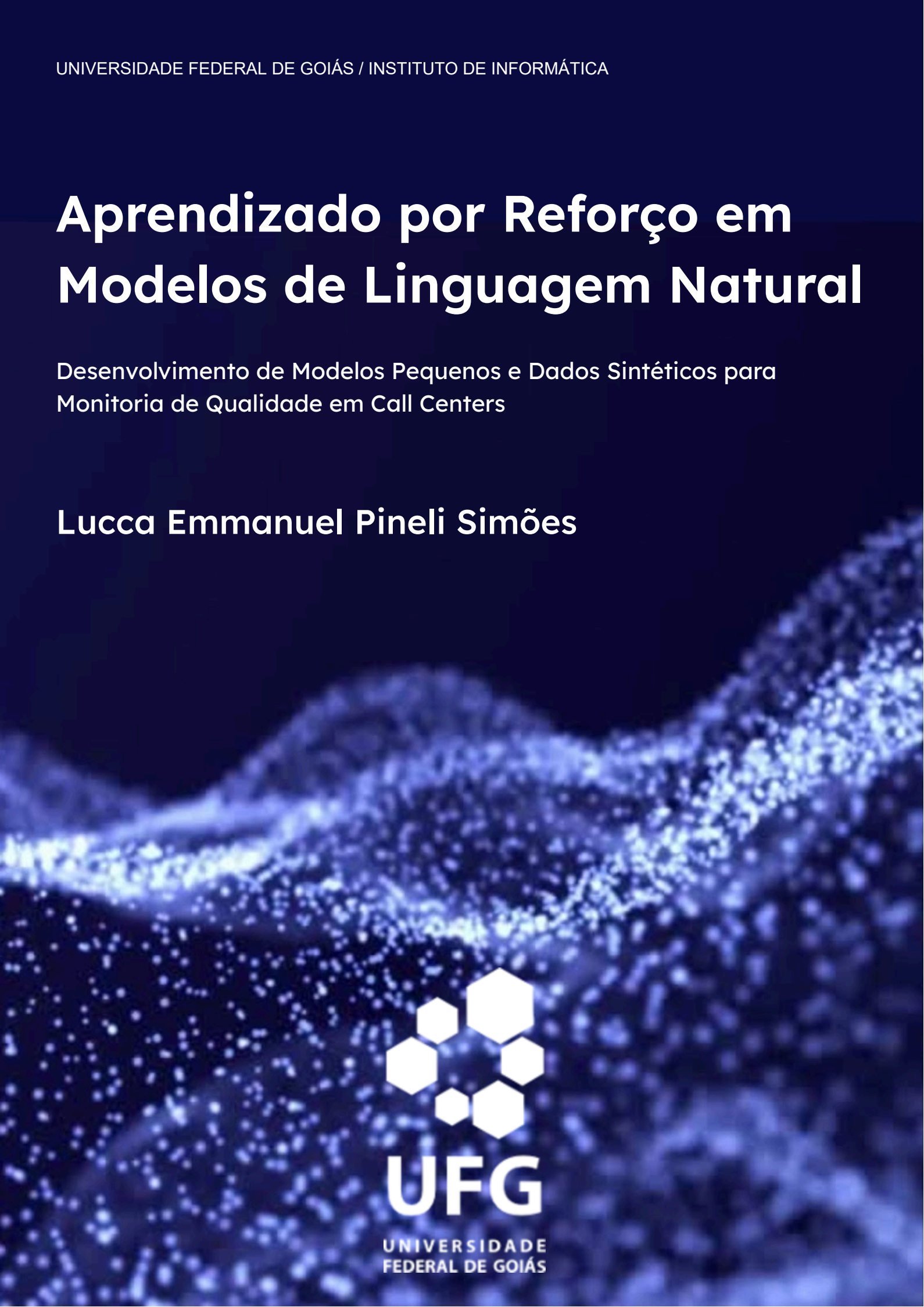
UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Aprendizado por Reforço em Modelos de Linguagem Natural

Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

Lucca Emmanuel Pineli Simões



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

LUCCA EMMANUEL PINELI SIMÕES

Aprendizado por Reforço em Modelos de Linguagem Natural

Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

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): LUCCA EMMANUEL PINELI SIMÕES

Título do trabalho: Aprendizado por Reforço em Modelos de Linguagem Natural

Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

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

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

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

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

Documento assinado eletronicamente por **Lucca Emmanuel Pineli Simoes**, **Discente**, em 15/01/2025, às 16:53, 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 16/01/2025, às 18:31, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089795** e o código CRC **9B53E9CC**.

Referência: Processo nº 23070.001594/2025-12 SEI nº 5089795

LUCCA EMMANUEL PINELI SIMÕES

Aprendizado por Reforço em Modelos de Linguagem Natural

Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

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.

SIMÕES, LUCCA EMMANUEL PINELI
Aprendizado por Reforço em Modelos de Linguagem Natural [manuscrito] : Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers / LUCCA EMMANUEL PINELI SIMÕES. - 2025.
83 f.

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. aprendizado por reforço. 3. modelos de linguagem. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

LUCCA EMMANUEL PINELI SIMÕES

Aprendizado por Reforço em Modelos de Linguagem Natural
Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

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: 17 de dezembro de 2024.

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)

Luiz Guilherme Corrêa
(CEIA-UFG)

LUCCA EMMANUEL PINELI SIMÕES

Aprendizado por Reforço em Modelos de Linguagem Natural

Desenvolvimento de Modelos Pequenos e Dados Sintéticos para Monitoria de Qualidade em Call Centers

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Aprendizado por Reforço com Feedback Humano (RLHF)**. 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 grandes de linguagem, geração automática de datasets.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Reinforcement Learning with Human Feedback (RLHF)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

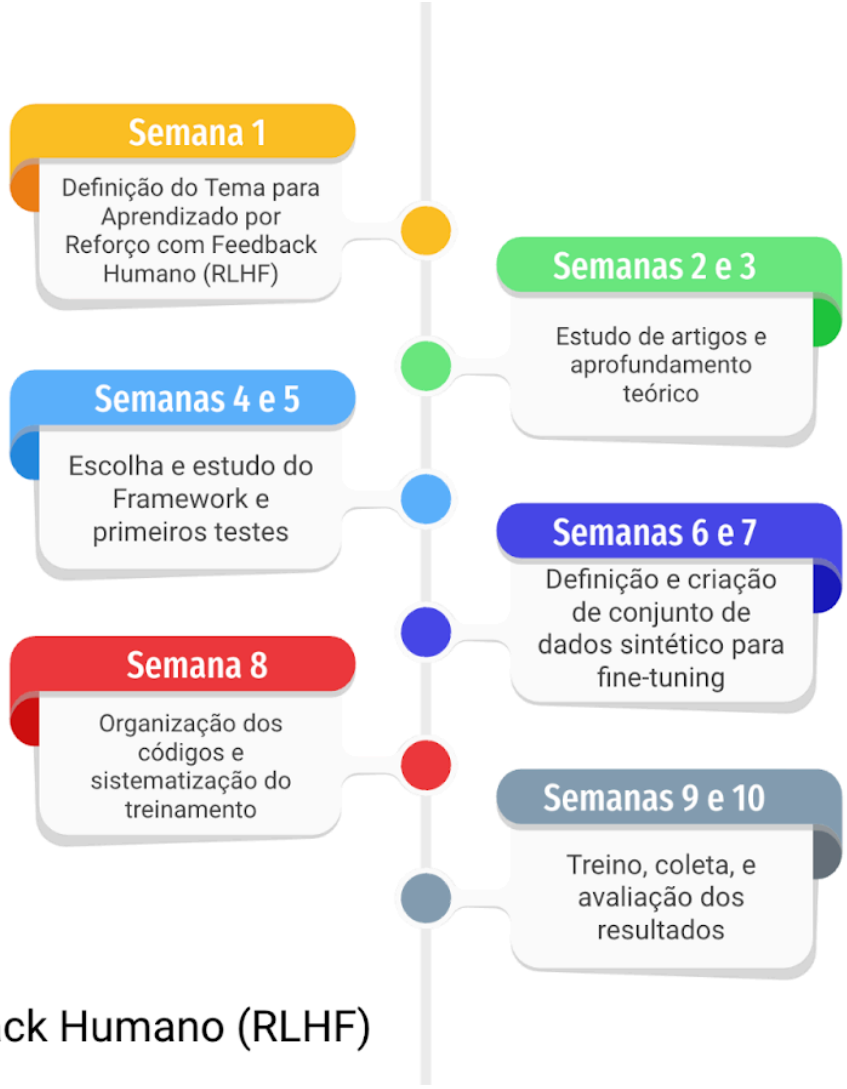
Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada



Lucca Emmanuel Pineli Simões

Especialista em: Aprendizado por Reforço com Feedback Humano (RLHF)

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

MINHA JORNADA

Nome: **Lucca Emmanuel Pineli Simões**

Especialidade: **Aprendizado por Reforço com Feedback Humano (RLHF)**

Objetivo deste documento

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

Minha Jornada

Minha jornada iniciou-se na **Semana 1** com a definição do tema a ser explorado em minha especialização. Decidi aprofundar-me na área de NLP + RL para monitoria automática de qualidade em call centers. Realizei buscas de artigos em bibliotecas digitais, definindo strings de busca baseadas no tema escolhido e selecionando artigos relevantes pelo título e resumo. Essa fase incluiu a leitura e estudo de diversos artigos, cujos detalhes e resumos estão disponíveis no **Apêndice 1**. Destaco a importância do aprofundamento inicial nos conceitos de RLHF (Reinforcement Learning from Human Feedback) e na compreensão da formalização matemática por trás desse método, que serviram como alicerce para os estudos subsequentes.

Nas **Semanas 2 e 3**, dei continuidade ao aprofundamento nos surveys identificados anteriormente, mapeando métodos e suas qualidades. Realizei análises detalhadas dos algoritmos e refinei a definição do problema central: “Melhorar as avaliações de qualidade de atendimentos de call center”. Questionei aspectos fundamentais como o que constitui uma

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.

avaliação, como medi-la e aprimorá-la, e como validar a eficácia da medição. A formalização matemática do problema, evidenciada no **Apêndice 2**, confirmou tratar-se de um problema adequado para solução via aprendizado por reforço. Com isso, planejei explorar soluções aplicadas em contextos similares, visando mapear as melhores práticas e metodologias para o meu projeto.

Do **Apêndice 3**, correspondente às **Semanas 4 e 5**, foquei na pesquisa de frameworks adequados para tarefas de RLHF. Após analisar os frameworks mais utilizados, optei pelo HuggingFace TRL devido à sua ampla variedade de algoritmos, atualização constante, facilidade de implementação e integração com métodos PEFT. Familiarizado com o PyTorch, a similaridade do TRL facilitou o desenvolvimento. Iniciei implementações preliminares de algoritmos e, conforme detalhado no apêndice, preparei o dataset da empresa para o treinamento, organizando-o em datasets de preferência e com feedback binário. Também comecei a elaborar uma solução planejada que incorpora tudo o que foi estudado até então. Esse trabalho foi crucial para a elaboração de uma prova de conceito e para o planejamento das etapas seguintes.

Nas **Semanas 6 e 7**, correspondentes ao **Apêndice 4**, concentrei-me no tratamento do dataset para adaptá-lo ao método de Binary Classification Optimization (BCO), criando tuplas de (prompt, output, label). Treinei um modelo de linguagem (llama 3.2 1b) utilizando PEFT, porém enfrentei problemas de overfitting devido ao tamanho reduzido do dataset (~1.000 amostras). Para contornar essa limitação, criei aproximadamente 130.000 amostras de dados sintéticos de transcrições. Essas amostras incluíam perguntas de sim ou não, transcrições, análises e respostas, visando prever a análise e a resposta com base na pergunta e na transcrição. Treinei um modelo utilizando Supervised Fine-Tuning (SFT), alcançando uma acurácia inicial de 68%, que aumentou para 79% após aprimoramento via aprendizado por reforço, conforme apresentado no apêndice.

Na **Semana 8**, detalhada no **Apêndice 5**, dediquei-me a sistematizar os treinamentos e resultados dos modelos. Escrevi e revisei scripts de treinamento (SFT e RL) e testes, padronizando arquivos de configuração, parâmetros e formatos. Realizei testes

mais robustos e comparáveis, treinando 14 modelos diferentes, incluindo TeenyTinyLlama, Tucano, GPT2Small, PT-T5 e BERTimbau, utilizando métodos como SFT, DPO, BCO e ORPO. Os resultados detalhados desses treinamentos estão disponíveis no apêndice, evidenciando melhorias significativas na acurácia e na consistência dos modelos avaliados.

Por fim, nas **Semanas 9 e 10**, focadas no **Apêndice 6**, concentrei-me na avaliação qualitativa das análises textuais geradas pelos modelos. Utilizei embeddings de sentença para calcular a similaridade entre as análises preditas e as respostas esperadas. Além disso, implementei um sistema de avaliação com uma LLM, que comparava a resposta predita com a desejada e atribuía uma nota de 0 a 10. Os resultados dessas avaliações estão detalhados no apêndice. Essa etapa permitiu identificar a eficácia dos modelos não apenas em termos de acurácia, mas também na qualidade das análises geradas, proporcionando insights valiosos para melhorias futuras.

Em função de tudo que vivi nesta jornada, gostaria de registrar que a integração de NLP com Aprendizado por Reforço tem um potencial transformador na automação e melhoria de processos em call centers. As experiências e desafios enfrentados ao longo deste projeto enriqueceram minha compreensão técnica e prática, reforçando a importância da pesquisa contínua e da aplicação cuidadosa de técnicas avançadas de IA em problemas do mundo real.

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: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

1. Definição do tema que será abordado
    a. NLP + RL para monitoria automática de qualidade em call centers

2. Busca de artigos em bibliotecas digitais
    a. Definição de strings de busca com base no tema escolhido
    b. Definição de bibliotecas digitais para se realizar a busca
    c. Seleção de artigos com base no título e abstract
        i. Critério para avaliação: similaridade com o tema

    - Artigos NLP + RL
    - Artigos Call Centers.docx

3. Leitura e estudo dos artigos
    - Resumo - Artigos NLP + RI
    - Resumo - Artigos Call Center

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

1. Aprofundamento nos seguintes surveys:

- **RLHF Deciphered.pdf**
- **THE RL_LLM TAXONOMY TREE.pdf**

2. Aprofundamento na formalização matemática por trás de RLHF
3. Mapeamento de métodos e suas qualidades

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumos dos Artigos em Aprendizado por Reforço

Artigo 1

O estudo de Uc-Cetina et al. [1] revisa o uso de algoritmos de aprendizado por reforço (RL) no processamento de linguagem natural (PLN), destacando que, embora o RL tenha obtido sucesso em áreas como jogos e robótica, ainda não atingiu seu potencial completo no PLN. Os autores discutem aplicações de RL em tradução automática, sistemas de diálogo e compreensão de linguagem, enfatizando como o RL pode ser integrado com redes neurais profundas, como transformers, para aprimorar políticas de ação e estados em tarefas de linguagem. Apesar de métodos como BERT e GPT mostrarem maior eficiência em muitas tarefas de PLN, o artigo sugere que o RL pode refinar modelos poderosos e ser útil em tarefas que envolvem aprendizado dinâmico e interativo, como agentes de conversação adaptativos.

Artigo 2

Cai et al. [2] apresentam uma visão geral do uso de aprendizado por reforço profundo (DRL) em tarefas de processamento e análise de dados, destacando como o DRL pode otimizar desde a preparação e limpeza de dados até análises avançadas. O DRL é aplicado em tarefas como correspondência de entidades, otimização de consultas SQL, sistemas de recomendação e publicidade digital, auxiliando na formulação de políticas ótimas para maximizar o desempenho. Os autores discutem desafios como a integração em sistemas existentes, falta de benchmarks padronizados e a interpretabilidade limitada dos modelos, especialmente em setores críticos como saúde e finanças. Sugere-se que pesquisas futuras devem focar na expansão do DRL para outros domínios e na melhoria da interpretabilidade e robustez dos modelos.

Artigo 3

Sharma e Kaushik [3] revisam o uso de aprendizado estatístico, profundo e por reforço no processamento de linguagem natural (PLN), discutindo a evolução e aplicações dessas técnicas em tarefas como extração de informações e reconhecimento de entidades. Os autores exploram modelos estatísticos como HMM, MEMM e CRF, técnicas de word embedding como CBOW e Skip-Gram, e arquiteturas de redes neurais como CNNs e RNNs. O aprendizado por reforço é destacado como promissor para resolver problemas de otimização em PLN, especialmente em tarefas de longo prazo e baseadas em recompensas. O artigo sugere que a integração dessas abordagens pode aprimorar sistemas interativos, como agentes conversacionais, e melhorar a tomada de decisões em sistemas de diálogo e tradução automática.

Artigo 4

Pternea et al. [4] revisam estudos que exploram a sinergia entre aprendizado por reforço (RL) e modelos de linguagem de grande escala (LLMs), propondo uma nova taxonomia para classificar suas interações: RL4LLM, LLM4RL e RL+LLM. Essas categorias abordam como o RL pode aprimorar LLMs, como LLMs podem auxiliar no treinamento de agentes RL e como ambos podem ser integrados para resolver problemas complexos sem modificações diretas. O artigo enfatiza a importância dessa sinergia para desenvolver sistemas mais eficientes em aplicações interativas e dinâmicas, como assistentes virtuais e planejamento autônomo, servindo como referência para pesquisas que buscam combinar as vantagens de RL e LLMs.

Artigo 5

Chaudhari et al. [5] exploram o uso de aprendizado por reforço com feedback humano (RLHF) para alinhar modelos de linguagem de grande escala (LLMs) com preferências humanas, melhorando a qualidade das respostas e evitando conteúdos tóxicos ou enviesados. O artigo discute desafios e abordagens na aplicação de RLHF em LLMs, como o enriquecimento dos sinais de recompensa por meio do feedback humano, métodos de regularização e técnicas para superar recompensas esparsas. Embora o RLHF possa melhorar a relevância e segurança dos resultados, os autores destacam limitações como a sensibilidade a hiperparâmetros e a necessidade de calibragem cuidadosa. O RLHF é visto como crucial para desenvolver sistemas de IA seguros e úteis em áreas como assistentes virtuais, saúde e serviços financeiros.

Artigo 6

Fernandes et al. [6] revisam técnicas de aprendizado por reforço a partir de feedback humano (RLHF) e sua integração em sistemas de geração de linguagem natural, visando melhorar a qualidade dos modelos e evitar a geração de conteúdo tóxico ou impreciso. O artigo apresenta uma taxonomia das formas de feedback humano (numérico, rankings, linguagem natural) e discute dois métodos principais de utilização: treinamento direto com feedback humano e uso de modelos de preferência para guiar a geração de texto. O RLHF é destacado como meio de alinhar os modelos às expectativas humanas, com aplicações em sistemas de diálogo, tradução automática e geração de resumos. Os autores sugerem que o RLHF tem potencial para transformar a interação com sistemas de IA, tornando-os mais autossuficientes e alinhados às necessidades dos usuários.

Resumos dos Artigos em Métodos de Monitoria em Call Centers

Artigo 1

Grézl et al. [7] propõem um método automatizado de monitoramento de qualidade em call centers utilizando tecnologias de fala e processamento de linguagem natural. O sistema combina um módulo de resposta a perguntas (QA), que detecta palavras-chave para responder questões de qualidade, e uma classificação de entropia máxima (ME), que utiliza características derivadas do reconhecimento de fala para avaliar a probabilidade de uma chamada ser "ruim". A abordagem combinada dos sistemas QA e ME resultou em uma precisão de 53% para identificar os 20% superiores de chamadas e 44% para os 20% inferiores, superando os sistemas individuais.

Artigo 2

Baraka et al. [8] validam o modelo de sucesso de sistemas de informação de DeLone e McLean para avaliar o desempenho de call centers. O modelo considera seis dimensões: qualidade do sistema, qualidade da informação, qualidade do serviço, uso, satisfação do usuário e benefícios líquidos. Os autores identificam indicadores de desempenho para cada dimensão e propõem um índice de desempenho ponderado (W-CCPI) para avaliar globalmente os call centers. Aplicando a metodologia a call centers no Egito, concluíram que satisfação do usuário e benefícios líquidos são as dimensões mais importantes, enquanto qualidade do sistema e do serviço são prioridades nas dimensões de entrada. O W-CCPI auxiliou gestores a identificar áreas de melhoria e ajustar estratégias conforme as prioridades organizacionais.

Artigo 3

Park e Gates [9] propõem um método para medir automaticamente a satisfação do cliente em tempo real, analisando transcrições de chamadas geradas automaticamente. Utilizando modelos de aprendizado de máquina como SVMs, árvores de decisão, Naive Bayes e regressão logística, extraíram características prosódicas, linguísticas e comportamentais das chamadas para prever o grau de satisfação. O melhor modelo binário alcançou 89,42% de precisão, superando significativamente as linhas de base. O estudo demonstrou que é possível medir a satisfação do cliente em tempo real, permitindo que supervisores intervenham durante a chamada para melhorar a experiência do cliente.

Artigo 4

Karakus e Aydin [10] propõem um sistema distribuído de monitoramento de desempenho em call centers utilizando big data analytics. O sistema converte gravações de chamadas em

texto usando a API de fala do Google e analisa grandes volumes de dados com Hadoop MapReduce e algoritmos de similaridade textual como Cosine Similarity e n-gram. Métricas como tempo de resposta, solução de problemas e uso de saudações são avaliadas automaticamente. O sistema demonstrou escalabilidade em um cluster Hadoop de 10 nós, reduzindo custos operacionais e fornecendo análises de desempenho em tempo real. Além disso, oferece uma interface para que gerentes visualizem pontuações de desempenho, gerem relatórios e monitorem chamadas eficientemente.

Artigo 5

Baraka et al. [11] introduzem uma técnica de avaliação de desempenho em dois níveis para call centers, utilizando o modelo de sucesso de DeLone e McLean e o modelo do gap de realidade de Heeks. O primeiro calcula um índice de desempenho do call center (L-CCPI) baseado em seis dimensões, enquanto o segundo mede a discrepância entre valores projetados e reais em sete dimensões, calculando um índice de gap (CCGI). A ferramenta CCPET foi desenvolvida para permitir que gestores avaliem sistematicamente o desempenho, identificando áreas de melhoria e pontos fortes. Os resultados mostraram que a aplicação conjunta dos modelos fornece uma avaliação abrangente, com o modelo de Heeks sendo mais eficaz ao utilizar benchmarks da indústria para identificar discrepâncias específicas.

Análise dos Métodos Descritos e Conclusão

Os artigos analisados [7]-[11] empregam uma variedade de técnicas para avaliar a qualidade em call centers, desde abordagens baseadas em regras e classificações estatísticas até métodos de aprendizado de máquina e análise de big data. Enquanto alguns utilizam técnicas mais tradicionais, outros incorporam inteligência artificial e aprendizado de máquina para automatizar e melhorar a precisão das avaliações. Contudo, os métodos empregados geralmente são clássicos, não explorando plenamente as capacidades dos modelos de linguagem de grande escala (LLMs). A ausência de LLMs na literatura indica uma oportunidade para pesquisas futuras desenvolverem metodologias mais avançadas para avaliação de qualidade em call centers.

Referências

[1] V. Uc-Cetina et al., "Survey on reinforcement learning for language processing," *Artificial Intelligence Review*, vol. 55, no. 6, pp. 4537–4575, Aug. 2022.

[2] Q. Cai et al., "A Survey on Deep Reinforcement Learning for Data Processing and Analytics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4446–4465, May 2023.

[3] A. R. Sharma and P. Kaushik, "Literature survey of statistical, deep and reinforcement learning in natural language processing," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 350–354, 2017.

[4] M. Pternea et al., "The RL/LLM Taxonomy Tree: Reviewing Synergies Between Reinforcement Learning and Large Language Models," *Journal of Artificial Intelligence Research*, vol. 80, pp. 1–30, Aug. 2024.

[5] S. Chaudhari et al., "RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs," *arXiv preprint arXiv:2404.XXXXX*, Apr. 2024.

[6] P. Fernandes et al., "Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation," *arXiv preprint arXiv:2305.XXXXX*, May 2023.

[7] A. Grézl, M. Karafiát, and L. Burget, "Automated Quality Monitoring for Call Centers using Speech and NLP Technologies," *Proceedings of the 9th International Conference on Text, Speech and Dialogue*, pp. 202–209, 2006.

[8] H. A. Baraka et al., "Assessing call centers' success: A validation of the DeLone and McLean model for information systems," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 99–108, 2013.

[9] Y. Park and S. C. Gates, "Towards real-time measurement of customer satisfaction using automatically generated call transcripts," *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 1387–1396, 2009.

[10] B. Karakus and G. Aydin, "Call center performance evaluation using big data analytics," *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, 2016.

[11] H. Baraka et al., "Information systems performance evaluation, introducing a two-level technique: Case study call centers," *Egyptian Informatics Journal*, vol. 11, 2014.

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: 25 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

1. Entrega do planejamento da semana passada
    a. Aprofundamento nos surveys descritos.
    b. Aprofundamento na formalização matemática por trás de RLHF
    c. Mapeamento de métodos e suas qualidades

    Análise dos algoritmos

2. Definição do problema (e como RL é a solução)
    a. "Melhorar as avaliações de qualidade de atendimentos de call center"
        i. Se uma avaliação pode ser melhorada então necessariamente pode ser **medida.**
        ii. Como estou avaliando qualidade de atendimentos, então necessariamente pode ser **medido**
    b. Problemas:
        i. O que é uma avaliação?
        ii. Como medir a avaliação? Como melhorá-lá?
        iii. Como saber que nossa medição da avaliação é boa? Podemos melhorar essa medição?
        iv. O que é qualidade de atendimento?

    Formulação matemática do problema

    A formalização do problema deixa claro que é um problema de reforço.

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

A etapa de **entender o problema e as ferramentas** está feita. A próxima etapa é **planejamento,** isto é, encontrar um plano que mapeia a definição do problema e as ferramentas a um algoritmo que "melhora

as avaliações de qualidade de atendimentos de call center".

Para isso preciso ter conhecimento extenso de práticas que já são aplicadas a contextos similares à minha Formulação matemática do problema. Assim, para próxima semana planejo:

1. Buscar soluções a problemas similares ao meu: screening de artigos, blogs, vídeos e etc.
    a. Métodos
    b. Dados
    c. Frameworks
    d. Resultados
    e. Conclusões
2. Compreender as soluções de modo extenso
3. Mapear as melhores soluções

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

Os dados que serão eventualmente usados para treinamento dos modelos já estão em mãos, isto é, as transcrições que serão usadas.

ACEITE DA ENTREGA:

LEONARDO ALVES: Go!

Análise dos algoritmos

RLHF PPO

Descrição

O Aprendizado por Reforço com Feedback Humano (RLHF) é uma técnica para alinhar grandes modelos de linguagem (LLMs) com as preferências humanas. Ele consiste em duas etapas principais: o treinamento de um modelo de recompensa a partir de feedback humano e a otimização do LLM utilizando algoritmos de aprendizagem por reforço, como Proximal Policy Optimization (PPO). A otimização ocorre em torno de uma função de recompensa média, regularizada por uma divergência KL, que impede o desvio significativo do modelo de referência [1].

Vantagens

- **Desempenho aprimorado:** Combina SFT e RLHF para melhores resultados.
- **Aplicação prática:** Modelos como ChatGPT usam RLHF com sucesso.
- **Flexibilidade:** Adapta-se a diferentes domínios e tarefas.
- **Robustez para dados fora de distribuição:** O regularizador KL ajuda a manter a performance em contextos inesperados.

Desvantagens

- **Complexidade:** Necessita de dados de alta qualidade e uma configuração detalhada.
- **Sensibilidade aos dados:** Pode ser impactado pela qualidade dos dados de preferência.
- **Custo computacional:** Treinamento iterativo e exigente.

- **Risco de otimização excessiva:** Modelos de recompensa podem levar a comportamentos não intencionais.

RLOO

Descrição

O REINFORCE Leave One-Out (RLOO) é um algoritmo para RLHF online, mais eficiente que o PPO. Ele trata a sequência inteira de tokens gerados como uma única ação, reduzindo os requisitos de memória e convergindo mais rapidamente [2].

Vantagens

- **Uso reduzido de memória:** Necessita 50-70% menos memória de GPU.
- **Convergência rápida:** É de 2 a 3 vezes mais rápido que PPO.
- **Eficiência:** Facilita a implementação com menos erros de memória.

Desvantagens

- **Instabilidade numérica:** Pode ser afetado sob configurações de baixa precisão.
- **Feedback granular limitado:** Tratando a saída inteira como uma única ação.

DPO

Descrição

Direct Preference Optimization (DPO) elimina a necessidade de um modelo de recompensa. Em vez disso, otimiza diretamente o LLM com base em dados de preferência, simplificando o treinamento [3][4].

Vantagens

- **Simplicidade:** Dispensa o modelo de recompensa.

- **Menor overfitting:** Evita problemas associados a sinais de recompensa imperfeitos.
- **Desempenho forte:** Resultados sólidos em benchmarks acadêmicos.

Desvantagens

- **Sensibilidade às mudanças de distribuição:** Pode ter dificuldades fora da distribuição dos dados.
- **Resultados tendenciosos:** Riscos de políticas enviesadas.

KTO

Descrição

KTO (Otimização Kahneman-Tversky) utiliza a teoria da perspectiva para alinhar LLMs ao feedback humano. Ele opera com sinais binários de feedback, mais fáceis de coletar que dados de preferência [5].

Vantagens

- **Eficiência de dados:** Lida bem com desequilíbrios extremos.
- **Robustez:** Filtra dados ruidosos e reduz alucinações.
- **Escalabilidade:** Funciona bem em escalas de 1B a 30B parâmetros.

Desvantagens

- **Underfitting:** Pode ocorrer com sinais binários fracos.
- **Hiperparâmetros:** Sensível a configurações inadequadas.

BCO

Descrição

Binary Classifier Optimization (BCO) é um algoritmo que utiliza sinais binários, como "positivo" ou "negativo", para otimizar modelos de linguagem. Em vez de utilizar modelos de recompensa ou dados de preferência complexos, o BCO alinha o LLM diretamente com o feedback binário. Ele trata o logit do classificador binário como uma função de recompensa implícita, ajustando os pesos para maximizar o alinhamento com os sinais recebidos.

Vantagens

- **Simplicidade:** Utiliza feedback binário, que é mais fácil de coletar e menos propenso a ruído.
- **Eficiência:** Dispensa a necessidade de modelos de recompensa complexos.
- **Generalização:** Funciona bem em diferentes domínios, devido à simplicidade dos sinais de feedback.
- **Robustez:** Alcança desempenho sólido mesmo em distribuições de dados desbalanceadas.

Desvantagens

- **Limitações no feedback:** Feedback binário pode ser insuficiente para capturar preferências complexas.
- **Underfitting:** Pode apresentar desempenho inferior em contextos que requerem refinamento mais detalhado.
- **Dependência da distribuição:** Sensível a desequilíbrios extremos entre os dados positivos e negativos.

ORPO

Descrição

Odds Ratio Preference Optimization (ORPO) é um método eficiente que integra o

alinhamento de preferências diretamente na fase SFT, eliminando a necessidade de um modelo de referência separado [6].

Vantagens

- **Eficiência:** Reduz sobrecarga computacional.
- **Desempenho:** Resultados melhores que outros métodos em benchmarks.
- **Escalabilidade:** Funciona em modelos de 125M a 7B parâmetros.

Desvantagens

- **Dependência de dados:** Requer dados de alta qualidade.
- **Comparações limitadas:** Necessita de mais benchmarking.

Tabela comparativa

Algoritmo	Online/Offline	Feedback	Complexidade de Implementação	Modelo de Recompensa	Escalabilidade
RLHF PPO	Online	Binário e Preferência	Alta	Sim	Alta
RLOO	Online	Binário e Preferência	Moderada	Sim	Alta
DPO	Offline	Preferência	Baixa	Não	Moderada
KTO	Offline	Binário	Baixa	Não	Alta
ORPO	Offline	Preferência	Baixa	Não	Moderada
BCO	Offline	Binário	Baixa	Não	Alta

Tabela 1: Tabela comparativa dos algoritmos

Referências

[1] S. Chaudhari *et al.*, "RLHF Deciphered: A Critical Analysis of Reinforcement Learning from Human Feedback for LLMs," *arXiv*, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.08555

[2] A. Ahmadian *et al.*, "Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs," *arXiv*, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.14740

[3] S. Xu *et al.*, "Is DPO Superior to PPO for LLM Alignment? A Comprehensive Study," *arXiv*, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.10719

[4] R. Rafailov *et al.*, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," *arXiv*, May 2023. [Online]. Available: https://arxiv.org/abs/2305.18290

[5] K. Ethayarajh *et al.*, "KTO: Model Alignment as Prospect Theoretic Optimization," *arXiv*, Feb. 2024. [Online]. Available: https://arxiv.org/abs/2402.01306

[6] J. Hong *et al.*, "ORPO: Monolithic Preference Optimization without Reference Model," *arXiv*, Mar. 2024. [Online]. Available: https://arxiv.org/abs/2403.07691

Definição do problema

Problema: "Melhorar as avaliações de qualidade de atendimentos de call center"

1. O que é uma avaliação?

Uma avaliação é uma análise criteriosa da qualidade de um atendimento. Uma análise criteriosa se pauta em critérios. A análise em critérios se resume na resposta de uma lista de perguntas referentes a um mesmo contexto, todas dependentes de definições específicas. Por exemplo:

"Abertura": O atendente disse "bom dia"? Cumprimentou o cliente de alguma forma?
"Proatividade": O atendente se dispôs a resolver o problema apresentado? Ele utilizou de seus recursos e conhecimentos para auxiliar o cliente?

2. O que é qualidade de atendimento?

A qualidade do atendimento é definida pelo cumprimento de critérios específicos. Um atendimento de excelente qualidade segue um padrão de qualidade definido por critérios pré selecionados e elaborados. Se foi determinado que a qualidade de um atendimento se refere tão somente sobre se o atendente cumprimentou o cliente, então é um atendimento de boa qualidade.

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 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

1. Busca de soluções similares ao meu problema
    a. Busca por termos chaves em bibliotecas digitais
    b. 6 artigos pertinentes ao trabalho foram encontrados
    c. Compreensão e resumo de cada artigo

        Link: Resumo dos artigos

    d. Mapeamento de pontos a levar em consideração:
        i. **Self-Correction**: área que tem crescido bastante em NLP nas últimas semanas. Se baseia no princípio que verificar se um erro existe na resposta é mais fácil que evitar o erro. É um conjunto de técnicas que podem significativamente melhorar a performance das respostas de LLMs.

            Por que é pertinente: minha formulação do problema permite que o modelo avaliador forneça feedback da resposta do modelo atuador em tempo de inferência, potencialmente consertando falhas do modelo e melhorando a performance.

        ii. **RLAIF:** Uso de feedback por IA. A princípio é necessário um conjunto de dados com feedback humanos para aperfeiçoar a performance de LLMs, mas recentemente o uso de feedback por IA tem crescimento com o aumento generalizado de performance de modelos de língua.

            Por que é pertinente: reduz tempo de coleta de dados, aumenta escalabilidade e treinamento interativo sem modelo de recompensa.

        iii. **Alinhamento de preferência:** há a discussão filosófica sobre a capacidade que dados de preferência possuem de aprimorar a performance de modelos de LLMs. É necessário pensar exatamente qual comportamento queremos dos modelos a

fim de gerar os dados para treinamento.

Por que é pertinente: essa reflexão pode orientar a geração de dados sintéticos com RLAIF para treinamento dos modelos a fim de alcançar a melhor performance dos modelos.

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

1. Planejamento de uma solução para o problema incorporando tudo que foi estudado:
    a. Métodos de alinhamento de LLMs com reforço (DPO, ORPO, KTO, etc..)
    b. Aplicações desses métodos incorporando os pontos levantados no gate.
2. Busca e estudo de frameworks utilizados para treinamento e alinhamento de LLMs com Aprendizado por reforço

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo dos artigos

Artigo 1

Wu et al. [1], em "Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge", propõem uma abordagem para superar a limitação de modelos de linguagem que rapidamente saturam suas capacidades de julgamento. Enquanto métodos existentes focam em melhorar a geração de respostas, negligenciam o aprimoração da habilidade de julgar essas respostas, levando a problemas como overfitting no sinal de recompensa. Os autores introduzem um passo inovador de Meta-Recompensa, permitindo que o modelo julgue suas próprias avaliações e utilize esse feedback para melhorar suas habilidades de julgamento sem supervisão humana. A solução envolve o modelo atuando em três papéis: ator, juiz e meta-juiz, em um ciclo iterativo que visa melhorar tanto a capacidade do ator em seguir instruções quanto a habilidade do juiz em avaliar respostas de forma autônoma. Os resultados mostram melhorias significativas no desempenho do modelo, superando abordagens anteriores em benchmarks como AlpacaEval 2 e Arena-Hard, e abordando problemas como viés de comprimento nas respostas. Essa abordagem é relevante para problemas que envolvem ator e juiz, como na avaliação de conversas em call centers, alinhando-se ao objetivo de otimizar o desempenho através de treinamento iterativo e múltiplos níveis de feedback.

Artigo 2

Shinn et al. [2], no artigo "Reflexion: Language Agents with Verbal Reinforcement Learning", apresentam a framework Reflexion, que propõe melhorar a aprendizagem de agentes de linguagem grandes (LLMs) através de reforço verbal, evitando métodos tradicionais que requerem grandes quantidades de dados e recursos computacionais. A abordagem permite que agentes aprendam de maneira mais rápida e eficiente por meio de reflexões verbais sobre suas ações, proporcionando um feedback mais rico e interpretável. A solução consiste em três modelos: um ator que gera ações com base em observações, um avaliador que atribui notas às saídas do ator e um modelo de auto-reflexão que gera feedback verbal

sobre as ações e avaliações. Esse ciclo iterativo permite que o ator incorpore o feedback para melhorar em tentativas subsequentes, utilizando memória episódica e de longo prazo para aprimorar o desempenho em tarefas futuras. Os experimentos mostram que o Reflexion melhora o desempenho em tarefas complexas de tomada de decisão, programação e raciocínio, superando modelos como o GPT-4 em benchmarks como o HumanEval. A estrutura de auto-reflexão e o uso de feedback verbal gerado pelo próprio agente são relevantes para problemas que envolvem ator e juiz, como na avaliação de conversas em call centers, fornecendo insights para melhorar a precisão do juiz e otimizar os modelos através de múltiplas tentativas e feedback contínuo.

Artigo 3

Kumar et al. [3], em "Training Language Models to Self-Correct via Reinforcement Learning", abordam a limitação dos grandes modelos de linguagem (LLMs) em autocorrigir respostas erradas. Os autores propõem a técnica SCoRe (Self-Correction via Reinforcement Learning) para treinar LLMs a se autocorrigirem de forma eficaz sem o uso de dados externos ou oráculos, utilizando dados autogerados. A abordagem envolve treinamento de autocorreção via aprendizado por reforço em múltiplas tentativas, com dois estágios: uma inicialização que ensina o modelo a corrigir suas respostas de segunda tentativa sem mudar drasticamente a primeira, e um treinamento por reforço que otimiza as correções, incentivando o modelo a melhorar suas respostas com base em tentativas anteriores. Os resultados mostram ganhos significativos de autocorreção em problemas de raciocínio matemático e geração de código, superando métodos de ajuste fino supervisionado. A estratégia de incentivar correções positivas sem supervisão externa é relevante para sistemas onde a automação de avaliações é crítica, como na avaliação de conversas em call centers, permitindo que o modelo juiz aprenda com seus próprios erros e melhore a precisão de forma autônoma.

Artigo 4

Kamoi et al. [4], no survey "A Critical Survey of Self-Correction of LLMs", investigam quando modelos de linguagem grandes (LLMs) conseguem se autocorrigir, categorizando diferentes abordagens utilizadas e analisando as condições necessárias para o sucesso da

autocorreção. O estudo identifica que a autocorreção é eficaz apenas em tarefas altamente especializadas ou com feedback externo confiável, e que muitos estudos anteriores apresentam resultados inconsistentes devido a falhas experimentais. Os autores categorizam as soluções em frameworks de autocorreção e fontes de feedback intrínseco ou externo, destacando tendências emergentes e lacunas, como a dependência de ferramentas externas e a necessidade de melhor definição de questões de pesquisa. As conclusões apontam que métodos de autocorreção enfrentam desafios significativos em tarefas gerais e sugerem que a autocorreção sem informações externas é ineficaz em tarefas complexas. Isso é relevante para a avaliação de conversas em call centers, indicando que o modelo juiz poderia necessitar de dados externos ou supervisão adicional para refinar suas avaliações e melhorar a precisão.

Artigo 5

Lee et al. [5], no survey "RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback", comparam duas abordagens de aprendizado por reforço aplicadas ao ajuste fino de grandes modelos de linguagem (LLMs): o aprendizado por reforço com feedback humano (RLHF) e com feedback gerado por IA (RLAIF). Os autores exploram a viabilidade do uso de feedback gerado por LLMs como alternativa mais escalável e eficiente em termos de custo em comparação ao feedback humano. O estudo mostra que o RLAIF alcança resultados comparáveis ao RLHF em tarefas como sumarização e geração de diálogo, reduzindo custos e tempo de coleta de dados. Além disso, apresentam o d-RLAIF (Direct Reinforcement Learning from AI Feedback), que supera o RLAIF tradicional ao evitar a necessidade de treinar um modelo de recompensa. A utilização de feedback gerado por IA para ajuste fino é relevante para problemas que envolvem a automação de feedback, como na avaliação de conversas em call centers, oferecendo uma solução escalável para treinar modelos sem depender de anotações humanas.

Artigo 6

Zhi-Xuan et al. [6], no survey "Beyond Preferences in AI Alignment", desafiam a abordagem predominante no alinhamento de IA que assume que as preferências humanas são uma representação adequada dos valores humanos. Os autores examinam as limitações desse

enfoque e propõem alternativas que consideram a complexidade dos valores humanos, sugerindo o alinhamento de sistemas de IA com normas sociais e papéis contextuais em vez de apenas preferências individuais ou agregadas. O estudo argumenta que o alinhamento baseado em preferências é insuficiente, pois trata as preferências humanas como estáticas e ignora a pluralidade e a natureza construída dos valores. Propõem o alinhamento com normas e padrões sociais, destacando a necessidade de modelos de decisão que reflitam melhor o comportamento humano. As conclusões são relevantes para a avaliação de conversas em call centers, indicando que o alinhamento dos sistemas de avaliação deve focar em normas e critérios de qualidade adequados ao papel social dos atendentes, em vez de meramente otimizar com base em preferências individuais, evitando desafios de agregação de preferências conflitantes entre diferentes stakeholders.

Referências:

[1]T. Wu, W. Yuan, O. Golovneva, J. Xu, Y. Tian, J. Jiao, J. Weston, and S. Sukhbaatar, "Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-a-Meta-Judge," arXiv, 28-Jul-2024. [Online]. Available: https://arxiv.org/abs/2407.19594

[2]N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," arXiv, 20-Mar-2023. [Online]. Available: https://arxiv.org/abs/2303.11366

[3]A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, L. M. Zhang, K. McKinney, D. Shrivastava, C. Paduraru, G. Tucker, D. Precup, F. Behbahani, and A. Faust, "Training Language Models to Self-Correct via Reinforcement Learning," arXiv, 4-Oct-2024. [Online]. Available: https://arxiv.org/abs/2409.12917

[4]R. Kamoi, Y. Zhang, N. Zhang, J. Han, and R. Zhang, "When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs," arXiv, 19-Aug-2024. [Online]. Available: https://arxiv.org/abs/2406.01297

[5]H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, and S. Prakash, "RLAIF vs. RLHF: Scaling Reinforcement Learning

from Human Feedback with AI Feedback," in Proc. 41st Int. Conf. Mach. Learn., Vienna, Austria, Jul. 2024, pp. 26874–26901. [Online]. Available: https://arxiv.org/abs/2309.00267

[6]T. Zhi-Xuan, M. Carroll, M. Franklin, and H. Ashton, "Beyond Preferences in AI Alignment," arXiv, 6-Nov-2024. [Online]. Available: https://arxiv.org/abs/2408.16984

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: 10 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers
PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

1. Pesquisa dos frameworks para tarefas de RLHF
    a. Frameworks mais utilizados
    b. Framework escolhido: **HuggingFace TRL**
        i. Ampla variedade de algoritmos
        ii. Constantemente atualizado
        iii. Fácil implementação
        iv. Fácil integração com métodos PEFT
        v. Framework mais similar ao PyTorch (que já estou bem acostumado)
        vi. Acesso a diversos modelos e datasets
            1. llama 3.2 1b

2. Implementação de algoritmos
    a. primeiros_testes.ipynb

3. Planejamento de uma solução
    a. Solução Planejamento

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

1. Preparo do dataset da empresa para treinamento com RLHF
    a. Com a solução AEC descrita em Planejamento de solução, atualmente temos um dataset de tentativa e erros.
    b. Com este dataset, irei prepará-lo em:
        i. Dataset de preferência
        ii. Dataset com feedback binário

2. Implementação da solução simples com este dataset e realizar um prova de conceito.
    a. Framework escolhido: HuggingFace TRL

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Principais Frameworks para RLHF

Hugging Face (TRL): A Hugging Face oferece o TRL, um framework otimizado para RLHF, projetado para escalabilidade e flexibilidade. Ele é focado em integrar modelos de linguagem pré-treinados, usando principalmente o algoritmo PPO (Proximal Policy Optimization), amplamente adotado em RLHF. O TRL permite o treinamento com modelos de até 200 bilhões de parâmetros, sendo uma das soluções mais robustas para ajuste fino de grandes LLMs. Um diferencial importante é sua integração com a biblioteca de Transformadores do Hugging Face, facilitando a customização e o treinamento tanto de LLMs online quanto offline

Link: https://huggingface.co/docs/trl/index

O **HALO**s (Human-Aware Loss Functions) é um framework de código aberto desenvolvido pela ContextualAI para treinar grandes modelos de linguagem (LLMs) com feedback humano offline. O objetivo é criar loss functions que alinhem os modelos com as preferências humanas em grande escala, utilizando técnicas como DPO, KTO e PPO. Uma de suas aplicações principais foi no desenvolvimento do conjunto de modelos “Archangel”, que é uma das maiores suítes de LLMs ajustados com feedback humano.

Link: https://github.com/ContextualAI/HALOs

OpenRLHF: É um framework de código aberto focado na eficiência e escalabilidade, projetado para treinar modelos de linguagem (LLMs) com mais de 70 bilhões de parâmetros. Ele utiliza Ray para distribuir diferentes componentes (ator, crítico, referência e recompensa) entre múltiplas GPUs, otimizando o uso da memória e o desempenho. OpenRLHF oferece integração direta com DeepSpeed e técnicas avançadas como paralelismo de tensor e atenção contínua, otimizando a geração de amostras e o treinamento de grandes modelos

Link: https://ar5iv.labs.arxiv.org/html/2405.11143v1

Encord RLHF: Encord RLHF é um framework que facilita o desenvolvimento de fluxos de trabalho de RLHF escaláveis, otimizando modelos de linguagem e visão-linguagem (VLMs). Ele oferece recursos colaborativos para moderação de conteúdo e rotulagem de dados, além de ferramentas de segurança robustas. Ideal para equipes que necessitam de workflows personalizados e altamente escaláveis para LLMs

Link: https://encord.com/blog/top-tools-rlhf/

Soluções para o problema

Solução Básica

A solução envolve treinar o modelo Ator usando feedback de um modelo Juiz para melhorar suas avaliações. O Ator gera respostas iniciais que são julgadas pelo Juiz. Com base nesses julgamentos, um dataset de preferências é construído, comparando a qualidade das respostas. O Ator é então treinado usando Direct Preference Optimization (DPO), ajustando suas respostas com base nas preferências indicadas pelo Juiz. Esse ciclo contínuo de feedback e treino aprimora as avaliações do Ator, tornando-o mais preciso ao longo do tempo.

Solução SCoRe

Incorporar a solução no seguinte artigo:
Training Language Models to Self-Correct via Reinforcement Learning.pdf

A técnica **SCoRe (Self-Correction via Reinforcement Learning)** ensina modelos de linguagem a corrigirem seus próprios erros durante a inferência, sem supervisão externa ou dados oraculares. Ela utiliza aprendizado por reforço (RL) multi-turns para ajustar as correções com base nos erros cometidos pelo próprio modelo. O treinamento ocorre em duas fases:

1. **Fase 1:** O modelo é treinado para gerar respostas iniciais que se assemelhem ao modelo base, mas é incentivado a corrigir suas tentativas subsequentes de forma eficaz.
2. **Fase 2:** O RL é aplicado com bonificações que incentivam correções que transformam respostas incorretas em corretas, prevenindo a tendência do modelo de fazer apenas pequenas edições sem melhorias significativas.

SCoRe melhora a capacidade de autocorreção dos modelos, obtendo resultados de estado-da-arte em benchmarks como MATH e HumanEval, com ganhos consideráveis de desempenho.

Solução AEC

A solução, chamada **Ator-Avaliador-Corretor (AEC)**, baseia-se na interação entre três modelos:

1. **Ator:** Realiza a primeira tentativa de avaliação ou resposta.
2. **Avaliador:** Avalia a qualidade da resposta fornecida pelo Ator.
3. **Corretor:** Se o Avaliador classificar a resposta como insatisfatória, o Corretor tenta ajustá-la para que seja aprovada pelo Avaliador. O processo de correção é limitado a um número máximo de tentativas.

No contexto de call centers, o **Ator** simula a avaliação de atendimento, enquanto o **Avaliador** mede a qualidade das respostas.

O modelo Ator corresponde a avaliar a qualidade de atendimentos de call centers.
O modelo Avaliador corresponde a medir as avaliações feitas.

As propostas inclui três principais componentes:

1. **RLAIF (Feedback por IA):** Utiliza Inteligência Artificial para proporcionar feedback contínuo e garantir um treinamento mais ágil e escalável.
2. **Self-correcting (Auto-Correção):** Durante a inferência, o modelo tentará corrigir automaticamente erros detectados.
3. **Feedback específico:** O feedback utilizado para o treinamento é adaptado às necessidades dos modelos, indo além de simples classificações binárias.

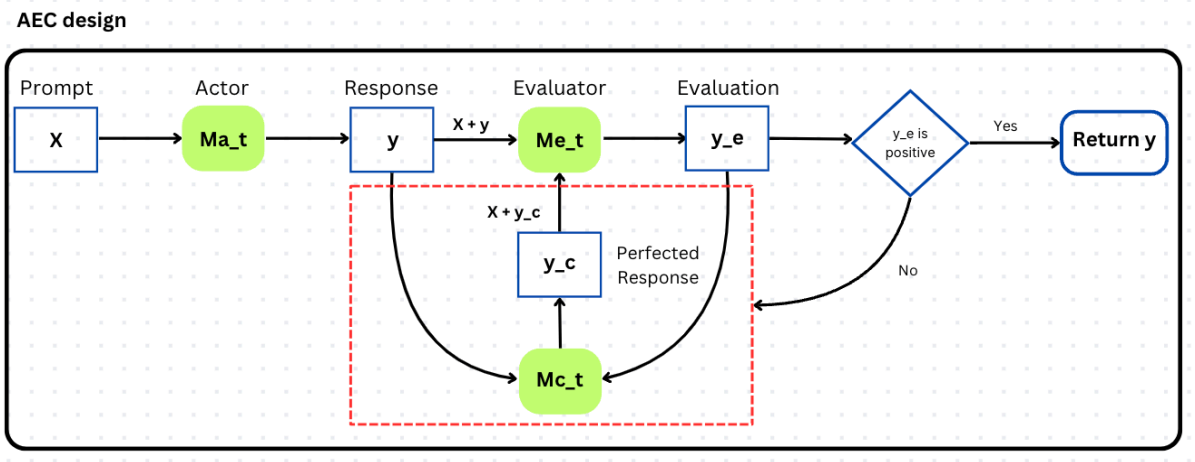


Figura 1: Esquema do modelo que será treinado e implementado

A solução foi desenvolvida tendo em mente 3 artigos:

1. ACritical Survey of Self-Correction of LLMs.pdf

2. Training Language Models to Self-Correct via Reinforcement Learning.pdf
3. LLMs and Judges.pdf

Essa abordagem pode ser classificada como **post-hoc, cross-model com external feedback**, conforme a estrutura teórica do primeiro artigo mencionado. Além disso, o treinamento possibilita a coleta de diferentes tipos de feedbacks por meio da técnica RLAIF.

Treinamento

O treinamento dos modelos será dividido em fases, que incluem:

1. **Coleta de Dados:**
   - O modelo Ator realizará uma série de avaliações baseadas em transcrições e critérios específicos de atendimento.
   - O Avaliador julgará as respostas do Ator, sendo seus julgamentos revisados por um modelo de juiz (meta-juiz).
   - Cada avaliação será classificada como 1 (aprovada) ou 0 (reprovada), permitindo o uso de métodos como KTO (Knowledge Transfer Optimization) e BCO (Binary Classifier Optimization).
2. **Treinamento:**
   - A coleta de dados resultará em dois conjuntos de dados, um para otimizar o modelo Ator e outro para o Avaliador.
   - Ambos os modelos serão ajustados com seus respectivos datasets, visando melhorar a qualidade geral das avaliações.

```
Início do Treinamento

1. Inicializar os modelos:
   - Ator
   - Avaliador
   - Corretor
   - Meta-Juiz (opcional)

2. Coleta de Dados:

   Para cada interação no conjunto de dados:
      - Ator recebe uma transcrição e instrução de avaliação.
      - Ator gera uma resposta (avaliação inicial).

      - Avaliador avalia a resposta do Ator.
      - Se a avaliação do Avaliador for insatisfatória:
         - Corretor tenta ajustar a resposta do Ator.
         - Avaliador reavalia a nova resposta.
         - Repetir até um número máximo de tentativas de correção.

      - Meta-Juiz pode ser usado para avaliar a resposta final.
      - Classificar a avaliação (0 para insatisfatório, 1 para satisfatório).

      - Registrar:
         - Respostas do Ator
         - Avaliações do Avaliador
         - Correções feitas pelo Corretor (se aplicável)
         - Avaliação final (1 ou 0)

3. Treinamento dos Modelos:

   - Dividir o dataset coletado:
     - Dataset para o modelo Ator
     - Dataset para o modelo Avaliador

   Para o Ator:
      - Treinar o modelo Ator usando o dataset de respostas e as avaliações finais (1 ou 0).

   Para o Avaliador:
      - Treinar o modelo Avaliador usando o dataset com as avaliações do Ator e a avaliação final
fornecida pelo Meta-Juiz (se disponível).

   - Repetir o processo de coleta de dados e treinamento até atingir a convergência desejada.

4. Fim do Treinamento
```

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 2024

Participantes da Entrega [matriculados em Residência em IA]:

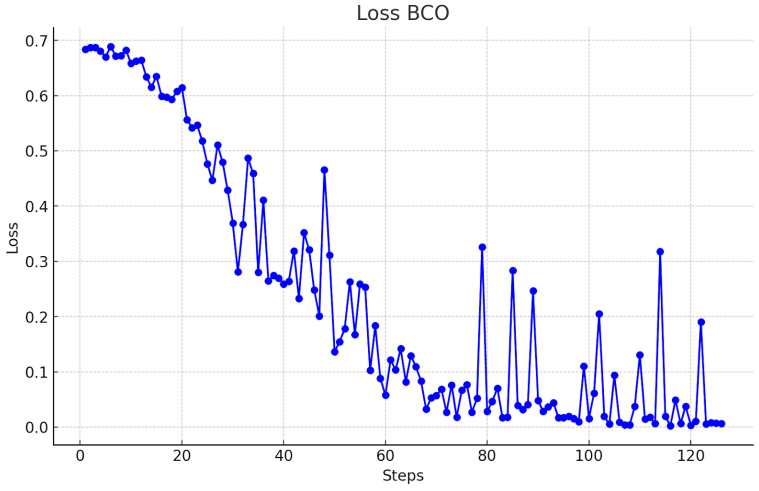
Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers
PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

1. Tratamento do dataset para binário
    a. Tuplas de (prompt,output,label)
    b. BCO (Binary Classification Optimization)
2. Treinamento de um modelo (POC) e observações iniciais:
    a. Poucos dados ~1000
    b. Uso de PEFT
    c. Modelo de 1B (llama 3.2)
    d. Treinamento ~ 1 hora
    e. Queda da loss

BCO_POC.ipynb



3. Novo método para a resolução do problema
    a. Aprendizado por Reforço com feedback Natural

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

1. Aumentar o tamanho do dataset
2. Revisar hiperparâmetros para evitar overfitting
3. Realizar um treinamento até o final
4. Avaliar a performance

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Primeiro teste

Objetivo:

O objetivo do primeiro teste foi avaliar a viabilidade do treinamento de um modelo de linguagem pequeno utilizando o método AEC (Ator-Avaliador-Corretor) e otimização binária (BCO) para análises automáticas de qualidade em atendimentos de call center.

Configuração Experimental:

- **Modelo Utilizado:** Llama 3.2 (1 bilhão de parâmetros) ajustado com PEFT (Parameter-Efficient Fine-Tuning).
- **Coleta de Dados:**
    - Dados coletados por meio da API implementada com a metodologia AEC.
    - Cada amostra consistiu em uma tupla contendo a transcrição, a avaliação gerada pelo modelo avaliador, e uma variável booleana indicando se a avaliação foi considerada satisfatória (1) ou insatisfatória (0).
- **Critério de Avaliação:**
    - Observação da redução da função de perda durante o treinamento com o método BCO.

Metodologia:

1. Coleta de exemplos de ações, observações e correções por meio da API do sistema AEC.
2. Construção de um dataset binário baseado nas avaliações geradas pelo modelo avaliador.
3. Treinamento do modelo ator utilizando Binary Classification Optimization (BCO) diretamente, sem um pré-treinamento supervisionado.

Resultados:

- Durante o treinamento, foi observada uma redução consistente da função de perda, indicando que o modelo estava aprendendo a otimizar o objetivo definido.
- No entanto, ao avaliar as respostas geradas pelo modelo, foi constatado que ele não estava funcionando conforme esperado. O modelo apresentou comportamentos como:
    - **Alucinações:** Geração de respostas sem sentido ou incoerentes com o contexto.
    - **Repetição Excessiva:** Palavras e frases sendo repetidas sem justificativa.
    - **Falta de Entendimento:** Dificuldade em interpretar transcrições longas com milhares de tokens.

Hipótese para os Resultados Observados:

- O uso direto de aprendizado por reforço (BCO) em um modelo sem pré-treinamento supervisionado se mostrou ineficaz, especialmente considerando a complexidade e o tamanho das transcrições utilizadas.
- A ausência de aprendizado supervisionado inicial dificultou a capacidade do modelo de compreender o contexto das transcrições, resultando em respostas inconsistentes.

Conclusão:

Os resultados indicam que, para treinar modelos pequenos no contexto de transcrições extensas e complexas, é necessário introduzir uma etapa de aprendizado supervisionado inicial. Este pré-treinamento pode fornecer uma base sólida para que o modelo compreenda o domínio dos dados antes de ser refinado por métodos de aprendizado por reforço. Além disso, dada a limitação no tamanho dos dados disponíveis, o uso de datasets menores, mas cuidadosamente preparados, será essencial para evitar problemas como overfitting e melhorar a generalização do modelo.

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: 30 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

Recapitulando: no último Stage treinei um modelo com RL com os dados da empresa (MaCall)

1. Problema: Houve overfitting
    a. Hipótese: Os estados (transcrições) são muito longos
2. Solução: Treinar um modelo que faz avaliação em pedaços de transcrições
3. Observação: Não tenho esses dados
4. Solução: Dados sintéticos
5. Ação: Com métodos de engenharia de prompt, e Gemini, preparei ~130.000 amostras de dados sintéticos ~ 10M tokens
6. Observação: Não preciso de um modelo grande. Posso treinar um modelo menor ainda.
    a. Bertimbau -> Embedding
    b. TeenyTinyLlama -> Autoregressivo
7. Ação: Treinei dois modelos
    a. Bertimbau: 88% de acurácia, sem análise
    SFT_MaCallity.ipynb
    b. TeenyTinyLlama: 70% de acurácia, com análise
    BERT_MaCallity.ipynb
8. Observação: Raciocínio do TeenyTinyLlama prejudicado

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

1. Transformar parte dos dados em dados de preferência/binários com engenharia de prompt
    a. Usar um prompt para degradar o raciocínio do dado sintético. Esse teste já foi feito
2. Usar métodos por reforço para treinar o modelo a realizar análises melhores

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

ACEITE DA ENTREGA:

LEONARDO ALVES: Em análise!

Preparo dos dados sintéticos

Detalhamento do Prompt: Estrutura "Top-Down"

O prompt foi projetado com uma abordagem "top-down", começando de uma visão macro e gradualmente especificando os elementos necessários para gerar exemplos consistentes e diversificados. Apesar de incluir informações detalhadas como a indústria, a empresa, o tipo de critério, entre outros, o objetivo final é criar exemplos sintéticos que resultem em uma tupla simples: **(pergunta, transcrição, análise da transcrição, resposta da pergunta)**. Essa estrutura garante que os exemplos gerados sejam aplicáveis a uma ampla gama de indústrias, sem perder consistência ou realismo.

Estrutura e Justificativa do Prompt

1. Indústria

Exemplo: *Indústria: Saúde*

Motivo no Prompt:

- A indústria contextualiza o atendimento e direciona os critérios de avaliação.
- Ao variar indústrias, como saúde, tecnologia ou varejo, é possível criar exemplos mais diversificados e representativos.

Impacto no Dataset:

- Garante que os exemplos sejam aplicáveis em diferentes contextos, ampliando a utilidade do dataset.

2. Empresa

Exemplo: *Empresa: Clínica Vida Saudável – Uma clínica que oferece consultas médicas e exames laboratoriais.*

Motivo no Prompt:

- A criação de empresas fictícias adiciona realismo ao contexto, permitindo que os critérios e transcrições sejam mais específicos.
- Empresas fictícias também ajudam a evitar problemas de privacidade.

Impacto no Dataset:

- Aumenta a personalização e detalhamento dos exemplos, especialmente para critérios complexos.

---

3. Critério Simples ou Complexo

Exemplo: *Critério simples ou complexo: Simples*

Motivo no Prompt:

- Define se o critério avaliado é geral (aplicável a qualquer contexto) ou específico (dependente das características da empresa).
- Exemplo de critério simples: *"O atendente cumprimentou o cliente educadamente?"*
- Exemplo de critério complexo: *"O atendente seguiu o protocolo de segurança do cliente para agendamento de exames?"*

Impacto no Dataset:

- Permite a geração de exemplos que variam em dificuldade e aplicabilidade, balanceando a diversidade do dataset.

---

4. Critério de Avaliação

Exemplo: *Critério de avaliação: O atendente cumprimentou o cliente educadamente.*

Motivo no Prompt:

- Define o foco da análise na transcrição.
- Critérios bem definidos ajudam a garantir que as perguntas e respostas sejam coerentes.

Impacto no Dataset:

- Cria um alinhamento direto entre as perguntas, respostas e análises, evitando inconsistências.

---

5. Pergunta

Exemplo: *Pergunta: O atendente cumprimentou o cliente educadamente?*

Motivo no Prompt:

- A pergunta objetiva e binária é o elemento central da tupla final, facilitando a avaliação automatizada.
- Perguntas curtas e claras são mais fáceis de processar por modelos.

Impacto no Dataset:

- Garante que cada exemplo tenha um ponto de avaliação bem definido, permitindo consistência no treinamento.

---

6. Resposta da Pergunta

Exemplo: *Resposta da pergunta: Sim*

Motivo no Prompt:

- Fornece o rótulo binário necessário para avaliação do critério.

Impacto no Dataset:

- Forma a base para o treinamento do modelo avaliador, garantindo que cada pergunta tenha um rótulo claro.

---

7. Contexto da Conversa

Exemplo: *Contexto da conversa: O cliente ligou para agendar um exame de sangue.*

Motivo no Prompt:

- Dá suporte à geração de transcrições realistas, conectando o atendimento a cenários plausíveis.

Impacto no Dataset:

- Melhora a naturalidade das transcrições e aumenta a capacidade do modelo de generalizar para contextos reais.

---

8. Porção da Conversa e Contexto

Exemplo: *Porção da conversa: No início da ligação, o cliente explica o motivo da chamada.*

Motivo no Prompt:

- Segmenta a transcrição em partes específicas (início, meio ou fim), simulando interações reais.

Impacto no Dataset:

- Oferece exemplos variados e ajuda o modelo a entender diferentes momentos de uma interação.

---

9. Falhas na Transcrição

Exemplo: *Falhas na transcrição: Sim*

Motivo no Prompt:

- Simula cenários comuns de erros em dados reais, como ruídos ou cortes.

Impacto no Dataset:

- Aumenta a robustez do modelo, preparando-o para lidar com transcrições imperfeitas.

---

10. Transcrição

Exemplo:

Atendente: Bom dia, como posso ajudá-lo?

Cliente: Eu gostaria de agendar um exame de sangue.

Motivo no Prompt:

- Fornece a base para análise e resposta da pergunta.
- Usar transcrições curtas ou com falhas aumenta a diversidade do dataset.

Impacto no Dataset:

- Garante que cada exemplo seja conciso, relevante e focado no critério avaliado.

---

11. Justificativa da Resposta

Exemplo: *Justificativa da resposta: O atendente cumprimentou o cliente no início da conversa, indicando educação.*

Motivo no Prompt:

- Explica a lógica por trás da resposta, ajudando a validar as análises geradas pelo modelo.

Impacto no Dataset:

- Melhora a qualidade dos exemplos, tornando-os mais úteis para o treinamento do modelo.

Benefícios da Estrutura "Top-Down"

- **Consistência:** Cada exemplo segue um formato padrão, reduzindo a chance de inconsistências.
- **Diversidade:** A segmentação e parametrização permitem gerar exemplos variados e representativos de múltiplos contextos.
- **Generalização:** Exemplos aplicáveis a diferentes indústrias aumentam a utilidade do dataset em diversos cenários.
- **Relevância:** Foco em perguntas binárias objetivas e transcrições concisas, alinhados ao objetivo do treinamento.

Essa abordagem garante que, mesmo gerando muitos dados detalhados, o resultado final seja funcional, consistente e diretamente aplicável ao treinamento de modelos para avaliação de qualidade em call centers.

Prompt

```
'''-> Contexto: Preciso gerar um dataset sintético para avaliação automática de unidades de atendimento em empresas. O dataset precisa ser composto por duplas de prompt e uma resposta. Meu objetivo é poder treinar um modelo de língua para fornecer respostas a perguntas sobre transcrições de conversa em atendimentos. O objetivo é medir a qualidade do atendimento, ou seja, a conduta que o atendente possuiu na conversa.

-> Instrução: Você irá gerar um dado sintético. Antes de gerar o exemplo sintético, você irá definir alguns pontos como o contexto da conversa e o critério que está sendo avaliado. Siga o formato para entender os pontos.
```

-> Formato: Retorne apenas o que for pedido. Desenvolva o exemplo sintético no seguinte formato:

Indústria: [Defina a industria em que a empresa atua. Escolha qualquer industria]

Empresa: [invente uma empresa fictícia e descreva o que a empresa faz]

Critério simples ou complexo: [Defina se o critério será algo simples, ou mais complexo que depende dos objetivos da empresa. Se o critério for complexo, ele se encaixa apenas no contexto da empresa que você definiu e não poderá ser usado em qualquer empresa. Se o critério for simples, então ele é um critério geral que pode ser usado para mais contextos. Responda apenas "simples" ou "complexo"]

Critério de avaliação: [Defina o critério que será avaliado na transcrição. Deve ser simples ou complexo como você definiu. Deve ser sobre o atendimento, ou seja, a conduta que o atendente possuiu na conversa]

Pergunta: [Crie uma pergunta bem objetiva, curta, e que seja de sim ou não. Essa pergunta quando respondida, oferece a resposta sobre se o critério foi cumprido]

Resposta da pergunta: [Aqui você deve definir se a resposta da pergunta que você construiu é sim ou não.]

Contexto da conversa: [Descreva o motivo do cliente ter ligado para a unidade de atendimento da empresa em questão. Crie um motivo convincente e plausível de ocorrer]

Porção da conversa e contexto: [Desenvolva onde a transcrição que você irá criar
se encaixa na conversa. Ela pode ser no início, no meio, ou no fim. Forneça algum contexto do desenvolvimento da conversa até agora sobre o contexto da conversa]

Falhas na transcrição: [É possível que haja erros na transcrição como palavras repetidas, palavras similares sendo trocadas, erros gramaticais e cortes na conversa. Defina com “sim” ou “não” se a transcrição possui falhas ou não]

Transcrição: [Deve usar os marcadores 'Atendente:' e 'Cliente:' para separar as falas. Você está criando um PEDAÇO da transcrição, não ela inteira. Escreva algumas interações consecutivas. Se for determinado que há falhas na transcrição, então crie uma com falhas, caso contrário, faça ela ótima. Deve haver apenas falas na transcrição, isto é, não deve indicar coisas como barulhos ou silêncios.]

Justificativa da resposta: [Desenvolva numa única frase bem curta o motivo da resposta da pergunta ser "sim" ou "não". A justificativa deve iniciar com uma observação sobre a conversa, seguida de uma explicação de como esse fato justifica a resposta, por exemplo: "..o atendente realizou tal ato, e como esse ato implica tal coisa, então a resposta é..."]”

```
            -> Restrições: Para o exemplo, você deve utilizar os
seguintes dados

            Indústria: {industria}

            Critério simples ou complexo: {criterio_tipo}

            Resposta da pergunta: {resposta}

            Falhas na transcrição: {falha}

            -> Exemplos de interações reais:

            Você deve imitar o estilo da seguinte interação. Utilize XXX
quando uma informação foi sensível e identifica a pessoa (nome, CPF,
email, número de celular, etc...)

            {exemplos_interações}
        '''
```

Detalhes da Geração de Dados Sintéticos

A geração dos dados sintéticos foi realizada utilizando dois modelos de linguagem da API Gemini, cada um com volumes diferentes de exemplos gerados:

- Gemini-Pro: 80.000 exemplos.
- Gemini-Flash: 50.000 exemplos.

Paralelismo e Otimização:

Para maximizar a eficiência do processo, foi utilizado o conceito de threading, permitindo a geração paralela de múltiplos exemplos e reduzindo o tempo total de execução.

Estrutura do Dataset:

Os dados gerados foram organizados em um arquivo CSV com as seguintes colunas principais:

- prompt: Texto detalhado contendo instruções, contexto e parâmetros para a geração do dado.
- response: Resposta gerada pelo modelo com base no prompt.
- tokens_in: Quantidade de tokens utilizados no prompt.
- tokens_out: Quantidade de tokens gerados na resposta.

Treinamento dos Modelos BERT e Teeny Tiny Llama (TTL)

Com base no dataset gerado, foram realizados experimentos de treinamento utilizando dois modelos: **BERT (Bertimbau)** e **Tiny Tiny Llama (TTL)**. Cada modelo foi treinado com abordagens específicas, explorando suas capacidades para atender aos objetivos da tarefa.

Treinamento com BERT (Bertimbau)

O modelo **BERT (Bertimbau)** foi treinado para realizar **classificação binária** com foco em prever se a resposta à pergunta seria "sim" ou "não".

- **Pré-processamento dos dados:**
  - A entrada do modelo consistiu na concatenação da pergunta e da transcrição.
  - Exemplo de entrada:
    "Pergunta: O atendente cumprimentou o cliente educadamente? Transcrição: Atendente: Bom dia! Como posso ajudar?"
- **Tarefa de classificação:**

- O modelo foi treinado para mapear a entrada concatenada para uma saída binária (0 para "não" e 1 para "sim")..

- **Resultados:**
  - **Acurácia obtida:** 88%.
  - **Limitações:** O modelo BERT não fornece uma análise descritiva das transcrições, restringindo-se à classificação binária.

Treinamento com Teeny Tiny Llama (TTL)

O **Teeny Tiny Llama (TTL)** foi treinado como um modelo **autoregressivo** para gerar análises descritivas das transcrições e fornecer a resposta binária ("sim" ou "não").

- **Pré-processamento dos dados:**
  - A entrada do modelo consistiu na concatenação da pergunta e da transcrição.
  - Exemplo de entrada:
    "Pergunta: O atendente cumprimentou o cliente educadamente? Transcrição: Atendente: Bom dia! Como posso ajudar?"
- **Saída esperada:**
  - O modelo foi treinado para gerar uma análise descritiva seguida da resposta binária.
  - Exemplo de saída:
    "Análise da transcrição: O atendente cumprimentou o cliente de forma educada no início da conversa. Resposta: Sim."
- **Resultados:**
  - **Acurácia obtida:** 70%.
  - **Diferenciais:** O modelo TTL foi capaz de produzir análises descritivas detalhadas, explicando o motivo das respostas, o que o torna mais útil em cenários que exigem explicações detalhadas.

Métricas de Avaliação

Para ambos os modelos, a **acurácia** foi utilizada como métrica principal para avaliar a tarefa de classificação binária. O modelo TTL, além de responder "sim" ou "não", gerou análises descritivas que enriqueceram os resultados.

Resumo Comparativo

Modelo	Objetivo	Entrada	Saída	Acurácia	Diferenciais
BERT	Classificação binária	Pergunta + Transcrição	"Sim"/"Não"	88%	Alta performance, mas sem análises descritivas.
TTL	Análise descritiva + classificação	Pergunta + Transcrição	Análise descritiva + "Sim"/"Não"	70%	Gera análises descritivas detalhadas.

Tabela 1: Resumo comparativo entre métodos utilizados

Os resultados demonstram que cada modelo possui pontos fortes complementares. O **BERT** é eficiente para classificação direta, enquanto o **TTL** agrega maior valor em tarefas que exigem explicações detalhadas, mesmo com menor acurácia. Esses experimentos fornecem insights valiosos para aprimorar futuras abordagens.

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: 7 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

Recapitulando:

1. Treinei um modelo com os dados da empresa utilizando **RL** (método BCO)
2. O modelo deu overfitting (muito overfitting)
3. Posso criar um modelo treinado para analisar **pedaços** de atendimentos. Mas eu não tenho dados para isso
4. Solução: Dados sintéticos. Foram criados ~130.000 amostras de transcrições
    a. Estrutura do dado sintético:
        i. Pergunta de sim ou não
        ii. Transcrição
        iii. Análise
        iv. Resposta
    b. O objetivo é prever a análise e resposta com base na pergunta e na transcrição.
5. Foi treinado um modelo (TeenyTinyLlama) usando SFT. Obtive 68% de acurácia.

Para este Gate:

1. Um dataset para aprendizado por reforço foi criado.
    a. Para amostras dos ~130.000, um dado sintético "degradado" foi gerado.
    b. A instrução foi piorar a análise e errar a resposta propositalmente.
2. O modelo inicial treinado com SFT foi aprimorado com RL.
    a. Acurácia: 68% -> 79%

3. Outros modelos foram treinados

4. Resultados preliminares foram compilados

Resultados preliminares

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

1. Organizar os treinamentos e resultados dos modelos
2. Refazer teste dos modelos com mais dados
3. Elaborar melhorias

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Criação de Dados Sintéticos para Aprendizado por Reforço

Objetivo da Geração:

O objetivo principal desta etapa foi criar um dataset de preferência para treinamento de um modelo utilizando aprendizado por reforço. O dataset foi projetado para melhorar o desempenho do modelo inicial, que já havia sido treinado com aprendizado supervisionado fino (SFT). A ideia central era introduzir exemplos sintéticos degradados, ou seja, exemplos onde as análises e respostas fossem propositalmente incorretas. Isso permitiu que o modelo aprendesse a distinguir entre boas e más avaliações, refinando suas capacidades.

Estrutura do Dataset:

Cada exemplo sintético gerado possui a seguinte estrutura:

- **Pergunta:** Uma pergunta objetiva de sim ou não sobre um critério de avaliação da transcrição.
- **Transcrição:** Um pedaço da conversa entre cliente e atendente.
- **Justificativa da Resposta:** Uma justificativa explicando por que a resposta é "sim" ou "não".
- **Resposta da Pergunta:** A resposta binária ("sim" ou "não") para a pergunta baseada na transcrição.

Geração de Dados Degradados:

Para criar um dataset que suportasse aprendizado por reforço, foi necessário gerar pares de respostas onde uma fosse intencionalmente pior que a outra. Um prompt foi desenvolvido para guiar esse processo:

- **Campos Modificados:**
    - **Resposta da Pergunta:** Sempre alterada de "sim" para "não" ou vice-versa.

- **Justificativa da Resposta:** Adaptada para ser errada, mas convincente. Isso incluiu criar observações incorretas e raciocínios logicamente falhos para justificar a resposta errada.

- **Objetivo do Prompt:** Fornecer exemplos degradados que simulassem avaliações inadequadas, ajudando o modelo a aprender a reconhecer e corrigir esses padrões.

Exemplo de Prompt Utilizado:

```
-> Contexto: Estou preparando um dataset sintético utilizando uma IA de geração de texto. Até o momento eu tenho um conjunto de exemplos numa planilha. Entretando, preciso agora de um dataset de preferência, ou seja, preciso que para um mesmo exemplo eu tenha duas respostas sintéticas, mas uma que seja melhor que a outra.
Os dados sintéticos constroem uma conversa entre um atendente e um cliente de alguma empresa. Meu objetivo com este dataset é treinar uma IA que irá fornecer uma avaliação de algum critério sobre a transcrição.
A geração de dados sintético prepara um contexto aleatório a fim de tornar a transcrição mais real. O contexto envolve campos como Empresa, Critério que está sendo avaliado e etc. Os campos que serão usados para treinamento são: Pergunta, Transcrição, Justificativa da resposta e Resposta da pergunta

-> Instrução: Sua tarefa é adaptar o dado sintético a fim de torna-lo pior.

Os únicos campos que você deve alterar são:

"Resposta da pergunta" : Você deve sempre alterar a resposta de "sim" para "não", ou de "não" para "sim".
```

"Justificativa da resposta": Você deve modificar a justificativa a fim de que ela esteja errada. Ela deve ser convincente, porém errada. A justificativa deve iniciar com uma observação sobre a conversa, seguida de uma explicação de como esse fato justifica a resposta, por exemplo: "..o atendente realizou tal ato, e como esse ato implica tal coisa, então a resposta é...", entretando, tanto a observação quanto o raciocínio devem estar errados.

-> Formato: Retorne apenas os campos "Resposta da pergunta" e "Justificativa da resposta" modificados e nada mais.

Resultados do Treinamento com Reforço:

O modelo inicial, treinado com SFT e atingindo 68% de acurácia, foi aprimorado com os dados de reforço utilizando o método BCO. Os resultados foram os seguintes:

- **Acurácia Inicial (SFT):** 68%.
- **Acurácia Após Aprendizado por Reforço:** 79%.

Este aumento na acurácia demonstra a eficácia do dataset de preferência em ajudar o modelo a refinar suas análises e respostas.

Conclusão:

A criação de dados sintéticos degradados foi essencial para permitir que o modelo aprendesse a diferenciar avaliações adequadas de inadequadas. Este processo não apenas melhorou a acurácia, mas também reforçou a capacidade do modelo de lidar com análises mais complexas e diversos cenários de atendimento. Esses avanços formam a base para futuras otimizações e testes em escala maior.

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: 13 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

Recapitulando:

1. Um dataset para aprendizado por reforço foi criado.
    a. Para amostras dos ~130.000, um dado sintético "degradado" foi gerado.
    b. A instrução foi piorar a análise e errar a resposta propositalmente.

2. O modelo inicial treinado com SFT foi aprimorado com RL.
    a. Acurácia: 68% -> 79%

3. Outros modelos foram treinados

4. Resultados preliminares foram compilados

Para este Gate:

1. Scripts de treinamento (SFT e RL) e teste foram escritos e revistos.
2. Arquivos de configurações foram sistematizados
    a. Parâmetros
    b. Formato
3. Testes mais robustos e comparáveis foram feitos (fazendo)
4. finetuning

Resultados: Treinamentos e testes agora são sistemáticos e resultados são mais comparáveis

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

1. Dos ~130.000 dados sintéticos gerados, apenas 70.000 foram usados em treinamento. Para o próximo Stage, vou refazer os treinamentos dos melhores modelos, mas agora com todos os dados.

2. Revisar geração de dados sintéticos para RL e utilizar mais dados.
3. Refazer treinamento do model TTL_160 (SFT e RL)

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Sistematização de treinamento e avaliação

O foco foi organizar e sistematizar os processos de treinamento e teste, com o objetivo de melhorar a comparabilidade dos resultados e garantir a reprodutibilidade. A estrutura final dos arquivos e pastas foi definida da seguinte forma:

- **data:** Pasta onde todos os dados gerados e utilizados no treinamento são armazenados. Subpastas organizadas por tipo de dado, como “SFT_data” e “RL_data”.
- **configs:** Pasta contendo os arquivos de configuração de treinamento. Cada arquivo inclui:
    1. Diretório para os dados utilizados no treinamento.
    2. Diretório para salvar os resultados gerados pelo modelo.
    - Exemplo de arquivo: config_ttl.json.
- **src:** Scripts de treinamento e teste dos modelos.
    - Scripts principais organizados para tarefas específicas
    - Scripts documentados para facilitar o entendimento e reuso.
- **results:** Pasta onde os resultados dos treinamentos e testes são salvos.
    - Subpastas categorizadas por modelos e etapas de treinamento, como “TTL_SFT” e “TTL_RL”.

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: 27 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

Recapitulando:

1. Scripts de treinamento (SFT e RL) e teste foram escritos e revistos.
2. Arquivos de configurações foram sistematizados
    a. Parâmetros
    b. Formato
3. Testes mais robustos e comparáveis foram feitos (fazendo)
4. finetuning

Para este Gate:

1. Dados foram organizados
    a. Treinamento para SFT
    b. Treinamento para Aprendizado com Reforço
    c. Avaliação

2. Treinamentos foram realizados de forma sistemática
    a. Treinados no mesmo dataset
    b. Avaliados no mesmo dataset
    c. Hiperparâmetros similares

3. 14 modelos foram treinados e avaliados com os métodos mais estudados:
    a. Modelos:
        i. Decoder-Only: TeenyTinyLlama,Tucano,GPT2Small
        ii. Encoder-Decoder: PT-T5
        iii. Encoder-Only: BERTimbau

    b. Métodos:
        i. **S**upervised-**F**ine**T**uning (SFT)

ii. **D**irect **P**reference **O**ptimization (DPO)
iii. **B**inary **C**lassifier **O**ptimization (BCO)
iv. **O**dds **R**atio **P**reference **O**ptimization (ORPO)

4. Resultados das acurácias finais:
   a. Resultados do Treinamentos

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

Análises feitas pelas LLMs não foram avaliadas. Portanto o planejamento é utilizar uma LLM para realizar a avaliação da qualidade da análise

- Escolher API e LLM a ser usada (Gemini/GPT/Haiku etc..)
- Preparar prompt para comparar resposta desejada com resposta predita.
- Avaliar todos os modelos novamente sob essa nova métrica

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

ACEITE DA ENTREGA:

LEONARDO ANTÔNIO ALVES: Em análise!

Experimentos e resultados de acurácia

Resumo dos Resultados

Nesta etapa, foram realizados treinamentos e avaliações sistemáticas de 14 modelos utilizando os métodos mais estudados para aprendizado supervisionado e por reforço. O objetivo foi comparar o desempenho de diferentes arquiteturas e técnicas de otimização na tarefa de avaliação de qualidade de atendimentos em call centers.

Modelos Treinados

Os experimentos incluíram modelos de diferentes arquiteturas, abrangendo:

- **Decoder-Only:** TeenyTinyLlama, GPT2Small, Tucano.
- **Encoder-Decoder:** PT-T5.
- **Encoder-Only:** BERTimbau.

Cada modelo foi treinado no mesmo dataset para garantir comparabilidade, utilizando hiperparâmetros similares e avaliados com métricas consistentes. Isso permitiu explorar o impacto de diferentes arquiteturas e métodos na acurácia final.

Métodos de Treinamento Avaliados

Os métodos testados incluíram:

1. **Supervised Fine-Tuning (SFT):** Ajuste fino supervisionado utilizando os dados sintéticos gerados.
2. **Direct Preference Optimization (DPO):** Método de aprendizado por reforço baseado em preferências diretas.
3. **Binary Classifier Optimization (BCO):** Treinamento supervisionado binário para avaliação de respostas.
4. **Odds Ratio Preference Optimization (ORPO):** Método de aprendizado por reforço com otimização de odds ratio.

Resultados de Acurácia

Model/Method	SFT	DPO	BCO	ORPO
TTL	0.656	0.532	**0.716**	0.668
GPT2SMALL	0.7	0.684	0.456	**0.808**
Tucano	0.756	0.788	**0.864**	0.784
T5	**0.696**	-	-	-
Bert	**0.85**	-	-	-

Tabela 2: Acurácia dos modelos treinados com fine-tuning nos dados sintéticos

Os resultados indicam que o modelo **Tucano**, utilizando o método BCO, obteve a maior acurácia geral, atingindo 0.864. No entanto, outros modelos, como o **GPT2Small**, demonstraram excelente desempenho com ORPO, atingindo 0.808.

O **BERTimbau**, apesar de ser limitado ao método SFT, apresentou excelente desempenho com uma acurácia de 0.850, evidenciando o potencial de modelos Encoder-Only para classificação binária simples.

Observação Importantes

Embora as acurácias sejam indicadores claros de desempenho, análises descritivas feitas pelas LLMs ainda não foram avaliadas. Essa limitação será abordada na próxima etapa, onde o foco será na qualidade das análises geradas.

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: 4 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Lucca Emmanuel Pineli Simões

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

TEMA : NLP + RL para avaliação automática de qualidade atendimentos em call centers

PROBLEMA: Melhorar as avaliações de qualidade de atendimentos de call center

Recapitulando:

1. Dados foram organizados
2. Treinamentos foram realizados de forma sistemática
3. 14 modelos foram treinados e avaliados com os métodos mais estudados:
4. Resultados das acurácias finais
    a. Resultados do Treinamentos

Para este Gate:

1. Análises textuais geradas pelos LLMs foram avaliados
    a. Sentence Embeddings - ground truth e análise predita foram codificadas em embeddings e suas similaridades foram calculadas
    b. Prompt com LLM - Foi pedido a uma LLM comparar o ground truth com o predito e fornecer uma Nota de 0 a 10
2. Resultados
    a. Avaliação das análises - LLM + Sentence Embeddins

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

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Em análise!

Resultados da qualidade do texto

Resumo dos Resultados

Na última etapa do projeto, foi conduzida a avaliação qualitativa das análises textuais geradas pelos modelos treinados nas fases anteriores. Os experimentos focaram em comparar as respostas preditas com os valores esperados, utilizando tanto embeddings de sentença quanto uma avaliação qualitativa feita por uma LLM.

Principais Atividades Realizadas:

1. **Avaliação com Embeddings**: **Ground truth** e análises preditas foram transformados em embeddings de sentenças. As similaridades entre os embeddings foram calculadas, oferecendo uma métrica quantitativa para avaliar a proximidade semântica.

Model/Method	SFT	DPO	BCO	ORPO
TTL	0.70	0.61	0.67	**0.70**
GPT2SMALL	0.74	0.67	0.69	**0.75**
Tucano	0.73	0.71	0.71	**0.74**

Tabela 3: Análise da qualidade das respostas utilizando embeddings

2. **Avaliação com LLM**:

   Foi utilizado um modelo LLM para comparar as respostas preditas e esperadas.O modelo atribuiu notas qualitativas entre 0 e 10, considerando a coerência e a precisão das análises.

Model/Method	SFT	DPO	BCO	ORPO
TTL	5.43	4.62	5.39	**5.56**
GPT2SMALL	6.69	5.21	6.11	**7.02**
Tucano	6.68	6.64	**6.93**	6.91

Tabela 4: Análise da qualidade das respostas utilizando LLM

Conclusão

A análise dos resultados indica que a abordagem de aprendizado por reforço aplicada a modelos de linguagem natural pode efetivamente melhorar a qualidade da monitoria automática de atendimentos em call centers. Em termos de acurácia, o modelo “Tucano” treinado com o método BCO alcançou o melhor desempenho (0,864), superando as demais combinações testadas. Esse resultado reflete a capacidade do modelo em responder de forma mais consistente aos critérios estabelecidos, traduzindo-se em maior precisão na detecção de conformidade com os padrões de atendimento.

Já na avaliação qualitativa das análises textuais, realizada por uma LLM que considerou coerência e precisão, o destaque foi para o modelo “GPT2Small” treinado com ORPO, que obteve uma nota média de 7,02 em uma escala de 0 a 10. Essa pontuação evidencia a habilidade do modelo em não apenas responder corretamente, mas também fornecer análises mais interpretáveis e alinhadas ao contexto.

Assim, enquanto o “Tucano” com BCO se sobressai em termos de acurácia, o “GPT2Small” com ORPO oferece melhor qualidade nas explicações textuais. Os resultados, portanto, apontam para soluções híbridas: a utilização de um modelo altamente preciso combinada a outro mais articulado em suas justificativas, abrindo caminho para aplicações mais eficazes e confiáveis na monitoria automática da qualidade de atendimentos.