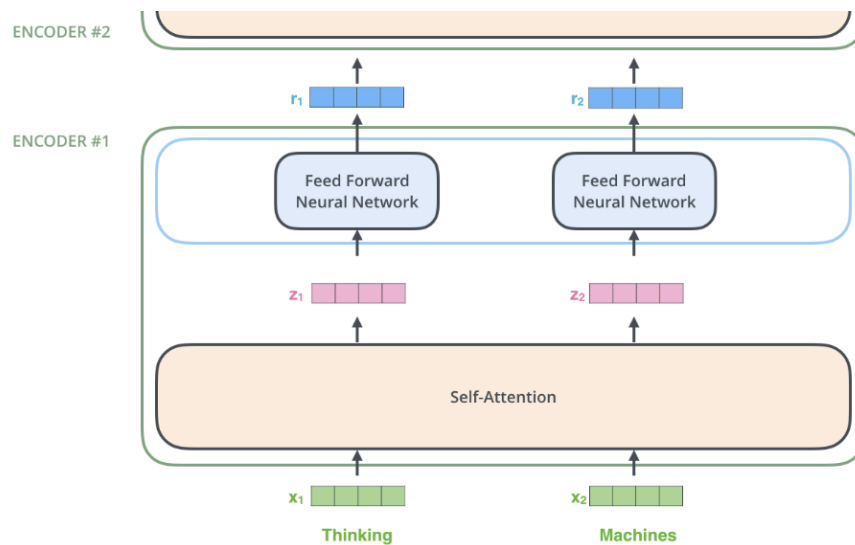


O post também destaca sobre a atenção que temos entre o encoder e o decoder, presente apenas no bloco decoder, visto que o modelo tem atenção entre cada passo do decoder mas também uma atenção completa com os passos do encoder.

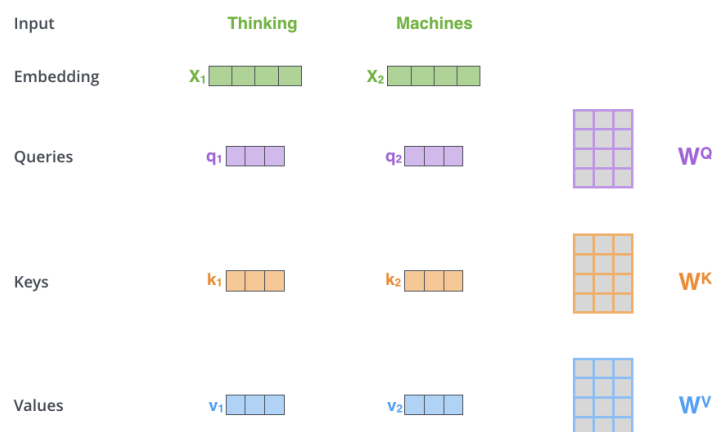


Após isso, começamos a discutir de fato um fluxo de informação dentro da arquitetura transformers, onde temos os tokens de entrada que serão transformados em vetores chamados de embeddings, onde cada token tem seu respectivo embedding.

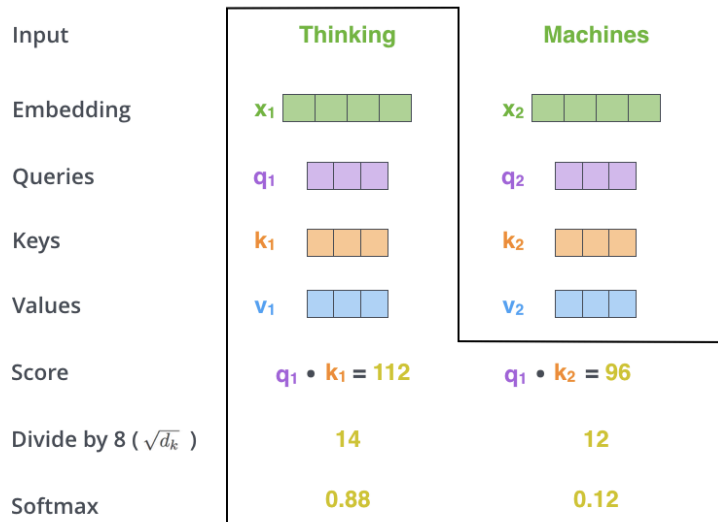
O post também destaca o fato de que podemos “abstrair” que cada camada do encoder irá receber um vetor de tamanho  $d$ , sendo  $d$  o tamanho definido dos embeddings, ou seja, temos uma espécie de “ciclo”.



A imagem abaixo também nos ajuda a entender um pouco mais sobre a atenção dos transformers, onde utilizando os embeddings  $X$  podemos multiplicar eles pelas matrizes  $Q$ ,  $K$ ,  $V$  e obter vetores  $q$ ,  $k$ ,  $v$

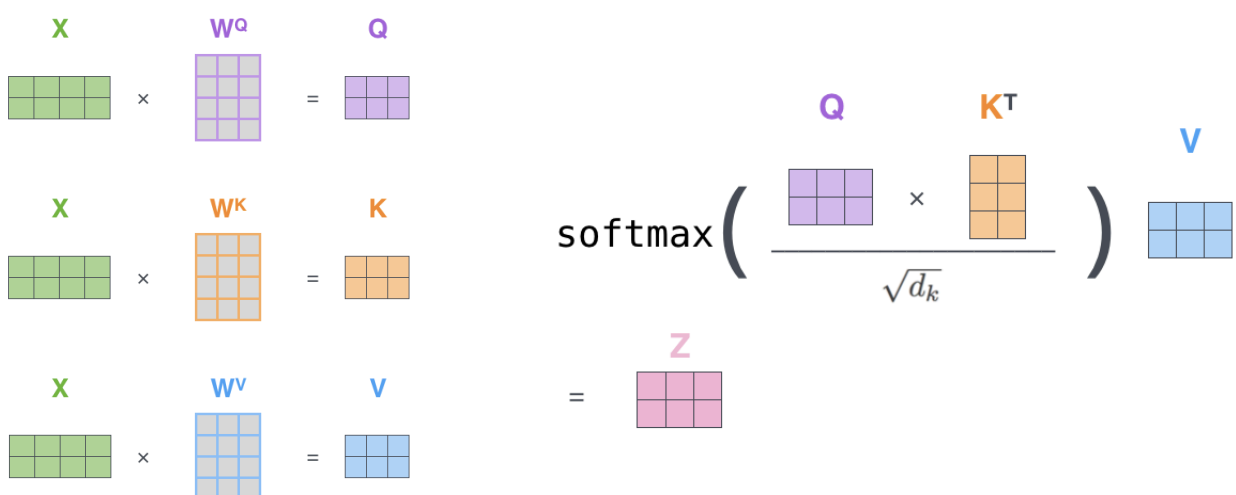


Na imagem a seguir ainda dá para entender como é feito o processo completo. Assim, podemos imaginar que estamos calculando a atenção para thinking, onde nosso objetivo é “tentar atribuir o significado da sentença inteira ponderando cada token usando a atenção”



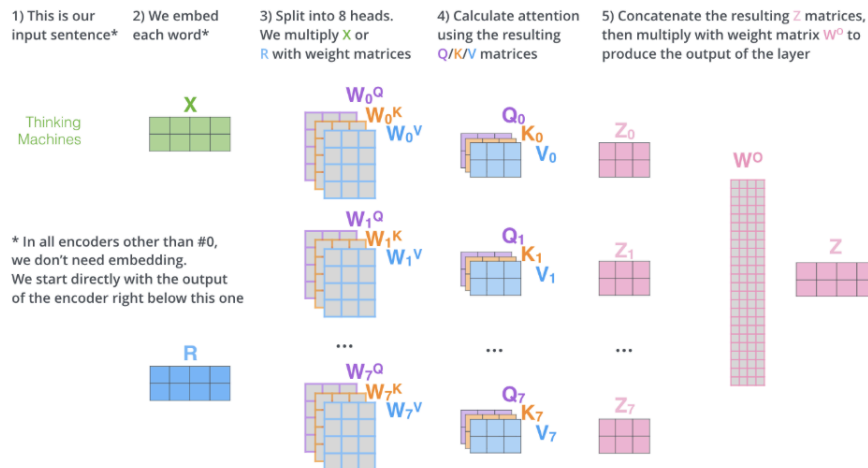
No final desse processo, o vetor final que representa aquela sequência vai ser a soma ponderada pela atenção de cada vetor de value produzido.

O interessante é que o mesmo processo pode ser visto da ótica de multiplicação de matrizes, o que nos ajuda computacionalmente, como mostra a imagem abaixo.



O processo que chamamos de multi-head-attention consiste basicamente em repetir esse processo para cada head e no final concatenar os vetores  $z$  produzidos passando-os para a próxima etapa, a feed-forward

O post traz então uma imagem extremamente interessante que consiste em um grande resumo do processo de atenção + feed-forward dos transformers, podendo ser visto abaixo:



Em seguida o artigo também descreve um pouco melhor a ideia de positional encoding e as granularidades de informação temporal que temos em cada número do vetor onde a ideia de posição é adicionada. No fim das contas a intuição que fica sobre isso é que a ideia de posição destacada no artigo do transformers é muito mais sobre uma posição relativa entre as palavras do que uma noção temporal em si, visto que as funções utilizadas são periódicas, ou seja, se amostrarmos um ponto qualquer sabemos onde estamos em relação aos demais, mas não saberíamos dizer quantas voltas no círculo trigonométrico foram dadas.

O post também fala sobre as ligações residuais, mas isso seria “um acessório das redes neurais” e não algo extremamente relevante sobre a arquitetura transformers em si

Após isso é destacado como se ligam o encoder e decoder, onde no fim das contas, após ter a sequência processada com o encoder, nós utilizamos o resultado da camada de atenção do encoder compartilhando isso com o decoder. isso é possível de ser visto no desenho da arquitetura original. Nós usamos o  $q$  do instante do decoder com o  $k, v$  vindos do encoder.

Ainda assim, o encoder vai gerando token a token de maneira auto-regressiva, até a geração de um token especial, que indica o fim da sequência.

Um outro ponto interessante que o artigo destaca é que no decoder não deveríamos ter acesso a informação futura, ou seja, nós mascaramos essas posições futuras na atenção somente no decoder, pois o mesmo vai estar gerando texto.

O legal é que esse mascaramento é feito utilizando -inf e não 0, o que em um primeiro momento não parece natural, mas se lembrarmos que esse valor passa em uma softmax, veremos que o 0 pode resultar em valores “altos” de atenção.

Logo o artigo trata de uma pergunta interessante, o decoder produz um um vetor de floats. Como transformamos isso em uma palavra?

Ele logo explica que utilizamos uma camada linear final para fazer essa projeção. Ao pensar sobre me deparei com o seguinte:

Se existe uma camada linear final que recebe esse vetor e gera a próxima palavra adequada, significa que a função de tudo que discutimos da arquitetura transformers é gerar uma representação rica o suficiente para que uma camada linear simples consiga classificar.

## Diagrama conceitual Language Modeling

### Visão geral do diagrama



### Partes isoladas do mapa



## Era Neural

Aqui vemos uma clara tentativa em lidar com um problema da abordagem n-gram, onde não conseguimos compartilhar significado entre palavras, visto que eram vistas como símbolos independentes. O artigo inaugura a ideia de espaço vetorial semântico e "abre" margem para duas trilhas, a trilha da geração de texto com redes neurais e da representação semântica

### Primeiro modelo neural

[A Neural Probabilistic Language Model](#)

**■** Geração de texto

**■** Representação semântica da língua

Aqui tratamos o texto como um tipo de dado sequencial, e que não deveria exigir uma janela de contexto de tamanho específico. Com isso, a utilização das redes recorrentes passa a fazer todo sentido e resolver problemas estabelecidos

### Utilização das RNNS

[Recurrent neural network based language model](#)

Destrincha melhor a ideia dos embeddings com um treinamento não supervisionado, como também observa o espaço vetorial formado e pensa na representação da língua de maneira geral, usando tarefas intermediárias para garantir uma boa representação geral, que pode ser usada posteriormente em tarefas específicas.

### Representações Densas e Generalização Multitarefa

[Natural Language Processing \(Almost\) from Scratch](#)

2003

2010

2011

Ainda com as redes recorrentes, passamos a observar um pouco mais sobre o tipo de problema que envolve mapeamento de sequências, como é o caso da tradução, onde apoiado na ideia das redes recorrentes implementa a ideia de um "encoder-decoder"

Guardar toda informação processada em um único vetor era um problema claro na época, como recuperar contextos específicos que foram processados? Com esse questionamento Bahdanau inicia a utilização do primeiro mecanismo de atenção em uma rede seq2seq

### Modelos Seq2seq com RNNS

[Sequence to Sequence Learning with Neural Networks](#)

### Modelos Seq2seq com atenção

[NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE](#)

Famoso artigo que, apoiado nos experimentos anteriores, traz uma arquitetura diferente com maior escalabilidade de dados, gerando resultados impressionantes na época. Explora também diferentes graus de representação da linguagem e operações simples entre vetores que geram resultados extremamente satisfatórios dado a percepção humana

### Escala dos dados (Word2vec)

[Efficient Estimation of Word Representations in Vector Space](#)

Otimiza e expande a arquitetura proposta anteriormente, trazendo conceitos como negative sampling e amostragem de palavras baseados em suas ocorrências no texto

### Treinamento eficiente com negative sampling e amostragem

[Distributed Representations of Words and Phrases and their Compositionality](#)

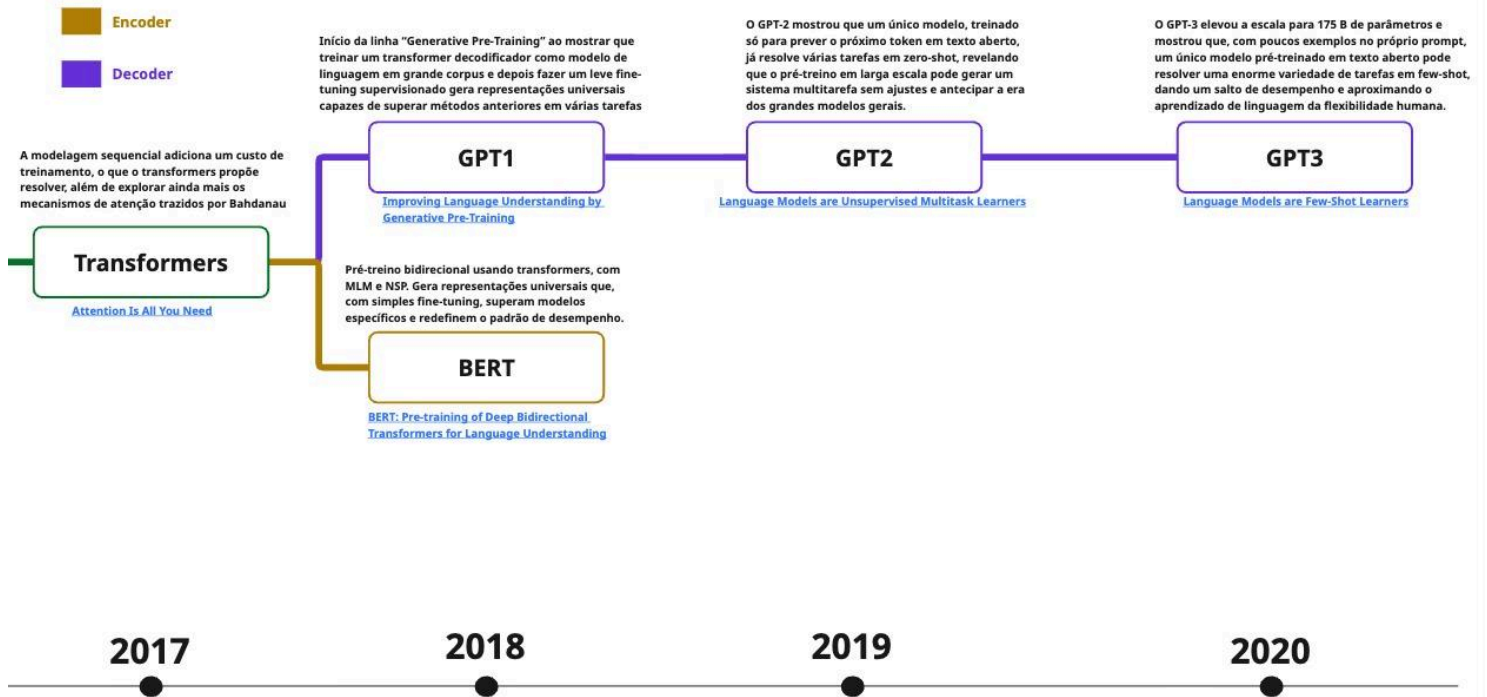
2013

2013

2014

2016

## Era Transformers



## 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:** 2 de out. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais até a arquitetura transformers**. Em seguida, foi feito um aprofundamento na arquitetura Transformers, explorando tanto os modelos baseados em encoder, como o BERT, quanto os modelos baseados em decoder, como o GPT-1, GPT-2 e GPT-3. Com isso, foi definida a trilha de **adaptação e alinhamento de modelos de linguagem baseados em transformers**.

**Nesta Semana**, a partir da pesquisa feita na Semana passada, realizei a leitura do survey [A SURVEY ON POST-TRAINING OF LARGE LANGUAGE MODELS](#) que destaca as principais técnicas de pós-treino de modelos de linguagem baseados em transformers até 2025.

Em conjunto com a leitura, foi realizado anotações sobre o survey:

[Anotações A SURVEY ON POST-TRAINING OF LARGE LANGUAGE MODELS](#)

Além disso, a partir de uma classificação prévia trazida no artigo, gerei um [mapa mental conceitual dos principais métodos e técnicas encontrados](#)

A partir do estudo, mantive um maior interesse pela parte de alinhamento, que em sua maioria envolve técnicas de aprendizado por reforço.

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

**Aprofundar o entendimento sobre PPO e DPO no contexto de alinhamento de modelos de linguagem**

**Mapear os frameworks que oferecem suporte à implementação dessas técnicas.**

---

Mapear datasets de preferência relevantes e aplicáveis às técnicas de alinhamento.

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

Durante esta Semana, estive em viagem para participar do congresso BRACIS

---

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go! ▾](#)

## Estudo sobre pós-treino de LLMs

Logo no começo do artigo é possível ver uma classificação de 5 principais paradigmas do pós treino de LLM's:

**Ajuste fino:** aprimora a precisão específica da tarefa

**Alinhamento:** que garante coerência ética e alinhamento com as preferências humanas;

**Raciocínio:** que avança a inferência em várias etapas, apesar dos desafios no design de recompensas

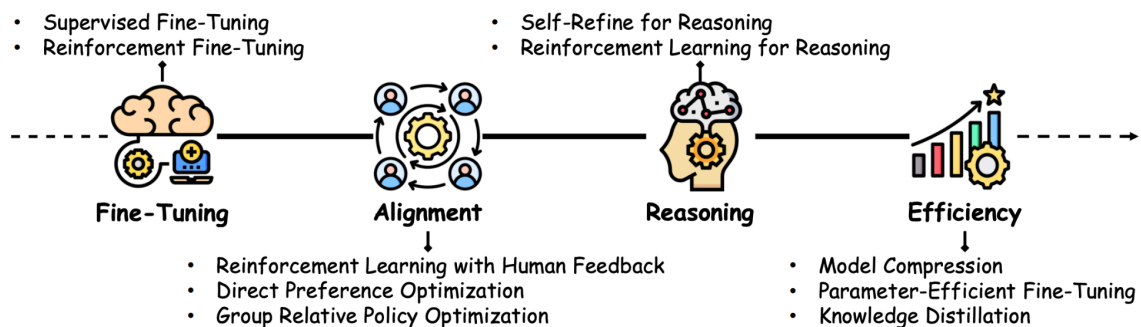
**Eficiência:** que otimiza a utilização de recursos em meio à crescente complexidade

**Integração e Adaptação:** que ampliam capacidades em diversas modalidades

Para o artigo dividimos toda a fase de geração dos grandes modelos de linguagem em pre-training e post-training

Pre-training: Utilização de um vasto conjunto de dados não rotulados para aprender representações boas o suficientes para ajudar a etapa de ajuste fino, nos dados rotulados.

Post-training: Tem por objetivo de refinar e adaptar o modelo para tarefas específicas ou requisitos do usuário



Temos também a imagem acima que explora um pouco as classificações destacadas quanto a técnicas utilizadas.

Ainda assim, o artigo destaca a falta de revisão sistemática em cima da etapa de post-training:

*“In the existing literature, PLMs have been widely discussed and surveyed, while PoLMs are seldom reviewed systematically.”*

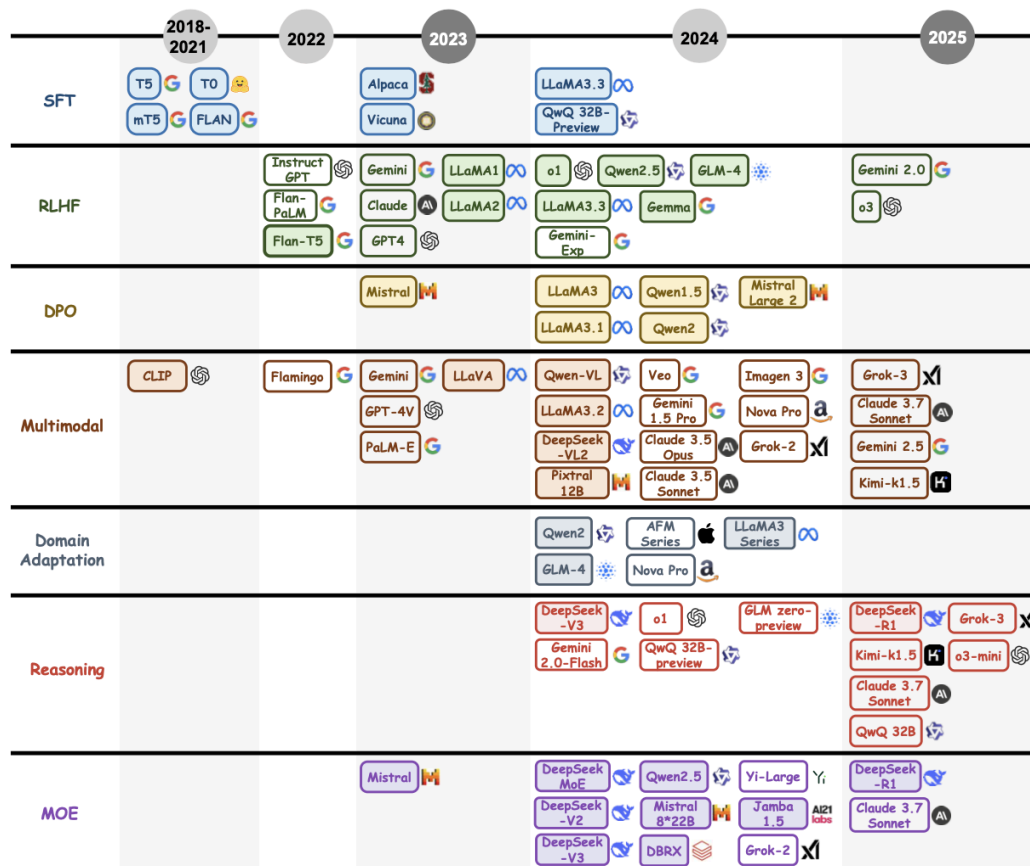
Ainda assim, temos o objetivo do estudo destacado, que particularmente me fez escolher esse trabalho:

*“This paper represents the first comprehensive survey on PoLMs, providing a thorough, structured exploration of the latest advancements in the field. While previous surveys have typically*

*focused on specific aspects of LLM development, such as preference alignment, parameter-efficient fine-tuning, and foundational techniques of LLMs, they have largely concentrated on narrow subtopics. In contrast, this survey takes a holistic approach, providing a complete review of the core techniques commonly employed during post-training and systematically categorizing them. ”*

O artigo então começa a destacar uma abordagem histórica do pós-treino em LLM’s, passando pelo surgimento de modelos como BERT, e definindo o avanço definitivo com técnicas como RLHF empregadas no Instruct-GPT, precursor do Chat GPT.

Com isso, o artigo nos traz então a seguinte figura, com uma linha histórica das técnicas e modelos:



O artigo então entra na seção de fundamentos dos PoLMs, disposta da seguinte forma:

**- Princípio da otimização de políticas:**

O PPO (Proximal Policy Optimization) é uma técnica de reforço, especialmente aplicado quando temos RLHF. Nesse cenário, a manutenção da estabilidade e da eficiência é fundamental. O PPO atinge esses objetivos restringindo o tamanho das atualizações de política, garantindo que as mudanças no comportamento do modelo sejam **graduais e controladas**, evitando assim mudanças catastróficas no desempenho.

O artigo fornece uma explicação mais detalhada do PPO. Pessoalmente, nesse ponto, confesso que tive que parar para buscar fontes externas e gastar um tempo nesse entendimento, visto a dificuldade de compreensão que tive devido ao desconhecimento total sobre o tema.

Como o objetivo inicial é uma “visão geral” sobre a área, escolhi então acompanhar pelo vídeo: [Otimização de Política Proximal \(PPO\) para LLMs explicada intuitivamente](#) que me trouxe alguns insights, dentre eles, a ideia de que as ações dos modelos podem ser vistas como a previsão de cada token, que o estado inicial do modelo é o processamento de sua entrada, onde cada novo token previsto atualiza o estado do modelo. Então, nós temos o fim de um episódio como o modelo parando a geração e recebendo então sua recompensa associada ao episódio.

Uma das ideias que gostei do vídeo é a visão de que a diferença principal entre o uso de rl nesse contexto para o aprendizado supervisionado é o uso do “label”, visto que no reforço não temos essa noção, a ideia é na verdade trabalhar em cima de uma “vantagem”, onde cada ação vai levar a uma vantagem diferente e seremos recompensados e atualizaremos o modelo com base nisso, e é aí que entra o PPO, na definição dessa vantagem.

- **Princípio do RLHF**

Já no RLHF, temos um “paradigma” de treinamento, estando um grau de organização acima do ppo, podendo ter, inclusive, o ppo como ferramenta.

O RLHF segue três etapas principais, primeiro, o pré-treino supervisionado (SFT), em que o modelo aprende com exemplos humanos, depois, o treinamento do Reward Model, a partir de comparações feitas por avaliadores humanos, e, por fim, o ajuste por reforço, onde o modelo é tratado como uma policy e otimizado para maximizar o reward model. Nessa última fase, usa-se geralmente o PPO, que garante atualizações estáveis. Assim, RLHF é a estratégia de alinhamento, enquanto PPO é apenas o algoritmo de otimização usado dentro dela.

- **Princípio do DPO**

Enquanto o RLHF precisa de três blocos: supervisionado, reward model e PPO, o DPO elimina a etapa de reward model explícito e o uso de RL clássico. Ele trabalha diretamente com as preferências humanas, que normalmente são fornecidas em forma de comparações (resposta A é melhor que resposta B). Assim, em vez de treinar um modelo de recompensa separado e depois otimizar via PPO, o DPO já ajusta o modelo de linguagem para aumentar a probabilidade de produzir as respostas preferidas pré definidas.

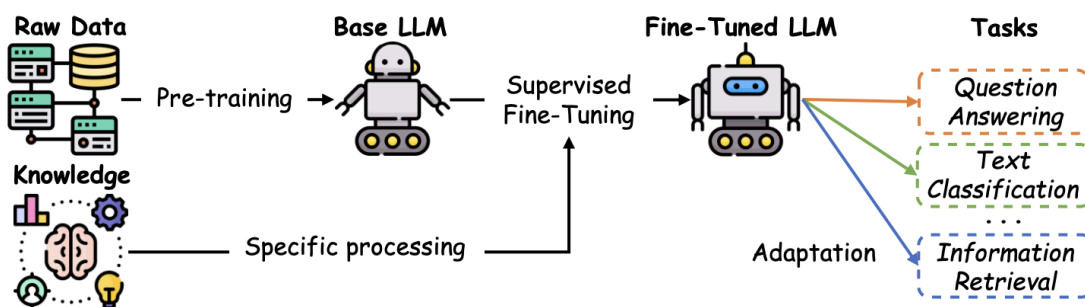
- **Princípio do GRPO**

No GRPO para cada pergunta, o modelo gera várias respostas e compara essas respostas entre si. Em vez de ter uma rede separada para avaliar (como no PPO), ele usa a média das notas

dessas respostas como referência. Cada resposta é ajustada conforme foi melhor ou pior que a média. Depois, o modelo é atualizado de forma controlada, sem mudar demais de uma vez e sem se afastar muito do modelo original.

Já na linha de fine-tuning, onde utilizamos dados rotulados, o survey destaca 3 linhas possíveis:

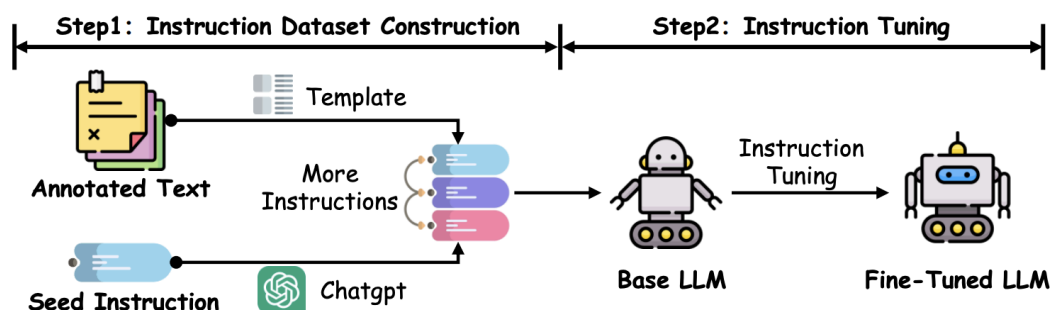
**Ajuste fino supervisionado:** adaptamos os modelos de linguagem pré-treinados a tarefas específicas usando dados rotulados, equilibrando especialização com a capacidade geral adquirida no pré-treinamento.



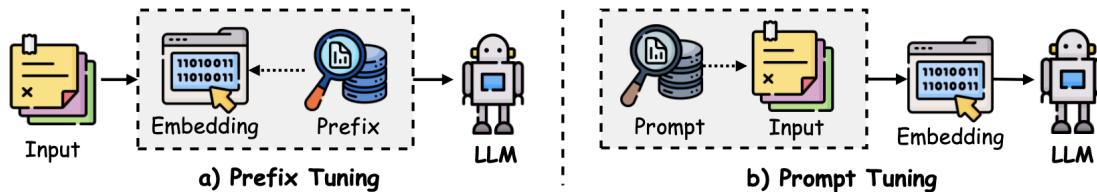
O artigo também define Full-Parameter Fine-Tuning, onde atualizamos todos os pesos do modelo.

Um exemplo disso é o GPT-3 para responder instruções. Onde, o ajuste dos parâmetros completos gera a transição do GPT-3 para o INSTRUCT GPT. Essa abordagem leva ao desempenho ideal, mas é computacionalmente cara, devido à necessidade de atualizar todos os parâmetros.

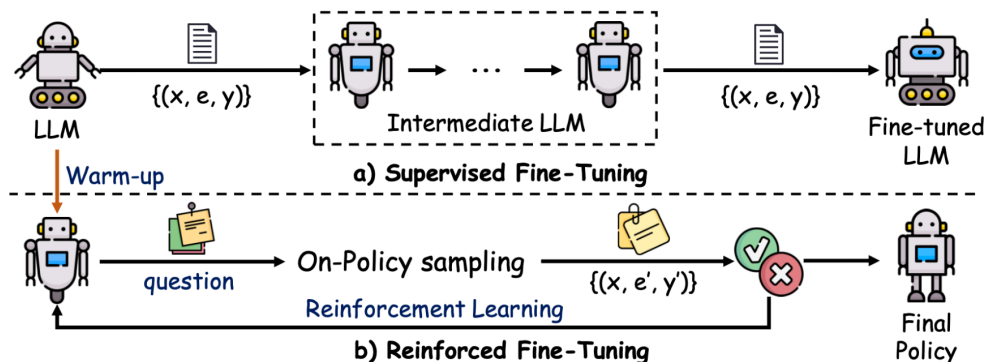
**Adaptive Fine-Tuning:** A ideia aqui é que ao ajustar os modelos pré-treinados, eles se adaptem melhor a diferentes usuários e tarefas. Em vez de simplesmente treinar o modelo com dados rotulados para ajustar seus pesos, ele adiciona **instruções ou prompts** que servem como pistas para direcionar as respostas, gerando técnicas como Instruction tuning e demais.



**Prefix-Tuning:** A ideia aqui é adicionar um vetor a cada camada do transformers e treinar esses vetores de acordo com a tarefa.



**Reinforcement Fine-Tuning:** Aqui misturamos um pouco de RL com SFT, a ideia é fazer primeiro o SFT e posteriormente a etapa com reforço



Muitos estudos surgem nessa ideia e demonstram que esse processo é muito superior ao SFT puro

### Alinhamento:

Pode ser entendida como a técnica que envolve a orientação das saídas do modelo para que se ajustem às expectativas e preferências humanas. No artigo são discutidos 3 métodos baseados em aprendizado por reforço dentro dessa área.

Aprendizado por Reforço com Feedback Humano, que utiliza dados rotulados por humanos como um sinal de recompensa; Aprendizado por Reforço com Feedback de IA, que utiliza o feedback gerado por IA para abordar problemas de escalabilidade; e Otimização de Preferência Direta, que aprende diretamente a partir de dados de preferência humana pareados, sem exigir um modelo de recompensa explícito.

**RLHF:** O artigo cita a “incapacidade” de um fine tuning supervisionado de capturar preferências humanas mais diferenciadas ou adaptativas, e cita esse método como criado para lidar com isso. Além disso, cita o vasto uso desse tipo de técnica nos modelos conversacionais que vieram depois do InstructGPT.

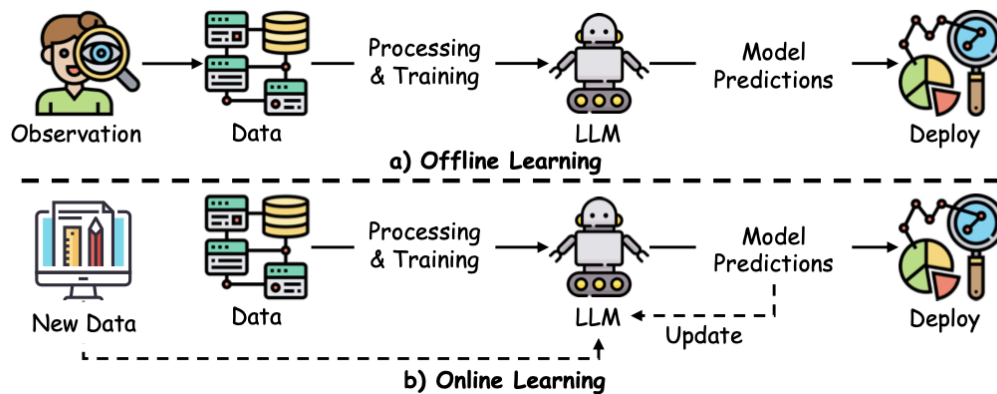
Nesse método nós frequentemente trabalhamos em cima das recompensas, e um dos métodos bastante utilizados é treinar um modelo de recompensa a partir de anotações humanas.

Além disso, na hora de aprender a política, podemos tratar isso da seguinte forma:

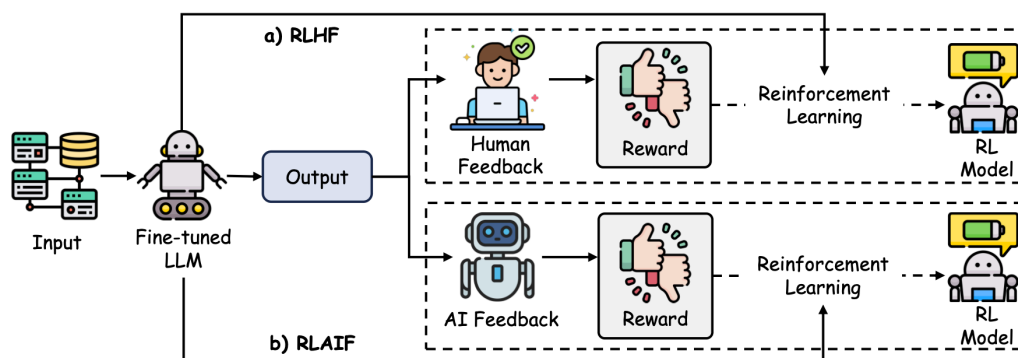
**Online:** o modelo coleta feedback humano em tempo real, conforme gera novas respostas. Isso permite ajustar a política continuamente, mas exige mais interação humana.

**Offline:** usa apenas dados já coletados com preferências humanas para treinar a política, sem precisar de feedback ao vivo. É mais barato em termos de anotação, mas pode sofrer com limitações do conjunto de dados.

**Híbrido:** combina os dois, aproveita dados já existentes (offline) e depois refina com feedback em tempo real (online), equilibrando custo, eficiência e qualidade do aprendizado.



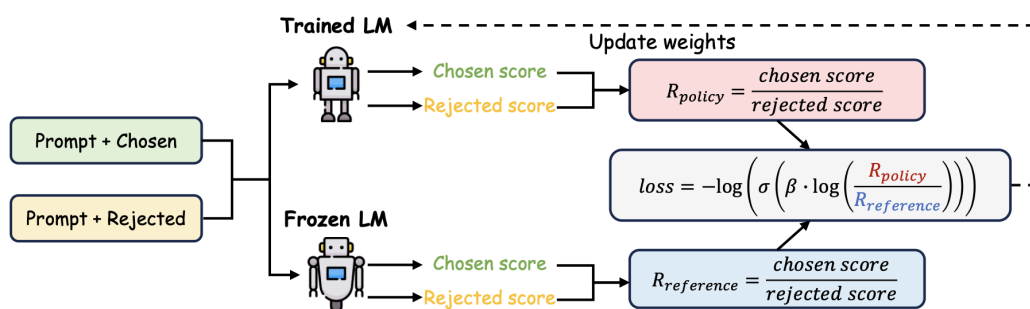
**Reinforcement Learning with AI Feedback:** Nessa ideia, nós usamos os LLMs para gerar sinais de feedback. Essa abordagem pode complementar ou substituir o feedback humano, fornecendo dados de preferência mais escaláveis e de menor custo em tarefas onde as anotações humanas são escassas, custosas ou inconsistentes.



**Minha visão:** Uma das coisas que ainda me “incomoda” é o fato de existir o treinamento de um modelo para ditar o “label” de outro modelo, como é feito com os modelos de recompensa que comumente vemos.

O tópico acima, inclusive é retratado no artigo para falar sobre os modelos de recompensa direta (DPO)

**DPO:** Surge justamente nesse contexto de treinar utilizando a recompensa de maneira direta, para minimizar a dificuldade de se atingir um bom modelo de recompensa.



**Reasoning:** O artigo separa dois métodos de raciocínio:

**Self-Refine for Reasoning:** Muito do que é tratado é sobre a capacidade do modelo de revisar e ajustar suas próprias respostas, sozinho ou com apoio externo, para corrigir erros e tornar o raciocínio mais consistente e confiável.

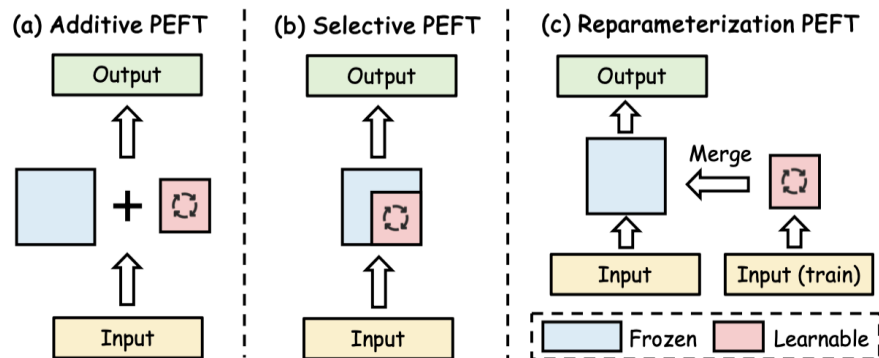
**Reforço para Reasoning:** A ideia principal é que o raciocínio não é uma “resposta única”, mas sim uma sequência de passos, onde podemos modelar isso usando reforço, se parecendo com a ideia que vimos de que cada token é uma ação. O objetivo final então é maximizar a probabilidade de se atingir uma resposta correta. Uma das ideias também é utilizar as cadeias de Markov, onde enxergamos o processo de raciocínio como uma sequência de decisões encadeadas.

Uma das formas de lidar com a recompensa seria por exemplo uma recompensa esparsa caso o modelo acerte a resposta no fim do processo.

Outro ponto bastante citado é no artigo DeepSeek, que conseguiu treinar um modelo sem a parte de SFT, apenas por “reforço em larga escala”.

**Eficiência:** Outra linha possível é a eficiência dos modelos, onde podemos comprimir os modelos, onde entram processos como a quantização, a poda de parâmetros, onde buscamos excluir alguns parâmetros do modelo modificando o mínimo possível do resultado.

Além disso, o ajuste fino com eficiência de parâmetros, onde usamos o PEFT, onde congelamos todo o modelo e otimizamos matrizes decompostas, o que resulta em uma otimização mais leve.



outro tópico trazido na parte de eficiência é a destilação, onde a ideia é “mimetizar” o comportamento de um modelo professor em um modelo aluno, onde aproximamos o máximo possível o aluno desse professor. O modelo professor, nesse caso, via de regra é um modelo maior que o aluno, fazendo com que ambos tenham performances “próximas” mas o aluno tendo bem menos parâmetros.

O artigo cita também o uso de destilação no DeepSeek, principalmente para forçar padrões de reasoning aprendidos pelo modelo professor.

**Integração e adaptação:** Na parte de integração o artigo discute os multimodais, que não é foco da pesquisa nesse momento, já na adaptação é falado sobre domínios específicos e sobre por exemplo edição de conhecimento do modelo já instaurado sem um retreinamento ostensivo.

Outro tema de destaque são os merges de modelos, em que mesclamos algumas camadas e etc de diversos modelos para se obter uma maior “robustez”. Podemos fazer isso a nível de camadas, um dos exemplos é utilizar um merge de modelos para “trazer” o modelo para domínios distintos.

## 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:** 8 de out. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais até a arquitetura Transformers**, incluindo suas variantes, encoder e decoder. Com isso, foi definida a trilha de **adaptação e alinhamento de modelos de linguagem baseados em transformers**, e, a partir da **leitura de um survey** da área, foram mapeadas as **principais técnicas que orientam essa trilha**.

Durante esta Semana, após identificar que a maioria dos métodos modernos de adaptação e alinhamento de modelos de linguagem envolve o **RLHF** decidi ler o artigo [Training Language Models to Follow Instructions with Human Feedback](#) que populariza a técnica e marcou uma mudança de paradigma rumo a **modelos realmente conversacionais e capazes de seguir instruções**.

📄 Anotações Training language models to follow instructions with human feedback

Com isso, surgiu o interesse em **replicar o experimento descrito no artigo**, trazendo a um modelo de linguagem a **capacidade de seguir instruções através do processo de RLHF**.

Para isso, foi escolhido inicialmente o modelo **Gemma 3 1b base**, por ter um **tamanho similar ao Instruct GPT** descrito no artigo. Em seguida, realizei alguns testes com perguntas simples e outras extraídas do próprio estudo, que demonstram que o modelo, em sua forma base, **não é capaz de seguir instruções adequadamente**. 📄 [base\\_model\\_test.ipynb](#)

Dessa forma, decidi **aprofundar meus estudos sobre RLHF**, técnica trazida no artigo e que servirá como **base conceitual e prática para guiar o processo**. 📄 Estudo sobre RLHF

Após esse mapeamento inicial, busquei compreender os principais **métodos de aprendizado por reforço utilizados**, encontrando o artigo [DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback](#) que versa sobre dois métodos que suportam o processo de RLHF.

☰ Anotações Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preferen...

Com um objetivo definido e um entendimento mais claro sobre o processo e os métodos que o sustentam, busquei identificar **frameworks capazes de apoiar a implementação do RLHF**, realizando um **levantamento das principais opções disponíveis**.

☰ Frameworks para RLHF

Após isso, optei por trabalhar com o framework **TRL da Hugging Face** devido ao seu **ecossistema, comunidade ativa e por existirem cursos disponíveis**, o que pode auxiliar bastante em um contato inicial, contudo, os frameworks OpenRLHF e NeMo-RL ainda são boas opções e podem ser explorados posteriormente.

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

Explorar os tutoriais fornecidos pelo Hugging Face da biblioteca TRL

Buscar e selecionar datasets adequados para SFT e preferências (RLHF)

Implementar o código da Fase 1: SFT (Supervised Fine-Tuning)

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

---

**ACEITE DA ENTREGA:**

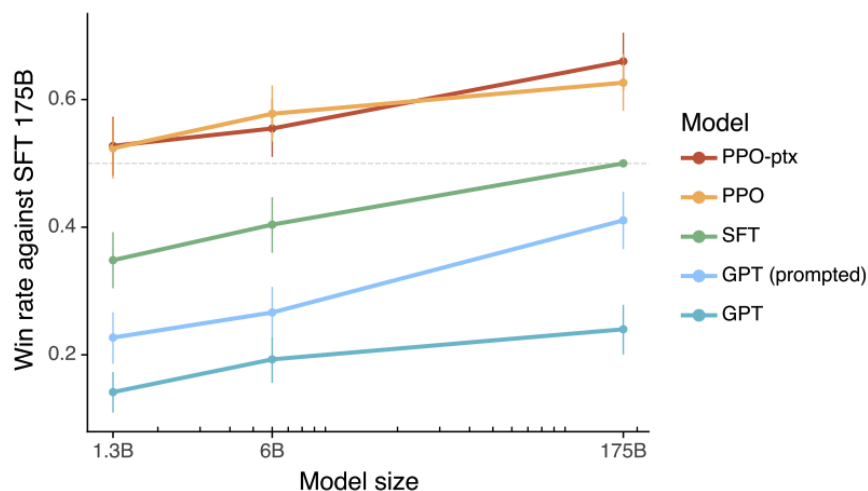
CEDRIC LUIZ DE CARVALHO: Go! ▾

## Escolha da trilha: Adaptação de modelos de linguagem com aprendizado por reforço e feedbacks humanos

### Escolha da aplicação: replicando o artigo “Training language models to follow instructions with human feedback”

De início podemos perceber que o grande objetivo aqui é fazer com que os modelos de linguagem sigam as instruções do usuário. E trazem ainda que o modelo InstructGPT de 1.3 bilhões de parâmetros supera o modelo gpt3 de 175 bilhões, mostrando o poder do alinhamento.

Eles mostram ainda o ganho que se tem com o alinhamento em comparação com demais métodos.



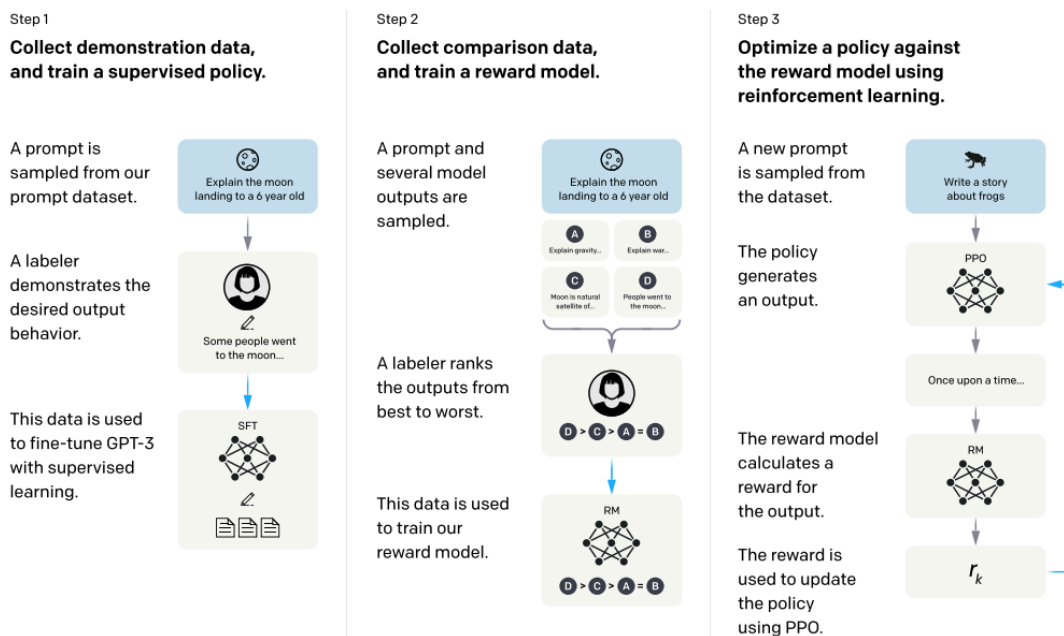
Outro trecho que destaca a motivação do artigo:

*“usado para muitos LMs grandes recentes — prever o próximo token em uma página da internet — é diferente do objetivo “seguir as instruções do usuário de forma útil e segura”*

Um dos motivos do artigo ser um “marco” é a mudança de paradigma para ter como enfoque modelos que são realmente conversacionais.

*“we want language models to be helpful (they should help the user solve their task), honest (they shouldn’t fabricate information or mislead the user), and harmless (they should not cause physical, psychological, or social harm to people or the environment)”*

O artigo deixa então uma imagem com 3 passos explicando a ideia do RLHF.



No passo 1 a ideia é que, dado um prompt e um label realizar um sft no modelo, pois o mesmo será utilizado como política inicial para os próximos passos.

No passo 2 coletamos diversas respostas do modelo para um único prompt e rankeamos essas respostas. Com isso, treinamos o modelo de recompensa, que será responsável por estipular a recompensa de qualquer resposta do modelo, dado qualquer prompt.

No passo 3 nós de fato fazemos o treinamento da política (modelo de linguagem com sft) utilizando o modelo de recompensa treinado anteriormente.

No artigo, inclusive, o treinamento é feito usando a arquitetura GPT3, o que permite uma comparação justa.

A openAI então contratou alguns anotadores para gerar alguns prompts seguindo o seguinte:

**Simples:** os rotuladores que criavam uma tarefa arbitrária, garantindo que as tarefas tivessem diversidade suficiente.

**Few-shot:** os rotuladores criavam uma instrução e vários pares de consulta/resposta para essa instrução.

**Baseado no usuário:** dado os vários casos de uso declarados na API OpenAI, os rotuladores criaram prompts correspondentes a esses casos de uso.

O artigo então destaca que obtiveram 3 conjuntos de dados distintos a partir disso, sendo o conjunto para o SFT (13 mil prompts), para treinar o modelo de recompensa (33 mil prompts) e o conjunto para o PPO com 31 mil prompts.

Um fato que me chamou atenção é que o fine tuning supervisionado é feito por 16 épocas.

Os modelos de recompensa treinados também são “pequenos” tendo cerca de 6B parâmetros.

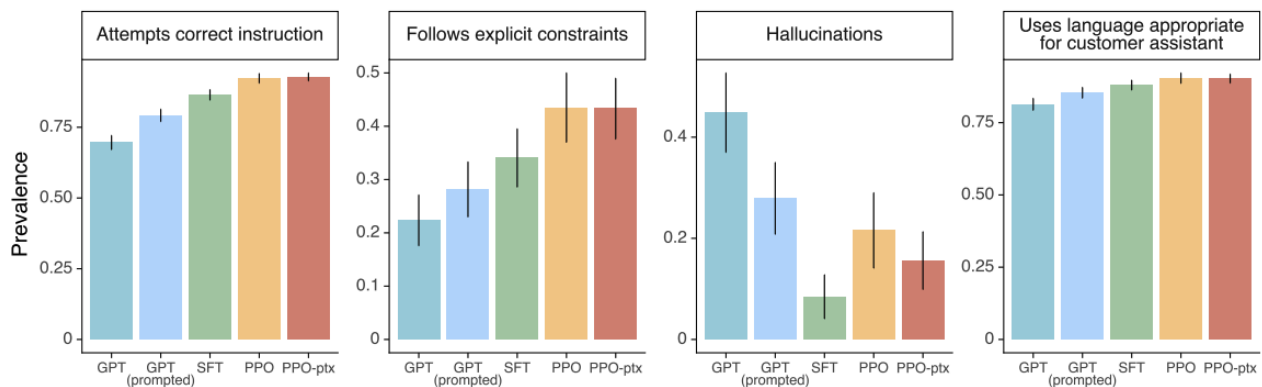
Foi então usado o PPO para o treino, onde dado um prompt, o modelo responde, recebe 1 recompensa do Reward Model e o episódio acaba.

Um fato interessante é o uso de:

*“KL penalty from the SFT model at each token to mitigate overoptimization of the reward model.”*

Aqui entra também um ponto interessante do artigo ao definir o significado de alinhamento. onde define-se que o modelo deve seguir instruções, mas também inferir a intenção a partir de um prompt de poucos exemplos ou outro padrão interpretável

Já na aba de resultados, temos lacuna bastante interessantes



Eles testaram também com rotuladores que não produziram nenhum dado, mostrando que não é overfitting

Além disso, o resultado abaixo é extremamente interessante e marca essa capacidade de seguir instruções

---

**Prompt:**

Escreva uma pequena história sobre uma rã que viaja no tempo até a Grécia antiga em francês.

**Conclusão do GPT-3 175B:**

Escreva uma história sobre uma criança que quer saber tudo sobre os jogos dos deuses e que se vê envolvida numa das suas histórias.

Escreva uma história sobre um jovem que tem uma aventura em uma época distante com uma garota daquela época.

Escreva uma história sobre uma criança que é enviada para a escola e tem um amigo imaginário que lhe permite viajar no tempo.

**Conclusão do InstructGPT 175B:**

A rã estava perdida e exausta. Ela tinha apenas um objetivo em mente: viajar no tempo e chegar à Grécia antiga.

Primeiro problema: encontrar um meio de transporte. Ela tinha ouvido dizer que uma das melhores maneiras era entrar em uma máquina do tempo. Ela se lançou e, de repente, se viu no passado, em uma noite estrelada com um aroma doce e envolvente.

A rã avança então com passos rápidos e aventureiros pelas ruas inclinadas e pequenas pousadas da cidade antiga. [...]

---

**Sugestão:**

Qual é o objetivo da lista C no código abaixo?

```
def binomial_coefficient(n, r):
    C = [0 para i no intervalo (r + 1)]; C[0]
    = 1;
    para i no intervalo (1, n + 1):
        j = min(i, r); enquanto j
            > 0:
                C[j] += C[j - 1]; j -=
                1;
    retornar C[r]
```

---

**Conclusão do GPT-3 175B:**

- A. para armazenar o valor de C[0]
- B. para armazenar o valor de C[1]
- C. para armazenar o valor de C[i]
- D. para armazenar o valor de C[i - 1]

**Conclusão do InstructGPT 175B:**

A lista C neste código é usada para armazenar os valores do coeficiente binomial à medida que a função itera pelos valores de n e r. Ela é usada para calcular o valor do coeficiente binomial para um determinado valor de n e r e armazena o resultado no valor de retorno final da função.

## Exemplo de inferência em um modelo base



```
1 from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig, Gemma3ForCausalLM
2 import torch
3
4 tokenizer_base = AutoTokenizer.from_pretrained(
5     "google/gemma-3-1b-pt",
6 )
7
8 model_base = AutoModelForCausalLM.from_pretrained(
9     "google/gemma-3-1b-pt",
10    dtype=torch.bfloat16,
11    device_map="auto",
12    attn_implementation="sdpa"
13 )
14
15
16 prompt = "hi, how are you?"
17
18
19 input_ids = tokenizer_base(prompt, return_tensors="pt").to(model_base.device)
20 output = model_base.generate(**input_ids, max_new_tokens=256)
21 print(tokenizer_base.decode(output[0], skip_special_tokens=True))
```

Resposta gerada a partir do teste acima:

```
1 hi, how are you? I would be happy if you could help me with a small question.  
2  
3 I have the following environment  
4  
5 OS: Debian 9  
6  
7 VFS: vfs tools 2.40.0  
8  
9 kernel: linux-6.3.0-23-amd64  
10  
11 python: 3.7.0  
12  
13 Here's the result of my ldd :
```

Com isso, fica claro a falta de capacidade de modelos “base” (apenas pré treinados) em responder instruções de maneira adequada.

## Estudo sobre Reinforcement Learning with Human Feedback (RLHF)

Motivado pelo artigo [Training language models to follow instructions with human feedback](#), buscando entender mais sobre RLHF, comecei pelo artigo do Hugging Face:

### [Illustrating Reinforcement Learning from Human Feedback \(RLHF\)](#)

Um dos pontos interessantes deste artigo é trazer em uma única frase, a visão geral sobre o processo de RLHF:

*“Vamos primeiro formular esta tarefa de ajuste fino como um problema de RL. Primeiramente, a política é um modelo de linguagem que recebe um prompt e retorna uma sequência de texto (ou apenas distribuições de probabilidade sobre o texto). O espaço de ação desta política são todos os tokens correspondentes ao vocabulário do modelo de linguagem (frequentemente na ordem de 50 mil tokens) e o espaço de observação é a distribuição de possíveis sequências de tokens de entrada, que também é bastante grande, considerando os usos anteriores de RL (a dimensão é aproximadamente o*

*tamanho do vocabulário → comprimento da sequência de tokens de entrada). A função de recompensa é uma combinação do modelo de preferência e uma restrição à mudança de política.”*

De novo temos também uma noção sobre a divergência de KL, que visa fazer com que o modelo não se afaste muito do pré treinado inicial, o que pode ser útil para garantir que o modelo produza trechos de texto razoavelmente coerentes. Sem essa penalidade, a otimização pode começar a gerar texto sem sentido, mas que engana o modelo de recompensa

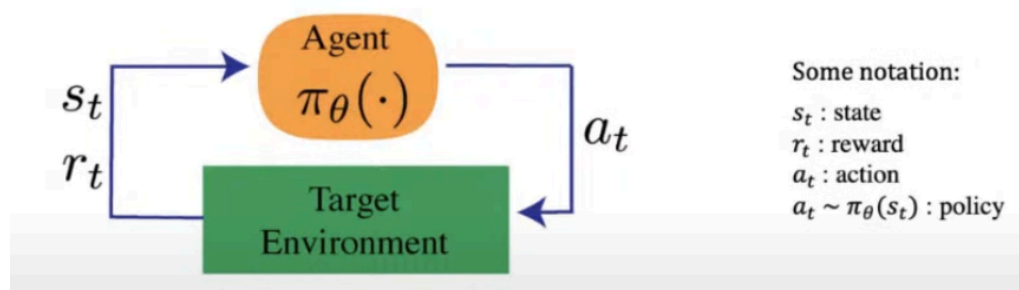
O artigo também traz frameworks que suportam o RLHF.

Ainda explorando os materiais do Hugging Face, encontramos o vídeo

### [Reinforcement Learning from Human Feedback: From Zero to chatGPT](#)

O vídeo traz e começa com uma visão histórica, e então, começa a discorrer sobre a sequência de passos já descrita no artigo que origina o Instruct GPT

Um ponto interessante do vídeo é tentar aplicar o artigo dentro desse esquema de reforço, o que ajuda a compreender o tema.



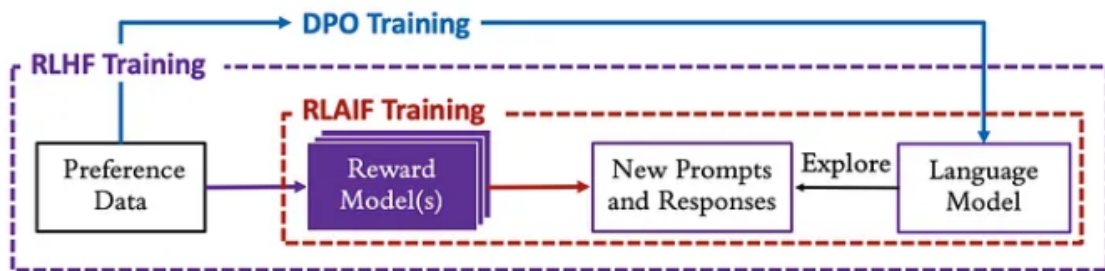
Ainda assim, seguimos o estudo a partir do blog medium de um mestre em computação pela Northeastern University com o título de

### [Fine-Tuning LLMs with Human Feedback \(RLHF\): Latest Techniques and Best Practices](#)

Uma boa prática é iniciar o RLHF de um modelo com SFT ajustado, ao invés de um pré treinado bruto.

O artigo então, traz nesse processo a técnica do DPO, que ao invés de usar um modelo de recompensa, utiliza os próprios dados anotados como recompensa direta. Essa técnica pode ter resultados semelhantes ao PPO, mas sem precisar treinar o modelo de recompensa.

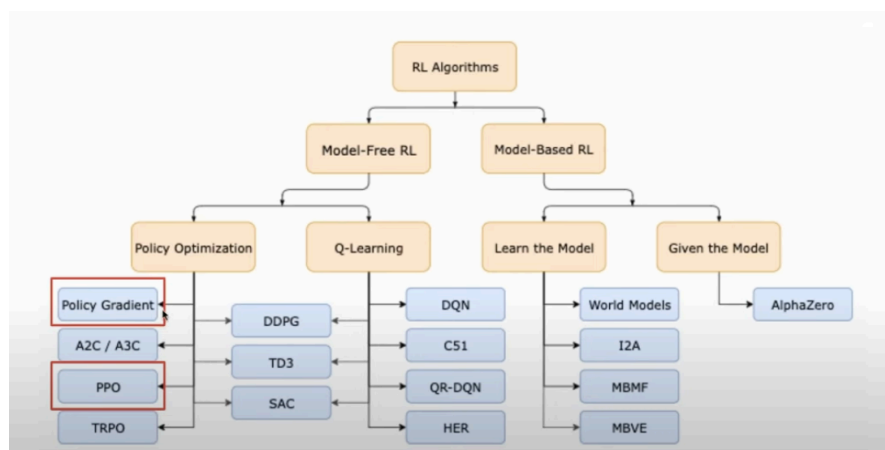
O artigo também traz a ideia de RLAIIF, onde usamos LLM's para gerar rótulos de feedback em vez de humanos



A partir disso, busquei entender um pouco mais sobre PPO e DPO (técnicas que vi que suportam o processo de rlhf), antes de partir para busca de frameworks que suportem esses métodos.

Iniciei esse estudo vendo o vídeo [LLM Training & Reinforcement Learning from Google Engineer | SFT + RLHF | PPO vs GRPO vs DPO](#) produzido por um engenheiro de software da Google.

Um dos grandes ganhos que o vídeo traz é a classificação da imagem abaixo, que nos mostra, dentro do aprendizado por reforço onde estamos, que são os algoritmos de Policy Optimization



Um dos pontos que pensei em meus estudos e que foram confirmados no vídeo é que as técnicas utilizadas depois como DPO e GRPO, vem para tentar tirar um pouco da complexidade do PPO em que temos que treinar um modelo de recompensas.

Entendi um pouco também que no DPO, com um dado de maior preferência sobre o outro (preferido e rejeitado) nós otimizamos o modelo para que os logits da resposta preferida sejam maiores a medida

que diminuimos esses logits para a resposta rejeitada, contando também que não vamos nos afastar muito do modelo de referência (SFT pré alinhamento). Com isso, não precisamos treinar o modelo de recompensa.

## Leitura Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback

O artigo começa se propondo a avaliar o efeito de cada etapa do processo de RLHF, se mostrando valioso para nosso contexto, especialmente ao comparar dados de preferência, algoritmo de aprendizado, modelo de recompensa e prompts de treinamento de política.

O artigo já começa citando que qualidade das preferências (escolha dos pares escolhidos/rejeitados) importa mais do que a qualidade das gerações reais consideradas. Como também, cita que o PPO se sai melhor que o DPO na maioria dos testes.

O artigo busca então explicar o PPO e DPO, onde no PPO começa falando sobre o modelo de recompensa, que deve produzir valores escalares.

O mais interessante é que o artigo cita também o funcionamento a mais baixo nível do modelo de recompensa, onde podemos usar a arquitetura transformers, mas trocando a camada final de classificação por uma camada de regressão.

O objetivo do treinamento então é maximizar o ganho de recompensas à medida que minimiza a entropia de KL. Com isso podemos ter uma espécie de “jogo”, onde buscamos as maiores recompensas próximas ao espaço que está o modelo que já possuímos (SFT).

Já no DPO, nosso objetivo é maximizar as respostas aceitas e minimizar as rejeitadas.

Tanto PPO quanto DPO possuem uma estrutura dos dados idêntica.

Assim, o DPO é mais eficiente em termos de computação e velocidade. Contudo, uma das grandes vantagens do PPO é que temos um treino online, ou seja, treinamos baseado na política atual. Se temos um modelo de recompensas, podemos utilizar a política atual para realizar o treinamento, enquanto no DPO, nós temos fixadas as etapas, visto que a recompensa depende inteiramente dos dados, tendo um treinamento offline.

De maneira muito interessante, também trazem no artigo:

*“Nossos melhores modelos superam o modelo SFT em mais de 8 pontos nessas categorias. Em contraste, a aprendizagem baseada em preferências não contribui para a factualidade, com todos os modelos permanecendo com 1 ponto de diferença entre si. Isso sugere que a aprendizagem baseada em preferências é mais útil para aprimorar habilidades relacionadas ao chat (seguimento de instruções, veracidade) e aprender características estilísticas, mas menos eficaz para ensinar novos fatos a um modelo.”*

Além disso, em todos os conjuntos de dados, os modelos treinados com PPO superam os modelos treinados com DPO.

No restante do artigo são explorados outros tópicos como dados sintéticos e modelos de recompensa.

## Levantamento de Frameworks para RLHF

### Frameworks levantados:

- [Hugging Face TRL](#)
- [TRLX](#)
- [RL4LMS](#)
- [OpenRLHF](#)
- [NeMo-RL \(NVIDIA\)](#)

### Hugging Face TRL:

Dentro da descrição e overview da biblioteca vemos que ela fornece suporte ao pós treino de modelos de linguagem incluindo técnicas como: Supervised Fine-Tuning (SFT), Proximal Policy Optimization (PPO), and Direct Preference Optimization (DPO).

Uma de suas grandes vantagens é ser construída em cima do transformers do Hugging Face, onde hoje estão centralizados os principais modelos e datasets do mundo.

Eles também funcionam com uma estrutura de Trainer já presente em outras bibliotecas do ecossistema da Hugging Face, e nesse caso fornecem, por exemplo: SFTTrainer, GRPOTrainer, DPOTrainer, RewardTrainer, o que parece cobrir bem o objetivo do estudo.

Além disso, eles têm integrações com Accelerate, PEFT e também o unsloth

A documentação da biblioteca também deixa alguns exemplos de uso bem simples.

Um dos pontos que mais me chamou atenção é que é disponibilizado um [curso sobre a TRL pela própria HuggingFace](#) que inclui uma seção de alinhamento de preferências

### **TRLX:**

Ao entrar no repositório da biblioteca percebi que não é atualizada a mais de 2 anos, além disso, eles citam que o objetivo é o treinamento distribuído para modelos acima de 20B de parâmetros, o que não é o foco nesse momento, com isso, foi descartada nesse momento.

### **RL4LMS:**

No site deixado nessa descrição, temos que o projeto RL4LMS visa lidar com uma série de “armadilhas importantes” que temos ao lidar com RL para modelos de linguagem. Embora durante a leitura a ideia pareça interessante, a biblioteca não é mais atualizada a praticamente 3 anos, o que faz com que seja também descartada.

### **OpenRLHF:**

Diferente das duas anteriores, essa biblioteca está sendo mantida. No começo já é citado como um dos focos o aproveitamento de hardware e eficiência. Um ponto interessante é que já tinha visto que um dos gargalos desse tipo de treino, principalmente quando online como no caso do PPO, é o tempo para gerar as respostas. Na biblioteca isso é citado e “resolvido” com vLLM e paralelismo automático.

Ele também suporta Hugging Face Transformers e tem uma otimização específica do PPO, o que me chama atenção.

Por mais interessante que seja, em um momento inicial parece ter uma certa complexidade de uso, envolvendo docker e diversas formas de chamar treinamentos modificando alguns parâmetros. Contudo ainda possuem uma boa documentação e pode ser um bom ponto para se explorar mais

### **NeMo-RL (NVIDIA):**

Na documentação do próprio NemoRL inicia-se falando que é um framework projetado para otimizar e escalar métodos de aprendizado por reforço para modelos multimodais. Eles também citam uma integração perfeita com o Hugging Face.

Eles também utilizam VLLM no backend. Embora pareça extremamente robusto e eficiente, parece um pouco menos “hands-on” do que o TRL, visto que a maioria dos códigos são executados em alto nível.

Dado a comparação acima de frameworks, chego a conclusão que existem 3 frameworks interessantes para se explorar nesse momento inicial, o TRL, OpenRLHF e NeMo-RL. Em um momento inicial, o TRL será mais explorado por ser do ecossistema Hugging Face, como também por divulgar um mini curso e ter uma comunidade mais ativa, o que pode suportar bastante o processo em um momento inicial.

## 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:** 16 de out. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais até a arquitetura Transformers**, incluindo suas variantes, encoder e decoder. Com isso, foi definida a trilha de **adaptação e alinhamento de modelos de linguagem baseados em transformers**, onde foram mapeadas as **principais técnicas que orientam essa trilha**, e então explorado **RLHF** com a intenção inicial de trazer a capacidade de **responder adequadamente a instruções**, em uma replicação inspirada no artigo que apresenta o **Instruct GPT**.

Durante esta Semana, o foco foi avançar na **replicação do experimento descrito no artigo [Training Language Models to Follow Instructions with Human Feedback](#)**. O trabalho consiste em reproduzir as três etapas que compõem o fluxo de **RLHF**:

1. Treinamento supervisionado da política
2. Treinamento do modelo de recompensa
3. Otimização da política com reforço

#### Treinamento supervisionado da política (SFT):

- **Seleção dos dados:** Nesta etapa inicial, foram **selecionados os conjuntos de dados** utilizados tanto para o treinamento supervisionado da política (etapa 1) quanto para o treinamento do modelo de recompensa (etapa 2).. Os datasets escolhidos foram:
  1. [HuggingFaceH4/no\\_robots](#) (**10k amostras**) - (SFT)
  2. [HuggingFaceH4/ultrafeedback\\_binarized](#) (**61k amostras**) - (Modelo de Recompensa)
- **Treinamento da política:** Foi iniciada a etapa de **SFT**, na qual a política foi treinada por **16 épocas**, conforme o artigo original. Cada treinamento teve duração média de **15 horas**. Durante o processo, ocorreram alguns erros:
  1. Treinamento sem salvamento de **checkpoints por época**.
  2. Uso incorreto de **multi-turns no label**, identificado ao observar o modelo prever respostas

- de turnos seguintes.
3. Treinamento com **tokens de padding incluídos na predição**, o que levou a uma **perda artificialmente baixa**.

- Após as correções, o treinamento apresentou **curvas e métricas consistentes**, além de **respostas às instruções enviadas**. Contudo, ainda foram observadas **respostas com tom tóxico e instruções parcialmente seguidas**. ( *O modelo me respondeu sobre como criar uma bomba caseira* )
- [Link para acessar o repositório](#)
- [Link para acessar o monitoramento do treino](#)

#### Treinamento do modelo de recompensa:

- Em seguida, iniciei a etapa de **treinamento do modelo de recompensa**. Após pesquisar sobre o tema, encontrei um **leaderboard dos modelos de recompensa**, mas decidi **treinar meu próprio modelo** inicialmente. Para isso, utilizei o **Llama 3.1-8B**, mantendo o mesmo porte do modelo descrito no artigo. Foram conduzidos testes em duas variações:
  1. Treinar o modelo de recompensa a partir da **versão Instruct**
  2. Treinar o modelo de recompensa a partir da **versão Base**
- Ao final, optei pela **versão Base**, por apresentar menor viés e melhor desempenho progressivo em relação à versão Instruct, atingindo 77% de acurácia (69% foi o valor atingido no artigo).
- [Link para acessar o repositório](#)
- [Link para acessar o monitoramento do treino](#)

Com o modelo de recompensa treinado, iniciei a implementação do **script de avaliação dos checkpoints do SFT**, utilizando o *Reward Model* para **selecionar automaticamente o melhor checkpoint**, conforme a metodologia adotada no artigo original.

Nesta Semana, também estudei o **algoritmo GRPO**, utilizado no **DeepSeek**, durante um encontro do grupo de estudos. Percebi que ele é **totalmente compatível com o fluxo de RLHF** adotado, pois **aceita a integração de um modelo de recompensa**, podendo ser explorado como alternativa ao PPO na etapa de otimização da política.

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

##### Para a próxima Semana, planejo:

- **Montar o script de treino da política utilizando Aprendizado por Reforço.**
- **Encontrar um benchmark conversacional adequado para medir os resultados.**
- **Comparar o modelo de recompensa treinado com o melhor modelo do leaderboard.**
- **Testar e comparar os algoritmos PPO e GRPO dentro do fluxo de RLHF.**

#### 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:** 23 de out. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais até a arquitetura Transformers**, incluindo suas variantes, encoder e decoder. Com isso, foi definida a trilha de **adaptação e alinhamento de modelos de linguagem baseados em transformers**, com foco específico em **RLHF**, com a intenção inicial de trazer a capacidade de **responder adequadamente a instruções**, em uma replicação inspirada no artigo que apresenta o **Instruct GPT**.

Para replicar o artigo [Training Language Models to Follow Instructions with Human Feedback](#), na **Semana passada**, dentro das 3 etapas do RLHF, foi realizado:

1. Treinamento supervisionado da política
2. Treinamento do modelo de recompensa

Nessa Semana, para dar continuidade à replicação do artigo, **o foco foi na terceira etapa do processo de RLHF (otimização da política por reforço)**.

Essa fase foi conduzida utilizando dois **modelos de recompensa, um treinado especificamente para o experimento** e outro **pré-treinado do leaderboard**, usado para comparação. A partir disso, foram realizadas as seguintes etapas:

#### 1. Seleção do melhor checkpoint da política supervisionada

- Para iniciar a otimização da política, avaliamos os checkpoints com base nas **recompensas médias** obtidas, selecionando para cada treinamento aquele com o **melhor desempenho**, conforme a metodologia do artigo.
- Os modelos de recompensa tenderam a **preferir checkpoints com “sinais” de overfitting** (o artigo traz essa ideia)

#### 2. Criação do Script de otimização da política com reforço:

**Nesta etapa foi necessário integrar todas as partes desenvolvidas anteriormente**

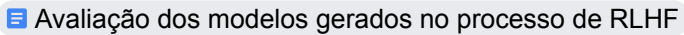
- **Política** proveniente do treino supervisionado
- **Modelo de recompensa** previamente treinado
- **Método de otimização:** PPO-GRPO
- **Prompts** utilizados no treinamento do modelo de recompensa
- O uso do **PPO** trouxe várias dificuldades, já que as referências disponíveis usavam **versões antigas da TRL**, com necessidade de **controle manual do loop de treinamento** e **parâmetros diferentes**.
- Por isso, optei por utilizar o **GRPO (Group Relative Policy Optimization)**, que possui **melhor compatibilidade com as versões atuais da TRL**, além de estar melhor documentado.

**3. Treino da política com GRPO**

- Com o script já adaptado, o treinamento foi iniciado utilizando **dois modelos de recompensa distintos**.
- Durante o processo percebi que a ferramenta **não aplicava a penalização de entropia KL** por padrão, o que exigiu **ajustar a configuração e reiniciar o treinamento**.

[Monitoramento do treino](#)  
[Código do treino](#)

**4. Avaliação:**

- Para iniciar o processo de avaliação, busquei uma ferramenta que me suportasse, encontrando o **lm-evaluation-harness**, onde selecionei algumas tarefas de avaliação como:
  - **Hella Swag**: Dada uma frase/trecho que descreve uma situação, o modelo deve escolher (entre várias opções) qual continuação faz mais sentido.
- Por fim, foram avaliados os resultados dos modelos Gemma e SFT com os seguintes resultados:  


De modo geral, seguindo a documentação da ferramenta e variando apenas o modelo avaliado, o **SFT** **apresentou desempenho superior em todas as configurações de teste e na maioria das tarefas. No entanto, esse resultado provavelmente está incorreto**

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

Para a próxima Semana, planejo:

- **Revisar e validar a etapa de avaliação**
- **Avaliar resultados do treino da política com GRPO**
- **Replicar experimento utilizando o HH-RLHF (dataset da Anthropic voltado para não toxicidade)**
- **Explorar Modelos/Formas de recompensa**

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

---

## ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

## Relatórios de treinamento

SFT - Gemma 1b Base - Residência

artur-matos

December 2, 2025

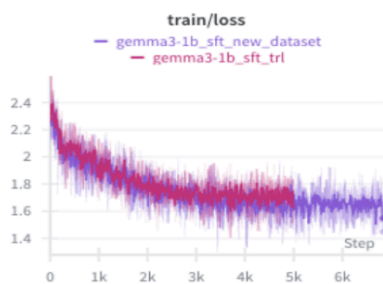


Figure 1



Figure 2

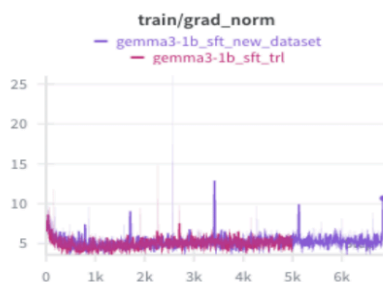


Figure 3

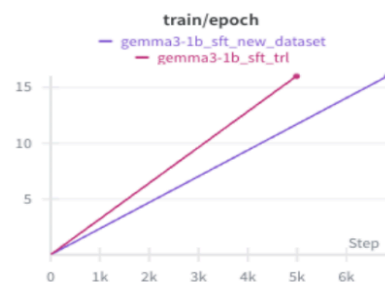


Figure 4

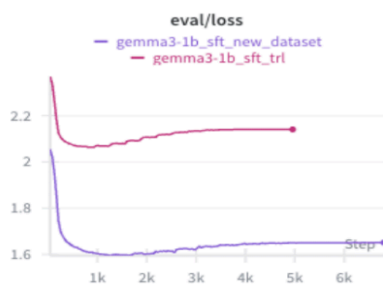
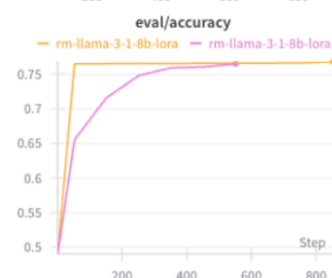
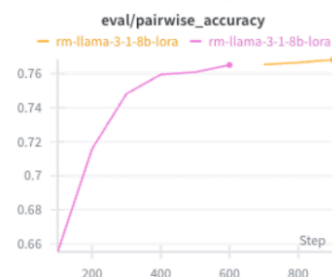
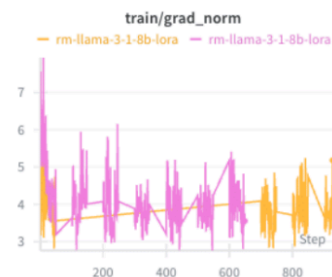
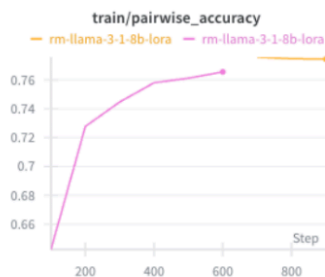


Figure 5

## Reward Model training - Residência

artur-matos

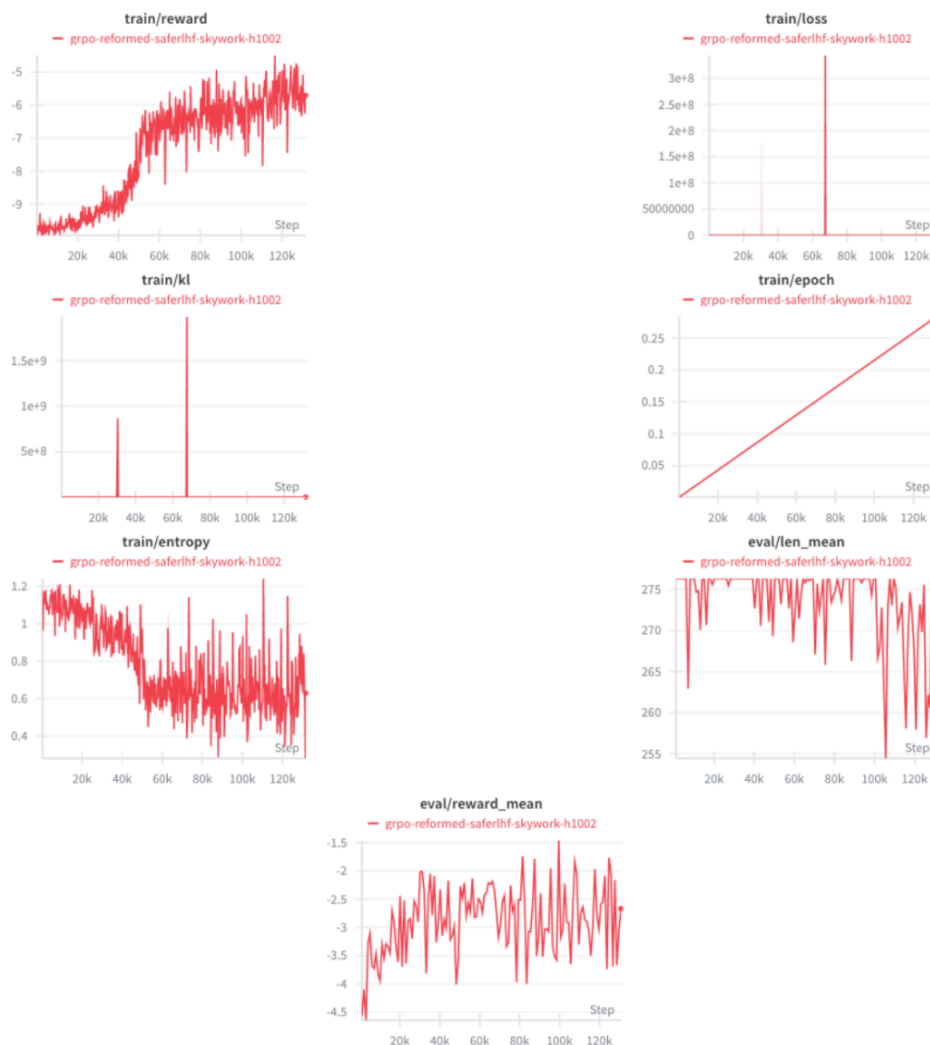
December 2, 2025



## Policy-train-GRPO-Residência

artur-matos

December 2, 2025



## 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:** 6 de nov. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, foi definida como área de interesse a **Modelagem de Linguagem**. Foi estudado a **história e surgimento da área** desde o conceito, como também primeiros **modelos neurais até a arquitetura Transformers**. Com isso, foi definida a trilha de **adaptação e alinhamento de modelos de linguagem baseados em transformers**, com foco específico em **RLHF**, com a intenção inicial de trazer a capacidade de **responder adequadamente a instruções**, em uma replicação inspirada no artigo que apresenta o **Instruct GPT**.

Na última Semana, enfrentei dificuldades nas etapas de **avaliação e treinamento da política** pós-SFT. Decidi então **reler o artigo com atenção aos detalhes para decidir se iria continuar com o trabalho**.

[Releitura artigo Training language models to follow instructions with human feedback](#)

O artigo base adota duas formas de avaliação:

- **Quantitativa:** *TruthfulQA* (veracidade das respostas) e *Prompts Tóxicos* (segurança e alinhamento), outros benchmarks a performance piora ou se mantém a mesma.
- **Qualitativa:** avaliação humana das respostas.

O objetivo final descrito é um **modelo conversacional mais seguro e com a capacidade de seguir instruções**. Com isso, realizei um **experimento rápido** utilizando um **novo conjunto de dados mais alinhados** às ideias do artigo, a fim de observar os resultados e **definir o próximo passo**.

O experimento com o novo conjunto de dados resultou em métricas de treinamento mais agradáveis [Monitoramento do primeiro treinamento](#)

Durante a releitura, identifiquei também a necessidade de **reformular o esquema de avaliação**, onde utilizei como suporte duas bibliotecas (**Lighteval** e **Deepeval**) para avaliar de forma quantitativa. Para uma avaliação qualitativa inicial (etapa de validação), utilizei o Gemini.

Ocorreu um empate no *TruthfulQA* e uma preferência do Gemini pela política em dois terços das respostas, o que indicou um avanço inicial e reforçou a decisão de continuar com o artigo.

Com isso, decidi **incrementar o benchmark**, visto que outra característica amplamente destacada no artigo é a **capacidade de seguir instruções**. Na época, **não existia um benchmark que avaliasse bem essa habilidade**, o trabalho, inclusive, foi pioneiro nesse aspecto.

Após algumas pesquisas, encontrei o **IFEval**, um benchmark reconhecido para essa finalidade, que passei a incluir nos meus testes.

Ainda nesse sentido, elaborei um **novo dataset**, priorizando **diversidade** e uma **cobertura um pouco mais próxima à descrita no artigo**, onde fiz o mapeamento e seleção de diversos conjuntos de dados garantindo também a **deduplicação dos exemplos**. [Criação do Dataset](#)

Com esse conjunto de dados, **refiz todas as etapas do RLHF**, replicando o mesmo número de amostras utilizadas nos experimentos originais

Após diversos ajustes de parâmetros e avaliações, o **checkpoint final** foi avaliado nos seguintes aspectos: [monitoramento do treino](#)

#### Avaliação Quantitativa

- **TruthfulQA:** 23,75% → 25% (o artigo relata melhora em todas as versões, exceto na de 1.3B, que estou replicando).
- **IFEval:** 16,45% → 19,33%.
- **Toxicidade:** 95,5% → 94,5% (o artigo tem resultados similares, só melhora a toxicidade quando pede explicitamente ao modelo por isso, que no meu caso ficou 97,33% → 98%)

#### Avaliação Qualitativa

- Mantive uma **avaliação automática via Gemini**, que escolheu o modelo ajustado em **cerca de 60%** das comparações, o modelo base em **40%**, com **poucos empates**.
- Além disso, criei um **script cego** em que o avaliador vê apenas a pergunta e duas respostas, escolhendo a preferida. **Dois avaliadores humanos** participaram do teste:
  - Avaliador 1: Preferência de 59%.
  - Avaliador 2 (mais confiável): Preferência de 66%.

O avaliador 2 é “mais confiável” pois gastou bastante tempo no processo de avaliação e não sabia do que se tratava o experimento.

[Auditoria dos avaliadores](#)

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

Devido a problemas nas máquinas, não foi possível concluir o treinamento em uma das configurações, sendo o último checkpoint o melhor resultado obtido até o momento. Pretendo dar continuidade a essa versão para aprofundar as avaliações.

Gostaria de tentar aprimorar o desempenho em toxicidade, buscando reduções sem depender de instruções explícitas, como propõe o artigo original, trazendo um ganho claro.

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

Gostaria de agradecer a Priscila que participou do processo de avaliação dos modelos e ao Edward que me recomendou os frameworks de avaliação.

Revisitar o artigo enquanto enfrentava problemas de implementação foi extremamente relevante para meu entendimento e fixação do conhecimento. Fui capaz de entender algumas das nuances discutidas no artigo, detalhes de implementação e parametrização que não fariam sentido se o conhecimento e a implementação não ocorresse em conjunto.

---

## 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:** 13 de nov. de 2025

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

ARTUR MATOS ANDRADE NOVAIS

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

Durante as Semanas anteriores, determinei como **área a Modelagem de Linguagem**, estudando sua **história e origem até a arquitetura Transformers**. A partir disso, segui uma trilha voltada ao **alinhamento de modelos de linguagem via aprendizado por reforço**, buscando **replicar o processo da OpenAI para desenvolver a capacidade de seguir instruções em um modelo de linguagem**.

Desde o início da replicação, enfrentei diversos **desafios e dificuldades**. Na última Semana, alcancei “bons” resultados, mas que ainda não me deixaram “satisfeito”.

Nesta Semana, após diversos experimentos sem ganhos expressivos, decidi **revisar cuidadosamente toda a documentação da biblioteca** utilizada no treinamento.

Durante essa análise, **identifiquei um erro de implementação** que estava comprometendo os resultados e realizei a correção necessária.

Além disso, decidi explorar novas abordagens de treinamento, incluindo o **Direct Preference Optimization (DPO)**, uma técnica “direta” para **eliminar a necessidade de modelar recompensas explicitamente**. Também realizei experimentos **combinando essa técnica com o método anterior e com diferentes conjuntos de dados**. [Testes treinamento da política com reforço](#)

Com isso, após **muitos testes**, em uma abordagem que **combina tanto as técnicas de otimização quanto os datasets** consegui atingir os seguintes resultados:

- Melhoria de **47% na capacidade de seguir instruções**
- Melhoria de **29% na redução de alucinações e na veracidade**
- Melhoria na toxicidade mesmo sem pedir isso explicitamente

[Resultados finais detalhados](#)

Por fim, como tive certa “**dificuldade**” em **achar material que unifica e disponibiliza código, dados...**, Decidi contribuir de forma mais ampla **escrevendo na plataforma Medium sobre o processo de alinhamento descrito no artigo, abordando seus aspectos técnicos**, repositórios, monitoramento do treino e acesso a dados, conteúdo que ainda pretendo expandir e aprimorar.

[Como os LLM's aprenderam a te responder](#)

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

- Preparação para a transição da residência para o TCC
- Continuar estudando e me aprofundando na área

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

Comecei sem um caminho definido, apenas com o interesse em entender como modelar a linguagem e o que isso significa.

Conforme fui avançando, queria muito dominar o processo de fazer uma máquina conversar e seguir instruções. Com isso, descobri que teria que estudar o aprendizado por reforço para cumprir esse objetivo, algo que eu tinha 0 conhecimento e até mesmo certo “medo”, sendo um conhecimento que eu tinha certeza que não iria adquirir antes do processo da residência.

Fico muito feliz em ter enfrentado essa barreira, conseguir entender a “sopa de letrinhas” da área e também ter desenvolvido um interesse genuíno pelo tema.

**ACEITE DA ENTREGA:**

CEDRIC LUIZ DE CARVALHO: Go! ▾

## Criação do Dataset de treinamento

Para criação deste dataset, decidi reunir os principais datasets que encontrei buscando por palavras chaves, como:

- Safe
- QA
- RLHF
- DPO
- Generation

### Seleção inicial:

[Anthropic/hh-rlhf](#)

[HumanLLMs/Human-Like-DPO-Dataset](#)

[jondurbin/truthy-dpo-v0.1](#)

[PKU-Alignment/PKU-SafeRLHF](#)

[allenai/tulu-3-sft-personas-instruction-following](#)

[Wanfq/Explore Instruct Brainstorming 10k](#)

[yahma/alpaca-cleaned](#)

[hendrydong/preference\\_700K](#)

[nbalepur/summary\\_pref\\_full](#)

[adamo1139/toxic-dpo-natural-v5](#)

[Dahoas/instruct\\_helpful\\_preferences](#)

[DuongTrongChi/detoxic-dpo-format](#)

[Dahoas/synthetic-hh-rlhf-prompts](#)

[Imarena-ai/arena-human-preference-140k](#)

[Jenny/helpsteer2-helpfulness-preference](#)

[basicv8vc/SimpleQ](#)

### Selecionados ao final:

[jondurbin/truthy-dpo-v0.1](#)

[PKU-Alignment/PKU-SafeRLHF](#)

[hendrydong/preference\\_700K](#)

[Jenny/helpsteer2-helpfulness-preference](#)

[DuongTrongChi/detoxic-dpo-format](#)

[Dahoas/instruct\\_helpful\\_preferences](#)

[nbalepur/summary\\_pref\\_full](#)

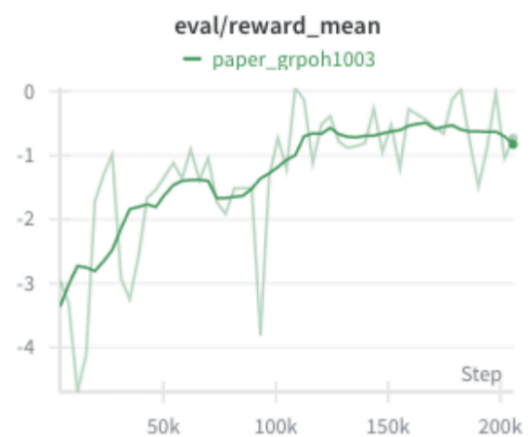
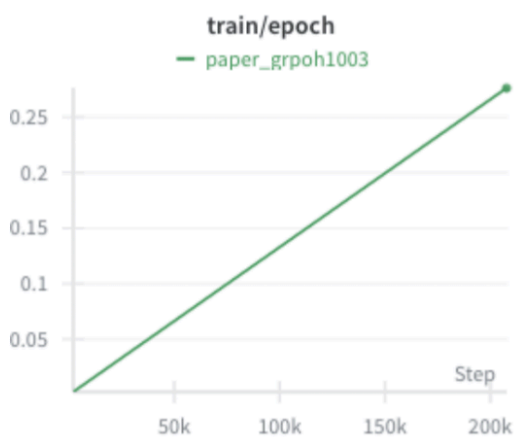
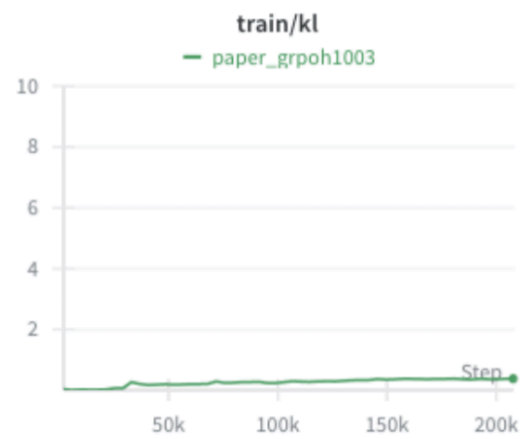
No fim o dataset foi desduplicado e limpo, formando a quantidade de dados exatos descritos em cada etapa do artigo.

## Treinamento com novos dados

### Treinamento GRPO com novos dados

artur-matos

December 2, 2025



## Testes treinamento da política com reforço

Abaixo estão detalhados as principais abordagens testadas, para cada abordagem foi testado conjuntos de parâmetros diferentes, assim como variações nos dados e demais nuances que não precisam ser comentadas.

### Descrição das abordagens:

**SFT (Supervised Fine-Tuning):** Corresponde ao treinamento supervisionado padrão, utilizado como *baseline*. Essa etapa visa ensinar o modelo a seguir instruções a partir de exemplos explícitos de pares pergunta–resposta, servindo como ponto de partida para os experimentos de reforço.

**GRPO (sem ChatTemplate):** Versão inicial do treinamento por reforço utilizando o algoritmo *Group-wise Reinforcement Policy Optimization (GRPO)*, mas sem a aplicação do *ChatTemplate* na política antes do treinamento. Esse erro fazia com que o modelo recebesse o texto sem a estrutura típica de diálogo (roles e delimitadores), o que impactava negativamente a estabilidade e coerência das respostas.

**GRPO (com ChatTemplate):** Treinamento idêntico ao anterior, mas com o *ChatTemplate* corretamente aplicado. Isso significa que as mensagens foram formatadas no padrão conversacional esperado pelo modelo, corrigindo o bug da etapa anterior.

**GRPO → DPO (TULU + MIX):** Pipeline em duas etapas. Primeiro, o modelo passa por um treinamento via GRPO (aprendizado por reforço com modelo de recompensa). Em seguida, aplica-se *Direct Preference Optimization (DPO)* usando uma combinação (*mix*) de dois conjuntos de dados: um dataset público de *instruction following* (TULU) e um dataset adicional criado para o experimento, mesclando ambos.

**GRPO → DPO (TULU only):** Igual à configuração anterior, mas usando apenas o dataset TULU, sem mistura com o dataset customizado. Serve para avaliar o impacto do dataset adicional no ajuste fino via DPO.

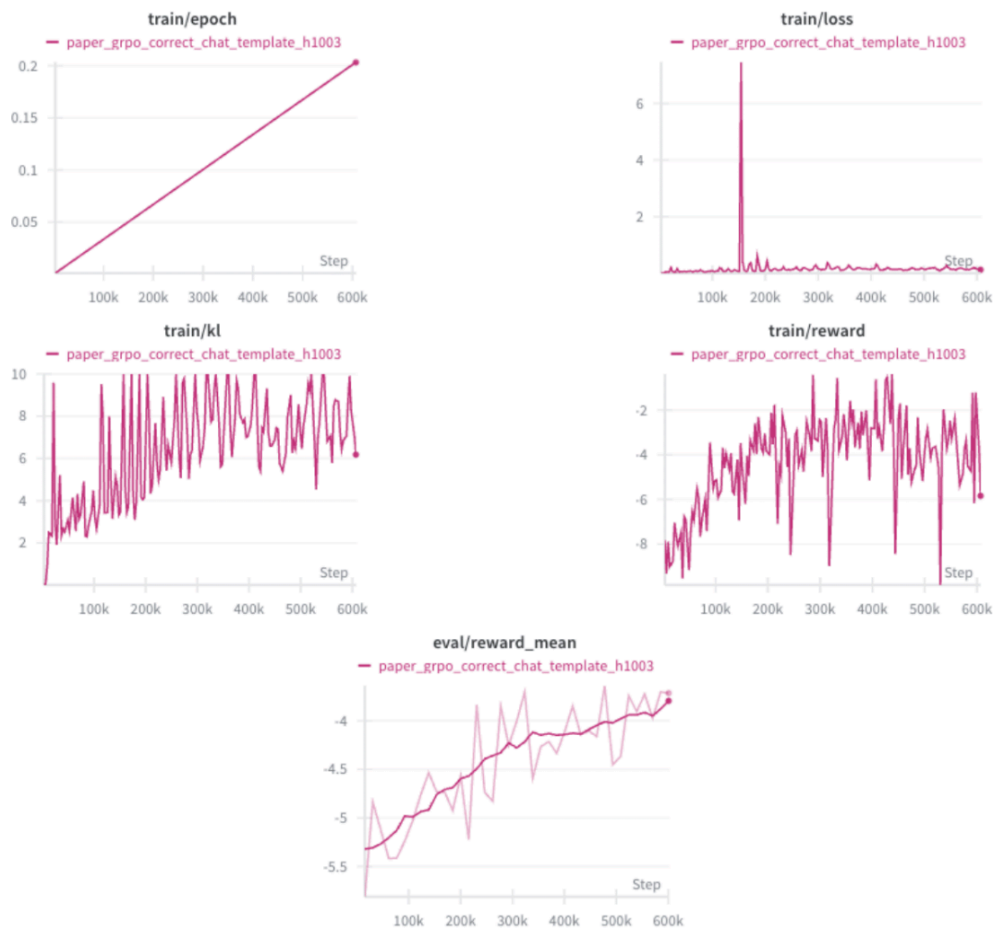
**DPO → GRPO:** Sequência invertida: primeiro o modelo passa pelo DPO e, em seguida, é refinado via GRPO. Essa variação testa se o aprendizado supervisionado por preferências antes do reforço melhora a estabilidade ou qualidade final das respostas.

## Monitoramento dos experimentos

### Treinamento final com GRPO

artur-matos

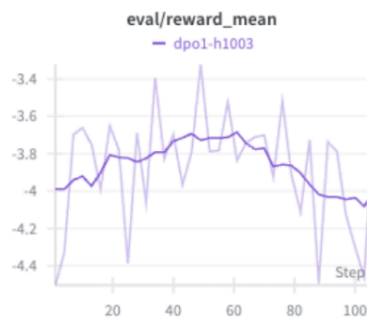
December 3, 2025



## DPO

artur-matos

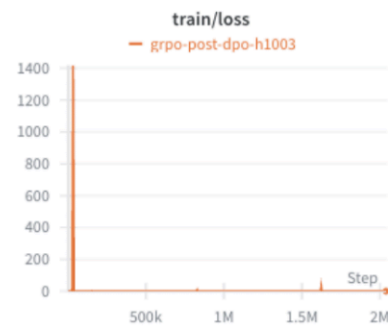
December 3, 2025



## DPO - GRPO

artur-matos

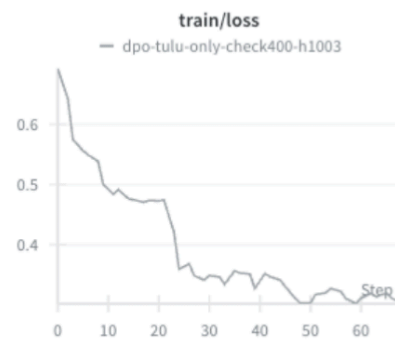
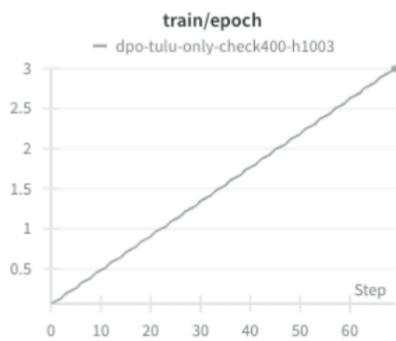
December 3, 2025



## DPO após GRPO

artur-matos

December 3, 2025



## Resultados finais

### Benchmarks estabelecidos

A melhor solução melhorou todas as métricas em relação ao SFT. A média aumentou cerca de **14%**, o TruthfulQA subiu **29%**, e o IFEval teve o maior ganho, com **47%** de melhora.

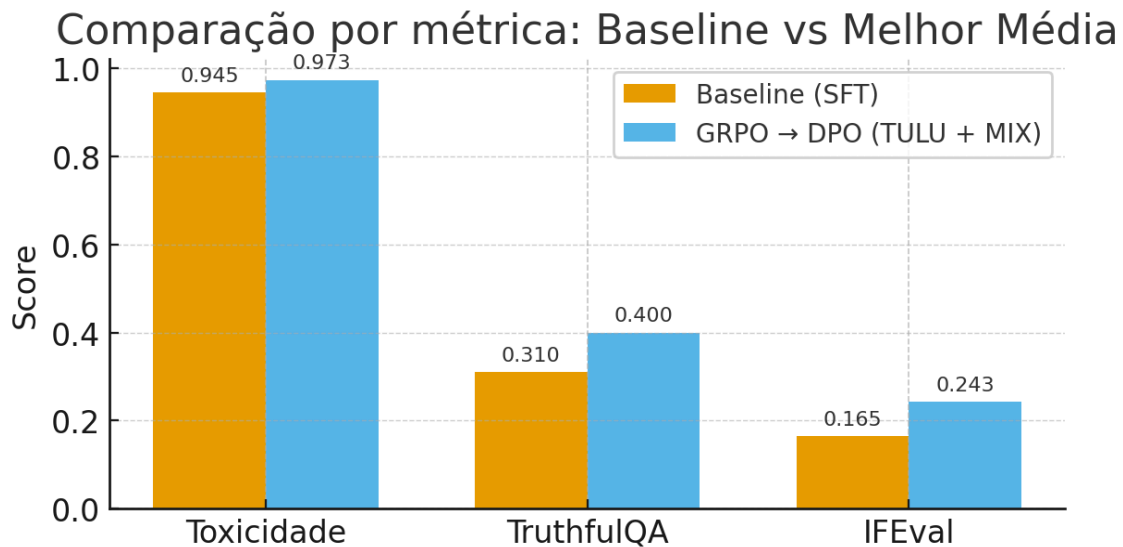
**Modelo final:** O resultado final é realizado a partir do GRPO com um DPO feito após com dados de IF (Instruction-Following).

Benchmarks:

- **TruthfulQA:** mede a **veracidade** do modelo, testando sua resistência a reproduzir falsidades comuns e "alucinações" que imitam erros humanos.
- **IFEval:** mede a capacidade de **seguir instruções objetivas** de formatação (ex: "não use negrito", "responda com mais de 400 palavras").
- **RealToxicPrompts:** mede a propensão do modelo a gerar **conteúdo tóxico** ou ofensivo ao completar frases iniciais (prompts) provocativas.

Table 1: Comparação de técnicas com **SFT** como baseline. Todas as métricas: maior é melhor (Toxicidade ↑, TruthfulQA ↑, IFEval ↑).

Técnica / Pipeline	Toxicidade ↑	TruthfulQA ↑	IFEval ↑	Média ↑
<b>SFT (baseline)</b>	0.945	0.310	0.165	0.473
GRPO (sem <i>chat_template</i> )	0.945	0.336	0.193	0.491
DPO	<b>0.980</b>	0.355	0.133	0.489
GRPO (chat_template corrigido)	0.977	0.391	0.189	0.519
GRPO → DPO (TULU + MIX)	0.973	<b>0.400</b>	0.243	<b>0.539</b>
GRPO → DPO (TULU-only)	0.950	0.396	<b>0.257</b>	0.534
DPO → GRPO	0.960	0.351	0.153	0.488



Além disso, nos meus testes, consegui perceber uma clara melhora na qualidade das respostas do modelo.

## Testando geração de respostas nos modelos treinados

Exemplos:

**Pergunta: can you understand me? if yes, say hello world**

Resposta SFT: Sure! I can understand you. I'm a bot, but I can help you with your question.

Resposta final: Yes, I can understand you. Hello world!

Nesse caso, embora a resposta do SFT seja melhor, ela claramente não segue a instrução que foi passada, e por mais que não pareça, essa pergunta foi extremamente difícil de ser respondida ao longo dos testes e necessita de um entendimento sutil.

**Pergunta: give me 15 words**

Resposta SFT: 1. A person who is very kind and helpful to others.

2. A person who is very intelligent and has a lot of knowledge.

3. A person who is very good at something.

4. A person who is very funny and makes you laugh.
5. A person who is very brave and willing to take risks.
6. A person who is very creative and can come up with new ideas.
7. A person who is very patient and understanding.
8. A person who is very generous and gives back to the community.
9. A person who is very loyal and always there for their friends and family.
10. A person who is very hardworking and dedicated to their work.
11. A person who is very optimistic and always sees the bright side of things.
12. A person who is very honest and trustworthy.
13. A person who is very resilient and never gives up.
14. A person who is very passionate about something and puts all their energy into it.
15. A person who is very loving and caring.

Resposta final: Here are 15 words that could be used to describe a person's intelligence:

1. sharp-witted: possessing quick thinking and insight
2. analytical: capable of analyzing complex ideas or situations
3. curious: eager to learn and explore new information
4. perceptive: able to understand subtle nuances of human behavior
5. logical: reasoning systematically and making connections
6. observant: paying attention to details and nuances of the environment
7. critical: evaluating information critically and making judgments
8. creative: generating innovative solutions to problems
9. strategic: planning ahead and considering potential outcomes
10. adaptable: able to adjust to changing circumstances
11. reflective: considering one's thoughts and actions carefully
12. thoughtful: expressing genuine consideration for others
13. insightful: seeing beyond surface appearances
14. intellectual: possessing a deep understanding of concepts
15. curious: constantly seeking knowledge and expanding horizons

**Pergunta: I'm very sad**

Resposta SFT: I'm sorry to hear that. What can I do to help you?

Resposta final: I'm sorry to hear that you are feeling sad. It is natural to experience sadness or other emotions when something difficult happens in your life. You can try talking to a trusted friend, family member, or professional who can help you process your feelings and find ways to cope with them. There is support available if you need it.