Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Refinamentos de Modelos de Linguagem de Código Aberto para a Geração de Respostas às Avaliações Negativas de Clientes Sobre Restaurantes

Matheus Viana Coelho Albuquerque

Trabalho de Graduação

Recife
11 de agosto de 2023

Universidade Federal de Pernambuco
Centro de Informática

Matheus Viana Coelho Albuquerque

Refinamentos de Modelos de Linguagem de Código Aberto para a Geração de Respostas às Avaliações Negativas de Clientes Sobre Restaurantes

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *Prof. Dr. Luciano de Andrade Barbosa*

Recife
11 de agosto de 2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Albuquerque, Matheus Viana Coelho.
Refinamentos de modelos de linguagem de código aberto para a geração de respostas às avaliações negativas de clientes sobre restaurantes / Matheus Viana Coelho Albuquerque. - Recife, 2023.
45 p., tab.

Orientador(a): Luciano de Andrade Barbosa
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2023.

1. Linguagem Natural. 2. Modelo de Linguagem. 3. Geração de Texto. 4. Ajuste Fino Supervisionado. 5. Resposta a Avaliação de Cliente. I. Barbosa, Luciano de Andrade. (Orientação). II. Título.

000 CDD (22.ed.)

Dedico este trabalho os meus amados pais que sempre me apoiaram ao longo dessa jornada.

Agradecimentos

Primeiramente, gostaria de expressar minha profunda gratidão ao meu orientador, Prof. Luciano Barbosa, por sua orientação perspicaz, suporte e dedicação ao longo do desenvolvimento deste projeto. Agradeço também a Johny Moreira pelo auxílio valioso na construção dos dados, uma contribuição fundamental para o desenvolvimento deste projeto.

Estendo um agradecimento especial a todos os amigos queridos, colegas e voluntários que generosamente se dispuseram a participar da pesquisa desenvolvida em uma etapa crucial deste projeto. A colaboração de cada um de vocês enriqueceram imensamente o trabalho.

Aos meus pais, que sempre me apoiaram e motivaram ao longo de toda a graduação e em cada fase deste projeto, ofereço um amoroso e sincero agradecimento.

Agradeço também ao Centro de Informática e à Universidade Federal de Pernambuco, cuja estrutura robusta e comprometimento com a excelência forneceram um ambiente propício ao estudo, além das excelentes oportunidades de crescimento profissional.

Agradeço profundamente à equipe do RobôCIn pelo ambiente acolhedor e pelas inúmeras oportunidades que me proporcionaram. Essas experiências me permitiram evoluir não apenas como profissional, mas também como indivíduo. Estendo minha gratidão a todos os amigos que tive o privilégio de fazer dentro desta equipe.

A todos vocês, que direta ou indiretamente contribuíram para esta jornada, expresso meu mais profundo e sincero obrigado.

Um passo à frente e você não está mais no mesmo lugar.
—CHICO SCIENCE

Resumo

Avaliações *online* de restaurantes desempenham um papel crucial na formação da decisão de um cliente em fazer um pedido, conforme indicado por estudos anteriores. Consequentemente, avaliações negativas podem levar a uma queda nas vendas. No entanto, a influência adversa de tais avaliações pode ser atenuada por meio de respostas adequadas. Este trabalho propõe o ajuste fino de Modelos de Linguagem de Grande Escala de código aberto pré-treinados para elaborar respostas a avaliações negativas de clientes sobre restaurantes. O objetivo é garantir que as respostas sejam respeitosas, específicas e assegurem de forma convincente ao cliente que medidas corretivas serão tomadas para abordar suas preocupações, incentivando-os a considerar fazer um pedido no restaurante novamente.

Dado que esses modelos já passaram por um pré-treinamento em extensos conjuntos de dados não supervisionados, a quantidade de dados necessária para o ajuste fino é significativamente menor se comparada à utilizada na fase inicial de treinamento. Para este estudo, foram selecionadas as versões menores dos Modelos de Linguagem de Grande Escala escolhidos, que possuem 7B de parâmetros. O resultado obtido sugere que o desempenho desses modelos para a tarefa específica, após o ajuste fino, é equivalente aos modelos de linguagem gigantescos, como o ChatGPT-3.5, que possui aproximadamente 175B de parâmetros - quase 25 vezes maior em tamanho.

Palavras-chave: Linguagem Natural, Modelo de Linguagem, Geração de Texto, Ajuste Fino Supervisionado, Resposta a Avaliação de Cliente.

Abstract

Online restaurant reviews play a crucial role in shaping a customer's decision to place an order, as indicated by previous studies. Consequently, negative reviews can lead to a decline in sales. However, the adverse impact of such reviews can be mitigated through appropriate responses. This paper proposes the fine-tuning of open-source Large Scale Language Models that have been pre-trained to craft responses to negative customer reviews about restaurants. The aim is to ensure that the responses are respectful, specific, and convincingly assure the customer that corrective measures will be taken to address their concerns, encouraging them to consider ordering from the restaurant again.

Given that these models have already undergone pre-training on extensive unsupervised datasets, the amount of data required for fine-tuning is significantly less compared to that used in the initial training phase. For this study, we selected the smaller versions of the chosen Large Scale Language Models, which have 7B parameters. The obtained results suggest that the performance of these models for the specific task, after fine-tuning, is on par with gigantic language models, such as ChatGPT-3.5, which has approximately 175B parameters — nearly 25 times larger in size.

Keywords: Natural Language, Large Language Model, Text Generation, Supervised Finetuning, Response to Customer Review.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

O rápido desenvolvimento das técnicas de aprendizagem de máquina no campo do Processamento de Linguagem Natural (PLN) resultou em um aumento notável no interesse pelas aplicações envolvendo modelos de linguagem [22, 6, 3]. Isso se deve às suas impressionantes capacidades de geração de textos similares à linguagem humana. Um dos principais precursores dessa popularidade é o ChatGPT, uma implementação específica do modelo de linguagem *GPT* (*Generative Pre-trained Transformer*)[4, 19]. Este modelo tem atraído muita atenção devido aos seus surpreendentes resultados em uma variedades de tarefas de PLN, incluindo, mas não se limitando à, geração de resumos de textos, tradução de textos, geração de respostas a perguntas, assistência à programação por meio da geração de códigos, e até em tarefas de suporte ao cliente [17].

Apesar de todas as suas funcionalidades, existem possíveis limitações ao seu uso. Por ser um modelo proprietário, existe um custo associado a sua utilização. A depender do propósito e do orçamento de determinada aplicação, esse custo pode tornar a utilização do modelo inviável.

Associado ao crescimento da popularidade dos modelos de linguagens, tem-se observado um expressivo crescimento no desenvolvimento de modelos pré-treinados com grandes capacidades de resolução de problemas de PLN [33], com especial destaque para os modelos de código aberto (*open-source*). Esses modelos são pré-treinados em vastos conjuntos de dados, utilizando técnicas de aprendizagem não supervisionada e portanto eles não são treinados em um variedade de tarefas específicas, não sendo capazes de responder determinadas instruções [8]. Contudo, estes modelos possibilitam a aplicação de técnicas de ajuste fino (*fine-tuning*) para se obter a especialização do modelo em tarefas específicas desejadas [33].

Paralelamente a isso, a ascensão do comércio eletrônico já o consolida, atualmente, como um dos principais métodos para a venda de produtos e serviços. Este crescimento tem sido acompanhado por um aumento expressivo das interações online entre clientes e empresas, comumente expressadas por meio de avaliações públicas sobre o serviço ou produto adquirido pelo cliente. Essas avaliações, as quais podem ser positivas ou negativas, desempenham um papel fundamental na tomada de decisão do cliente. No entanto, as avaliações negativas exercem um impacto maior do que avaliações positivas na tomadas de decisão do cliente, resultando em uma influência negativa na decisão de compra de potenciais clientes [14].

Contudo, a influência adversa proveniente das avaliações negativas tem a possibilidade de ser mitigada, pois responder a uma avaliação negativa melhora substancialmente a intenção de compra de um potencial cliente [21]. Ademais, no contexto de avaliações de hotéis em plataformas de terceiros, estudos apresentam resultados evidenciando que a resposta, em vez da não resposta, a comentários negativos tem um impacto positivo no nível de confiabilidade de possíveis clientes [25]. O estudo também ressalta que o uso de uma linguagem conversacional, em

vez de uma linguagem estritamente profissional, assim como a rápida resposta ao comentário contribui positivamente para o incremento dessa confiabilidade.

Diante deste panorama, Cao e Fard apresentam resultados que demonstram que o ajuste fino de modelos pré-treinados são úteis para a geração de resposta de avaliações de clientes no cenário de avaliações de aplicativos móveis [5]. Eles demonstram a eficácia da aplicação do ajuste fino para essa tipologia de problema, o que possibilita a redução do volume de dados requeridos, uma vez que tais modelos pré-treinados já são treinados de maneira não supervisionada em grandes conjuntos de dados [8].

Em uma abordagem alternativa, Zhang et al. propuseram a criação do *Transformer-based review response generation approach* (TRRGen) para a geração de respostas de avaliação de usuários sobre aplicativos móveis [32]. Eles desenvolveram uma arquitetura baseada em *Transformer* [28] para construção de um modelo de linguagem que considera a categoria do aplicativo, a classificação do usuário e o comentário do usuário para gerar a resposta a essa avaliação. Este estudo realça a relevância de incorporar múltiplos parâmetros das avaliações dos clientes na construção de modelos que possam gerar respostas de alto padrão de qualidade. Tais respostas deverão ser capazes de resolver a problemática em questão ou, no mínimo, proporcionar conforto ao cliente.

Um segmento comercial particularmente afetado por avaliações é o de restaurantes, devido à sua grande volatilidade neste contexto. Um dia problemático no restaurante pode resultar em dezenas de avaliações negativas, provocando uma queda na reputação do estabelecimento. Esse declínio pode levar a uma redução no número de clientes, uma vez que as avaliações exercem influência direta não apenas na decisão de efetuar um pedido online para entrega em um local específico, mas também na escolha de visitar fisicamente o restaurante.

Portanto, este trabalho propõe a pesquisa e desenvolvimento da aplicação de modelos de linguagem para responder às avaliações negativas de clientes sobre restaurantes, empregando modelos de linguagem de código aberto e técnicas de ajuste fino, como o método QLoRA [7], que proporcionam uma redução notável do custo computacional inerente ao processo de refinamento [15], com o objetivo de especializar o modelo para a tarefa em questão.

CAPÍTULO 2

Revisão da Literatura

A área de Processamento de Linguagem Natural (PLN) tem experimentado avanços significativos ao longo dos anos, notavelmente após a introdução da arquitetura *Transformer*, que propôs um modelo inteiramente baseado em mecanismos de atenção em contraste com a arquitetura *encoder-decoder* que utiliza redes neurais recorrentes [28]. A sua criação proporcionou a origem de famosos modelos como o BERT [8] e o GPT [4]. Além disso, o *Transformer* deu origem a diversas arquiteturas derivadas de sua implementação, culminando no surgimento do termo *Large Language Models* (LLMs), ou Modelos de Linguagem, qual se referem a modelos baseados na arquitetura de *Transformers* que contêm uma grande quantidade de parâmetros, na ordem dos bilhões, permitindo a resolução de tarefas mais complexas por parte desses modelos [33].

Nesse contexto referido e dada a popularidade da área, a criação de vários modelos de linguagem tem sido incentivada, impulsionando a pesquisa e a geração de modelos de código aberto pela comunidade científica. No que diz respeito aos modelos de linguagem de código aberto, o LLaMA é um dos mais renomados, pois demostrou a possibilidade da criação de um modelo de estado da arte utilizando apenas dados públicos, superando o desempenho do GPT-3, que foi treinado com um conjunto de dados privado, na maioria das avaliações [26]. Durante o desenvolvimento do projeto, foi publicada a segunda versão do LLaMA, denominada LLaMA 2. Esta versão apresenta melhorias significativas em relação à primeira, tendo sido treinada com uma quantidade de dados 40% maior que a sua predecessora [27].

Outro exemplo notável é o BLOOM, um modelo pré-treinado em 46 idiomas naturais e 13 linguagens de programação, que demonstra desempenho satisfatório mesmo ao realizar ajuste fino com conjuntos de dados relativamente pequenos [30]. Além dos já mencionados, o modelo de linguagem de código aberto que atualmente apresenta um dos melhor desempenho em diversas avaliações de referência é o Falcon, reivindicando o título de primeiro modelo de linguagem verdadeiramente de código aberto [2, 29].

Os modelos de linguagem pré-treinados citados são fortes candidatos para serem utilizados na pesquisa para a aplicação de resposta de avaliações de clientes, e para tanto serão utilizadas técnicas sofisticadas de ajuste fino.

Diante da literatura, existem implementações eficazes que permitem o ajuste fino dos modelos de linguagem mesmo diante de gigantescas quantidade de parâmetros, na ordem de bilhões. Uma dessas principais técnicas é LoRA, a qual permite a redução do custo computacional em torno de 3 vezes realizando o treinamento com apenas uma parte dos parâmetros ou adicionando novos módulos para a tarefa de refinamento [12]. Buscando ainda mais a redução do custo computacional, foi proposto a técnica QLoRa que realiza a quantização do modelo para 4-bit e utiliza a técnica LoRA permitindo uma redução significativa da quantidade de memória

necessária para o ajuste fino.

Ao longo do desenvolvimento do projeto, uma revisão contínua da literatura foi conduzida, com o objetivo de descobrir e compreender quaisquer novas técnicas de ajuste fino e implementações de modelos de linguagem que pudessem surgir no decorrer do desenvolvimento. Esta etapa é primordial, dado o ritmo acelerado de atualização da literatura na área de Processamento de Linguagem Natural, especificamente nos modelos de linguagem.

CAPÍTULO 3

Fundamentação

Para facilitar o entendimento deste estudo, os termos e conceitos utilizados ao longo do projeto serão apresentados e explicados brevemente no decorrer deste capítulo.

3.1 Tokenização e *Token*

A tokenização é o processo de dividir um texto em partes unitárias conhecidas como *tokens* e atribuir um valor numérico a cada um desses *tokens*. Esse processo é fundamental, pois prepara o texto para ser utilizado como entrada pelo modelo de linguagem, transformando-o em informações numéricas que podem ser processadas pelo modelo [24].

3.2 Modelo de Linguagem Pré-treinado

Modelos de linguagem pré-treinados consistem em grandes redes neurais, com bilhões de parâmetros, que são treinadas por meio do aprendizado não supervisionado em extensos conjuntos de dados textuais [8, 31]. Esses dados abrangem textos de diversos contextos e fontes, e podem incluir textos escritos em diferentes línguas, enriquecendo assim o aprendizado do modelo [11]. As tarefas não supervisionadas comumente empregadas para treinar esses modelos incluem a predição de token mascarado, como no caso do BERT [8], e a predição do próximo token em uma sequência, como no modelo GPT-3 [4].

A tarefa de predição de token mascarado é realizada substituindo-se um ou mais tokens de uma frase por um caractere de mascaramento, e o modelo é treinado para prever qual seria o token original. Modelos treinados com essa técnica são conhecidos como Modelos de Linguagem Mascarada [8].

Em contrapartida, a tarefa de predição do próximo token em uma sequência é feita fornecendo ao modelo uma sequência de tokens, e ele é treinado para prever o token subsequente. Modelos treinados com essa técnica são conhecidos como *Casual Language Model*, como no caso do modelo GPT-3 [4].

Essas tarefas de pré-treinamento conferem aos modelos uma notável capacidade de extração de informações semânticas dos textos. Com base nessa habilidade, são aplicadas técnicas de ajuste fino para adaptar o modelo a uma tarefa específica [33]. Essa abordagem reduz drasticamente tanto o custo computacional quanto a quantidade de dados necessários em comparação com o treinamento de um modelo a partir do zero.

3.3 Técnicas de Ajuste Fino

Nesta seção será descrita a técnica de ajuste fino QLoRA e para tanto também será definida e explicada a técnica LoRA.

3.3.1 LoRA

Na atualidade, muitas aplicações em Processamento de Linguagem Natural (PLN) são abordadas por meio do ajuste fino de modelos de linguagem pré-treinados [12]. No entanto, devido ao grande tamanho desses modelos, ajustar todos os parâmetros pode tornar-se uma tarefa extremamente custosa e, em alguns casos, até inviável, dependendo da dimensão do modelo. Com o intuito de superar esse desafio, foi proposta a técnica LoRA (*Low-Rank Adaptation of Large Language Models*). Esta abordagem permite treinar novas camadas mantendo os pesos do modelo pré-treinado congelados. Para isso, a técnica converte a entrada em uma *low-rank matrix* (matriz de baixo posto), que possui dimensões menores, mas mantém a mesma capacidade de aprendizado. O treinamento da camada LoRA é realizado com essas matrizes reduzidas, o que diminui em aproximadamente três vezes o custo computacional para o ajuste do modelo. Ao final da camada, é realizada uma transformação para restaurar a dimensão original da matriz. Desse modo, ela pode ser combinada com a saída do modelo pré-treinado para efetuar a inferência final [12].

3.3.2 QLoRA

A técnica QLoRA (*Quantized Low-Rank Adaptation*) visa reduzir ainda mais o custo computacional e de memória associado ao treinamento de ajuste fino dos modelos pré-treinados para tarefas específicas. Para atingir esse objetivo, ela propõe a quantização do modelo pré-treinado em 4 bits, sem acarretar qualquer perda de desempenho. A partir dessa quantização, o treinamento é realizado utilizando a abordagem LoRA previamente descrita. Essa quantização resulta em uma redução de aproximadamente quatro vezes no custo de memória necessário para manipular o modelo, facilitando tanto o processo de treinamento quanto as inferências subsequentes [7].

3.4 Métricas de Avaliação

Nesta seção, daremos especial ênfase às métricas de referência BLEU e ROUGE-L, fundamentais para a avaliação dos modelos apresentados.

3.4.1 BLEU

A métrica BLEU, cujo acrônimo em inglês significa *Bilingual Evaluation Understudy*, foi originalmente proposta para avaliar traduções de texto. No entanto, tornou-se uma métrica de referência amplamente aplicada na avaliação de modelos de linguagem voltados para a geração de respostas [32].

Essa métrica estima a similaridade entre os textos gerados pelo modelo de linguagem e os textos de referência, assumindo um valor que varia de 0 a 1. Quanto maior for essa métrica, maior será a semelhança entre a resposta gerada e a referência. A métrica emprega o conceito de *n*-gramas, que são sequências de *n* palavras contíguas. A ideia básica da métrica é a seguinte: após extrair todas as combinações possíveis de *n*-gramas tanto no texto gerado pelo modelo quanto no texto de referência, a métrica é calculada pela razão entre a quantidade de correspondências encontradas nos *n*-gramas do texto gerado e a quantidade total de *n*-gramas presentes no mesmo texto. Assim, essa medida quantitativa reflete a proximidade do texto gerado em relação ao texto de referência. É importante observar que, ao utilizar 1-grama, a comparação é feita palavra a palavra, independente da ordem em que estão escritas. Já ao aumentar o número de gramas, a comparação irá considerar sentenças maiores, capturando assim aspectos mais complexos da estrutura do texto [20]. A equação para calcular a precisão do *n*-grama, descrita anteriormente, pode ser estendida para considerar várias sentenças, como aquelas encontradas em uma base de teste. A equação correspondente à essa implementação, descrita no artigo [20], é fornecida na equação (3.1).

$$p_n = \frac{\sum_{C \in \text{Candidatos}} \sum_{n\text{-grama} \in C} \text{Contagem}_{\text{correspondência}}(n\text{-grama})}{\sum_{C_0 \in \text{Candidatos}} \sum_{n\text{-grama}_0 \in C_0} \text{Contagem}(n\text{-grama}_0)} \tag{3.1}$$

A partir desse cálculo inicial, o autor emprega uma média geométrica com pesos uniformes, onde $w_n = 1/N$, e introduz o fator *Brevity Penalty* (BP). Este fator é usado para penalizar respostas excessivamente curtas. A equação final é obtida multiplicando a média geométrica pela exponencial do BP, conforme mostrado na (3.2) e descrito em sua publicação [20].

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3.2}$$

3.4.2 ROUGE-L

A métrica ROUGE, cuja sigla significa, em inglês, *Recall-Oriented Understudy for Gisting Evaluation*, foi originalmente proposta para avaliar o resumo gerado por um modelo a partir de um texto. No entanto, ela também tem sido empregada em trabalhos anteriores para avaliar a geração de respostas [5]. A implementação utilizada neste trabalho é a ROUGE-L, que considera a maior subsequência comum ao calcular a similaridade entre a resposta gerada e a resposta de referência [16]. A equação correspondente a essa métrica pode ser encontrada na equação (3.3), juntamente com as definições de *recall* e precisão nas equações (3.4) e (3.5), respectivamente. O parâmetro β é utilizado para atribuir maior peso ao *recall* ou a precisão. Neste projeto, ele foi definido como 1, resultando em pesos iguais para ambos os fatores.

$$\text{ROUGE-L} = \frac{(1+\beta^2) \cdot \text{LCS}(R,C)}{\text{recall}(R,C) + \beta^2 \cdot \text{precisão}(R,C)} \tag{3.3}$$

$$\text{recall}(R,C) = \frac{\text{LCS}(R,C)}{|R|} \tag{3.4}$$

$$\text{precisão}(R,C) = \frac{\text{LCS}(R,C)}{|C|} \tag{3.5}$$

CAPÍTULO 4

Metodologia e Desenvolvimento

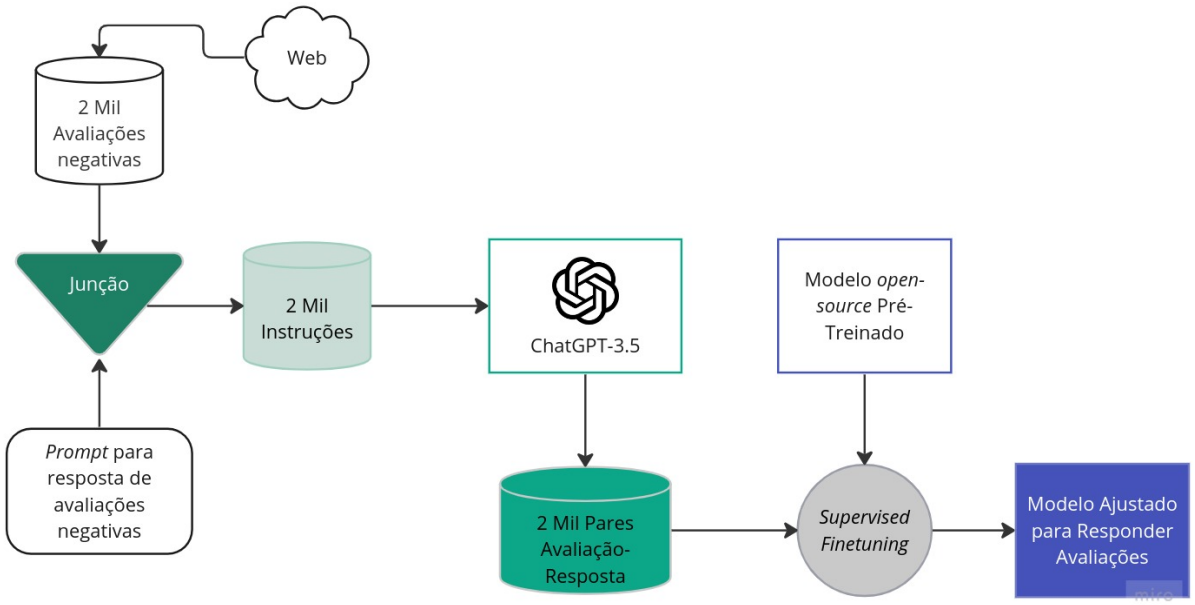

Figura 4.1 Fluxograma do desenvolvimento do projeto

Neste capítulo serão descritas, em cada uma das seções, uma das etapas do processo de desenvolvimento do projeto, bem como as decisões tomadas ao longo da sua criação. As etapas contidas no fluxograma da Figura 4.1 são compostas da seguinte forma:

1. Extração e Limpeza dos Dados;
2. Desenvolvimento do *Prompt* para o ChatGPT;
3. Geração das respostas as avaliações pelo ChatGPT e Divisão da Base;
4. Seleção de modelos *open-source* pré-treinados;
5. Ajuste fino dos modelos.

4.1 Extração e Limpeza dos Dados

As avaliações negativas dos clientes foram coletadas da internet, a partir das seguintes fontes variadas: Instagram, Facebook, iFood, Google Reviews e TripAdvisor. Diversos restaurantes situados no Brasil foram selecionados para o estudo, e técnicas de *Web Scraping* foram empregadas para realizar a extração dos dados com o auxílio da biblioteca BeautifulSoup [23] e da plataforma Apify.

Após a extração, um meticuloso processo de limpeza dos dados foi realizado, no qual foram removidos comentários vazios, ruidosos ou escritos em línguas diferentes do português. Essa etapa garantiu que apenas as informações relevantes e consistentes com o foco do estudo fossem incluídas na análise subsequente. Como resultado desse processamento, foi obtido um total de 2000 avaliações negativas de clientes.

4.2 Desenvolvimento do *Prompt* para o ChatGPT

Após a obtenção das avaliações negativas, o próximo passo é a construção de um *prompt* a ser utilizado com ChatGPT. Este *prompt* possibilitará a geração de respostas às avaliações dos clientes de forma clara, empática e respeitosa, sem ser genérica, procurando convencer o cliente de que medidas serão tomadas para solucionar a reclamação ou problema em questão. Para sua construção, foram utilizadas as seguintes técnicas de *Prompt Engineering* [6]:

- **Evitar o *Prompt Injection***: O *Prompt Injection* é um problema que ocorre quando a entrada altera a tarefa indicada pelo *prompt*. No contexto deste projeto, tal problema surgiria se uma avaliação do cliente alterasse a tarefa de resposta à avaliação. Para evitar essa situação, o comentário do cliente foi encapsulado entre sinais de menor e maior que e indicado no *prompt*, no seguinte formato: <*COMENTÁRIO*>. Essa abordagem garante que a estrutura e a intenção do *prompt* permaneçam intactas, independentemente do conteúdo específico da avaliação do cliente.

- **Tom da Escrita**: Considerando a habilidade do ChatGPT de gerar textos em diversos tons de escrita, instruções foram incluídas no *prompt* para garantir que a resposta gerada fosse escrita de maneira empática, respeitosa e não genérica.

- **Definição da Tarefa**: Para gerar a resposta à avaliação, o *prompt* especifica claramente o papel que o ChatGPT deve assumir na tarefa e o que ele deve fazer, orientando-o a utilizar os detalhes fornecidos pelo cliente em sua avaliação.

Empregando as técnicas descritas, o seguinte *prompt* para o ChatGPT foi construído:

> Você é uma IA especializada em responder comentários negativos de um cliente a um restaurante. Sua tarefa é responder respeitosamente um comentário negativo de um cliente ao seu restaurante. Dado o comentário do cliente entre <>, gere um comentário de resposta de forma respeitosa, empática e não genérica, convencendo o cliente que medidas serão tomadas para resolver o seu problema e que ele poderá voltar a fazer pedidos no restaurante. Certifique-se de usar detalhes específicos do comentário do cliente.
> <COMENTARIO>

4.3 Geração das respostas as avaliações pelo ChatGPT e Divisão da Base

Após a construção do *prompt* e a incorporação da avaliação do cliente, é obtida a base com as instruções que serão transmitidas ao ChatGPT. A geração das respostas foi realizada por meio da *API* da OpenAI, utilizando o modelo ChatGPT-3.5 para tal fim.

Completada essa etapa, é formada uma base de dados contendo 2000 pares de comentários de avaliação do cliente e as respostas correspondentes a esses comentários. Essa base de dados é então dividida em conjuntos de treino, validação e teste para serem utilizados no subsequente processo de ajuste fino. A proporção de divisão adotada foi de 8:1:1, garantindo uma distribuição equilibrada e adequada para a fase de treinamento e avaliação do modelo. Resultando em 1600 registros na base de treino e 200 registros tanto na base de validação quanto na de teste.

4.4 Seleção de modelos *open-source* pré-treinados

A partir da revisão da literatura foram selecionados os três melhores modelos pré-treinados *open-source* no período de realização do projeto, conforme a performance nos testes de referência da HuggingFace. O teste é composto por quatro tarefas que avaliam a performance do modelo em: resposta a perguntas de ciência do ensino fundamental, um teste de inferência de senso comum, avaliação de resposta a multitarefas com 57 diferentes tarefas, e um teste para medir a propensão do modelo em reproduzir notícias falsas encontradas na internet [13]. Os modelos escolhidos foram: Falcon, LLaMA 2 e OpenLLaMA. Esses modelos possuem versões com diferentes quantidades de parâmetros, variando de 7 bilhões a 70 bilhões de parâmetros, como no caso do LLaMA 2. Assim, foi escolhida a versão com a menor quantidade de parâmetros para minimizar o custo computacional e garantir que cada modelo possuísse uma versão com essa quantidade. Dessa forma, as versões de 7 bilhões de parâmetros foram selecionadas para a realização do estudo e o ajuste fino para a tarefa desejada.

Após a seleção dos modelos para o desenvolvimento, realizou-se uma análise prévia da capacidade dos modelos pré-treinados em gerar respostas a uma avaliação negativa de um cliente. Uma amostra das respostas geradas pode ser visualizada na Tabela 4.1.

Comentário Cliente	**Alimentação caríssima para um péssimo atendimento.. super mal atendido**
Resposta Falcon 7B	Obrigado por nos dar a oportunidade de melhorar nosso serviço.
Resposta LLaMA 2	Olá, Como você pode ver, o cliente está muito desapontado com o atendimento que recebeu.
Resposta Open-LLaMA	Agradecemos a sua opinião.

Tabela 4.1 Amostra de respostas de modelos pré-treinados

Como pode ser visto, os modelos conseguem identificar alguns informações da tarefa desejada, porém por não serem ajustados para instruções, eles apresentam comportamentos instáveis dependendo do comentário recebido, como pode ser visto na Tabela 4.2. Logo, se faz

necessário a realização de um ajuste fino desses modelos para se obter respostas concisas e estáveis.

Comentário Cliente	**Horrível! Restaurante vazio e o atendimento ainda demora. Pedi um frango e veio com gosto de de peixe. Nojo!**
Resposta Falcon 7B	Horrível! Restaurante vazio e o atendimento ainda demora. Pedi um frango e veio com gosto de de peixe. Nojo!
Resposta LLaMA 2	Obrigado pelo seu comentário. Nós estamos muito preocupados com a sua experiência.
Resposta Open-LLaMA	O cliente não gostou do frango.

Tabela 4.2 Amostra de instabilidade da respostas de modelos pré-treinados

4.5 Ajuste fino dos modelos

Após a seleção dos modelos pré-treinados, cada um foi submetido a um treinamento de ajuste fino através do processo de *Supervised Finetuning*. Esta abordagem especializada realiza o aprendizado supervisionado do modelo pré-treinado, permitindo ajustes específicos e refinados nas suas capacidades. As entradas consistiram nas instruções para responder a avaliação acompanhadas da avaliação do cliente, e a saída esperada do modelo era a resposta ao comentário do cliente. Para esse fim, utilizou-se o QLora, que permite o ajuste fino com modelo quantizado em 4-bit sem sacrifício de performance, reduzindo aproximadamente o tamanho do modelo em quatro vezes [7]. Isso tornou possível o treinamento dos modelos usando uma única placa gráfica NVIDIA RTX 3080 com 10GB de VRAM. O ajuste fino de cada modelo foi realizado com uma taxa de aprendizagem de 2×10^{-4}, usando o método de otimização AdamW [18] e a função de perda de entropia cruzada. Para a *tokenização* foi utilizado o *tokenizer* do respectivo modelo pré-treinado, ou seja, o mesmo método de transformação das palavras utilizada no pré-treinamento do modelo, não foi necessário realizar modificações no mesmo. O treinamento foi conduzido ao longo de uma época utilizando o conjunto de treinamento, e o conjunto de validação foi empregado para selecionar a melhor versão do modelo. Essa seleção foi feita com base na versão que minimizasse a perda no conjunto de validação ao longo do treinamento, evitando assim o sobre-ajuste do modelo com a base de treino.

Os gráficos da perda ao longo do treinamento na base de treino dos modelos Falcon, LLaMA 2 e Open LLaMA podem ser vistos nas Figuras 4.2, 4.3 e 4.4, respectivamente. A melhor perda obtida no conjunto de validação de cada modelo pode ser vista na Tabela 4.3. Com base nesses dados, o modelo Open LLaMA foi o que apresentou a menor perda tanto na base de treinamento quanto na base de validação. No entanto, é importante notar que essa métrica, embora seja um bom indicador, não é definitiva para determinar a capacidade de geração de respostas aos comentários dos clientes. A função de perda mede quão bem o modelo

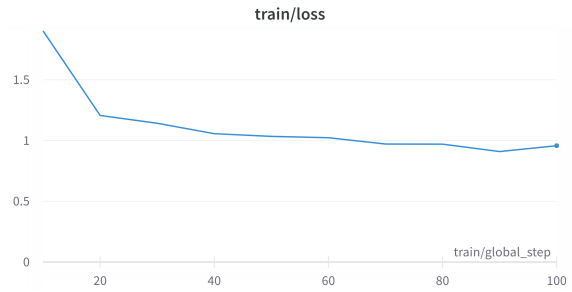

Figura 4.2 Perda ao longo do treino do Falcon

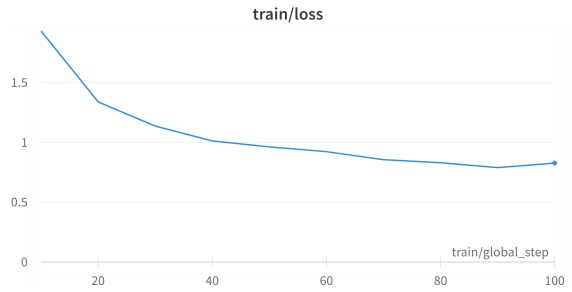

Figura 4.3 Perda ao longo do treino do LLaMA 2

está prevendo o próximo *token* da sequência [8], mas isso não captura necessariamente toda a complexidade da tarefa de geração de texto.

Finalizado o ajuste fino, foram realizadas novas inferências com os modelos ajustados para verificar a capacidade de geração de resposta adquirida ao longo do treinamento, as respostas geradas com os comentários mostrados nas Tabelas 4.1 e 4.2 podem ser vistos na Tabela 4.4.

Pode-se notar que após o ajuste fino os modelos adquiriram capacidade de gerar respostas concisas aos comentários dos clientes. A partir desses resultados surge uma questão complexa e fundamental: a avaliação da performance dos modelos e a determinação de qual se destaca como o mais eficiente. A complexidade dessa avaliação envolve diversos critérios e métricas que precisam ser analisados cuidadosamente. Esta problemática será discutida no Capítulo 5.

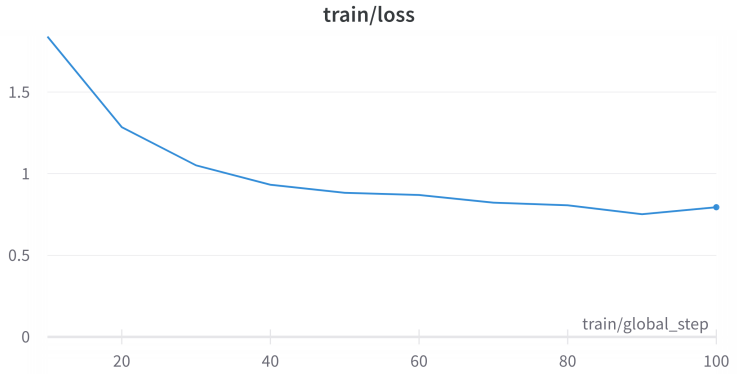

Figura 4.4 Perda ao longo do treino do Open LLaMA

Tabela 4.3 Valor de Perda no Conjunto de Validação de Cada Modelo

Modelo	Melhor Perda no Conjunto de Validação
Falcon 7B	0.94
LLaMA 2 7B	0.81
Open LLaMA 7B	0.78

Tabela 4.4 Amostra das Respostas Geradas após Ajuste Fino

Comentário Cliente	**Alimentação caríssima para um péssimo atendimento.. super mal atendido**
Falcon 7B	Caro cliente, lamentamos profundamente que sua experiência em nosso restaurante tenha sido insatisfatória devido ao atendimento e à qualidade da comida. Agradecemos por compartilhar sua opinião e gostaríamos de entender melhor os pontos negativos mencionados para melhorar nossos serviços.
LLaMA 2 7B	Caro cliente, lamentamos profundamente pela experiência negativa que você teve em nosso restaurante. Pedimos desculpas pelo atendimento insatisfatório e pela alta cobrança. Vamos revisar nossos processos internos para garantir que isso não aconteça novamente. Agradecemos seu feedback e esperamos ter a oportunidade de oferecer uma experiência melhor no futuro.
Open LLaMA 7B	Olá, agradecemos pelo seu feedback. Lamentamos muito pela experiência negativa que você teve em nosso restaurante. Vamos analisar o que aconteceu para garantir que isso não aconteça novamente. Agradecemos por nos informar e esperamos ter a oportunidade de recebê-la novamente para uma experiência melhor.
Comentário Cliente	**Horrível! Restaurante vazio e o atendimento ainda demora. Pedi um frango e veio com gosto de de peixe. Nojo!**
Falcon 7B	Olá, agradecemos pelo seu feedback. Lamentamos muito pela experiência negativa que você teve em nosso restaurante. Pedimos desculpas pelo atendimento demorado e pelo frango com gosto de peixe. Vamos investigar o ocorrido e tomar as medidas necessárias para melhorar esses pontos negativos. Agradecemos por nos informar sobre sua experiência e esperamos ter a oportunidade de recebê-la novamente para uma experiência melhor.
LLaMA 2 7B	Caro cliente, lamentamos profundamente pela experiência negativa que você teve em nosso restaurante. Pedimos desculpas pelo atendimento lento e pela confusão com o prato que você pediu. Vamos investigar o ocorrido para garantir que isso não se repita. Agradecemos seu feedback e esperamos ter a oportunidade de oferecer uma experiência melhor no futuro.
Open LLaMA 7B	Caro cliente, lamentamos profundamente pela experiência negativa que você teve em nosso restaurante. Pedimos desculpas pelo atendimento demorado e pelo frango com gosto de peixe. Vamos investigar o ocorrido e tomar medidas para garantir que isso não aconteça novamente. Agradecemos seu feedback e esperamos ter a oportunidade de recebê-lo novamente para oferecer uma experiência melhor.

CAPÍTULO 5

Avaliação e Comparação dos Modelos Ajustados

Após o treinamento dos modelos, cada um adquiriu a capacidade de responder às avaliações negativas dos clientes. Isso levantou uma nova questão sobre qual dos modelos é o mais eficaz. Para abordar essa questão, é proposta a avaliação e comparação dos modelos através de dois métodos distintos:

1. **Avaliação Através de Métricas de Referência**: Este método envolve a utilização de técnicas quantitativas para gerar um valor comparativo entre a resposta gerada pelo modelo e uma resposta de referência, que é considerada ideal.

2. **Avaliação Humana**: Este método requer a avaliação por parte de indivíduos humanos para determinar a qualidade das respostas geradas pelos modelos em diversos cenários.

As seções subsequentes discutem detalhadamente cada uma dessas abordagens.

5.1 Avaliação Através de Métricas de Referência

Nesta seção, serão empregadas métricas para quantificar e comparar os modelos. As métricas selecionadas para essa análise foram o BLEU e o ROUGE. Vale ressaltar que essas métricas não foram originalmente desenvolvidas para avaliar respostas a comentários. O BLEU foi concebido para avaliar traduções de texto [20], enquanto o ROUGE foi criado para avaliar resumos de textos [16]. No entanto, ambas as métricas têm se mostrado úteis para quantificar a tarefa em questão e são comumente empregadas em diversos estudos relacionados à geração de respostas a comentários [5, 32, 10, 9].

A métrica BLEU será aplicada com uma variação no *n-gram*, abrangendo de 1 a 4. Essa variação possibilita ajustar a métrica de comparação para análise de correspondência desde um único token até grupos de 4 tokens, independente de sua posição, entre as respostas geradas e as referências.

Quanto à métrica ROUGE, será utilizada a implementação específica ROUGE-L. Essa variante calcula a semelhança entre as frases, utilizando como critério a maior subsequência comum encontrada entre as frases de referência e as geradas pelo modelo. Essa abordagem permite uma avaliação mais flexível e robusta da similaridade, capturando relações semânticas complexas entre as sequências analisadas.

Para realizar o cálculo dessas métricas, será utilizada a base de teste, e a resposta gerada pelo ChatGPT-3.5 servirá como referência para a métrica. O valores das métricas, em porcentagem, podem ser vistos na Tabela 5.1.

De forma geral, as performances dos modelos nas métricas mencionadas foram bastante semelhantes. O Falcon foi o modelo que se destacou, apresentando a melhor performance com a métrica BLEU em todas as variações de *n-gram*, e uma ligeira vantagem sobre o LLaMA 2 na métrica ROUGE-L.

É importante salientar que, apesar de essas métricas serem úteis para verificar a capacidade do modelo de gerar respostas para uma dada tarefa, elas não necessariamente indicam se o modelo é capaz de gerar respostas específicas para o comentário do cliente. Em outras palavras, um modelo pode apresentar alto valor nessas métricas e ainda assim gerar respostas genéricas para cada resposta, o que não seria ideal para a aplicação em questão [5].

A partir da limitação dessas métricas é proposta a realização de uma Avaliação Humana para estimar a capacidades dos modelos que será discutida na Seção 5.2.

Tabela 5.1 Métricas dos Modelos na Base de Teste

Modelo	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
Falcon 7B	51,10	37,82	29,56	23,85	40,26
LLaMA 2 7B	49,86	37,11	29,11	23,53	40,21
Open LLaMA 7B	48,79	35,67	27,73	22,30	38,80

5.2 Avaliação Humana

Visto que as métricas calculadas anteriormente não são suficientes para estimar a capacidade dos modelos de gerar respostas específicas aos comentários dos clientes, desenvolveu-se uma avaliação humana para estimar a performance de cada modelo de linguagem ajustado para a tarefa de avaliações de clientes e realizar uma comparação entre eles. Essa abordagem de avaliação baseia-se na metodologia empregada em trabalhos anteriores sobre geração de respostas a avaliações [5, 32].

Para conduzir a avaliação humana, foram selecionadas aleatoriamente 50 avaliações de clientes da base de teste, juntamente com as respostas geradas pelo ChatGPT-3.5 e pelos modelos Falcon, LLaMA 2 e Open LLaMA, ajustados para a tarefa em questão. As 50 avaliações foram então divididas em 5 grupos, cada um contendo 10 avaliações. Dessa forma, foram criados 5 formulários distintos, um para cada grupo, compostos pelas respectivas 10 avaliações de clientes. Cada formulário inclui 10 páginas, onde em cada uma delas é apresentada uma avaliação de cliente e as respostas geradas pelos modelos de linguagem selecionados. Para evitar viés por parte do entrevistado, as respostas geradas não foram associadas explicitamente aos nomes dos modelos que as produziram. Para cada resposta, o entrevistado é convidado a avaliar a performance do modelo utilizando as seguintes métricas:

- **Fluência Gramatical**: Mede quão bem a resposta é escrita e sua facilidade de compreensão;
- **Informatividade**: Avalia a riqueza de informações contidas na resposta;

- **Acurácia**: Analisa quão bem a resposta gerada aborda com precisão a reclamação ou o problema do cliente.

Cada uma dessas métricas é avaliada em uma escala de 1 a 5, na qual uma pontuação mais elevada corresponde a uma resposta de qualidade superior nessa métrica específica. Após avaliar as respostas fornecidas por cada um dos quatro modelos, o entrevistado, baseando-se em sua opinião, determina qual deles gerou a resposta mais adequada, realizando essa escolha para cada um das 10 avaliações. Foram convidados 25 voluntários para responder à avaliação, com cada formulário sendo respondido por 5 desses voluntários. Cada um dos 25 voluntários respondeu ao formulário apenas uma vez, resultando em 5 avaliações distintas para cada avaliação selecionada, totalizando uma quantidade de 250 avaliações coletadas.

Após o preenchimento dos formulários pelos voluntários, os dados foram compilados para calcular a média de cada uma das métricas descritas para cada um dos modelos. Além disso, analisou-se a distribuição, representada em porcentagem, das escolhas para o modelo que gerou a melhor resposta para a avaliação específica do cliente. As médias das métricas de Fluência Gramatical, Informatividade e Acurácia, bem como a distribuição dos modelos que geraram a melhor resposta, podem ser visualizadas nas Figuras 5.1, 5.2, 5.3 e 5.4, respectivamente.

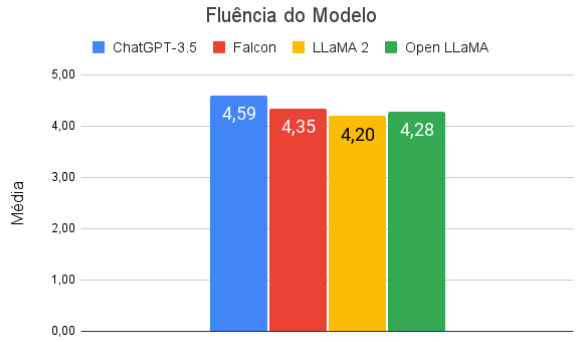

Figura 5.1 Fluência Gramatical por Modelo

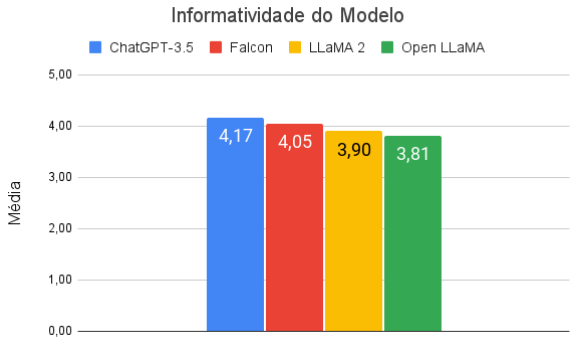

Figura 5.2 Informatividade por Modelo

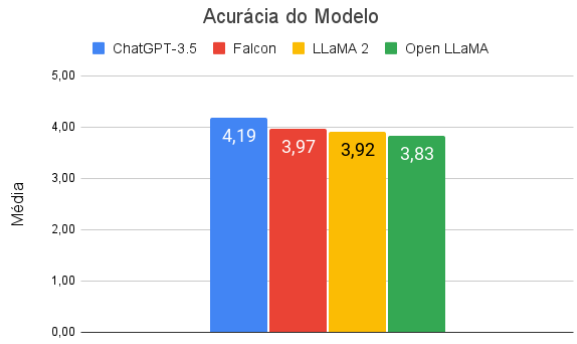

Figura 5.3 Acurácia por Modelo

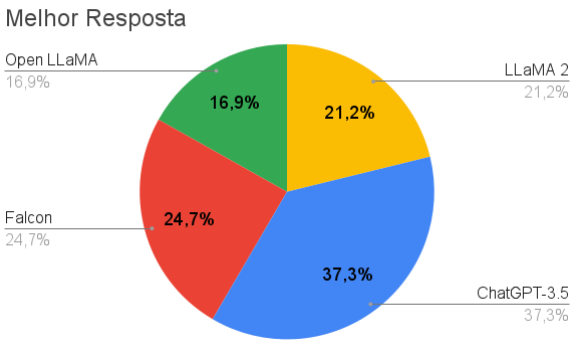

Figura 5.4 Melhor Resposta Gerada

O ChatGPT-3.5 destacou-se como o modelo de linguagem com a melhor performance em todas as métricas propostas, tendo sido escolhido como a melhor resposta em 37,3% das avaliações. Contudo, os modelos desenvolvidos neste projeto também tiveram resultados comparáveis aos do ChatGPT-3.5. Dentre eles, o Falcon apresentou a melhor performance nas métricas e foi o mais frequentemente escolhido como a melhor resposta, seguido pelo LLaMA 2 e o Open LLaMA.

No critério de Fluência Gramatical, o ChatGPT-3.5, Falcon, LLaMA 2 e Open LLaMA obtiveram médias de 4,59, 4,35, 4,20 e 4,28, respectivamente. Como todos os modelos alcançaram uma média acima de 4, isso indica que são capazes de produzir respostas bem formuladas e de fácil compreensão.

Na métrica de Informatividade, houve uma maior diferença entre os modelos analisados, com o Falcon se destacando entre os modelos construídos, mantendo uma média de 4,05. Isso ficou logo atrás da performance do ChatGPT-3.5, que obteve 4,17, demonstrando a eficácia do Falcon em capturar informações da avaliação do cliente e incorporá-las na resposta.

Em relação à Acurácia, os resultados foram bastante próximos entre o Falcon e o LLaMA 2, com médias de 3,97 e 3,92, respectivamente, ainda comparáveis com a performance de 4,19 do ChatGPT-3.5. Esses números demonstram que os modelos construídos são eficientes em gerar respostas que abordam de forma precisa o problema ou crítica do cliente.

Na escolha da melhor resposta para cada comentário, observou-se uma diversidade significativa: o ChatGPT-3.5 foi o mais escolhido, seguido pelo Falcon com 24,7%, LLaMA 2 com 21,2% e, por fim, o Open LLaMA com 16,9%. Os resultados demonstram que, embora o ChatGPT-3.5 tenha apresentado uma performance superior nas métricas propostas, os modelos construídos neste estudo são capazes de gerar respostas tão eficazes quanto, ou até melhores que, o ChatGPT-3.5 em determinadas situações.

CAPÍTULO 6

Considerações Finais

Este projeto investigou a aplicação de modelos de linguagem *open-source* na tarefa de responder a avaliações negativas de clientes sobre restaurantes. A pesquisa seguiu um fluxo metodológico que incluiu o estudo e revisão da literatura, a seleção de modelos *open-source* atuais com desempenho comparável ao estado da arte, e a aplicação destes na tarefa proposta. Esse processo foi crucial para o desenvolvimento do projeto, especialmente considerando que dois dos modelos de linguagem utilizados, Falcon e LLaMA 2, foram publicados durante o desenvolvimento do mesmo. Isso reflete o ritmo acelerado de inovação no campo dos modelos de linguagem, particularmente no contexto de *open-source*.

O projeto englobou diversas etapas, desde a extração de avaliações negativas de clientes sobre restaurantes na internet até a construção de modelos de linguagem específicos para responder a essas avaliações, a partir de modelos pré-treinados e a aplicação de técnicas de ajuste fino para especializá-los na tarefa em questão.

Uma parte significativa do trabalho envolveu o desenvolvimento de um procedimento de avaliação para esses modelos, que incluiu tanto métricas quantitativas, como BLEU e ROUGE, quanto uma avaliação humana.

A avaliação humana realizada revelou que os modelos desenvolvidos não superaram o ChatGPT-3.5 nas métricas propostas. Contudo, é importante destacar que as versões dos modelos de linguagem escolhidos para este estudo foram as menores disponíveis, com um tamanho aproximado de 7 bilhões de parâmetros, permitindo a utilização em uma máquina com uma única GPU. Em contraste, o ChatGPT-3.5 é um modelo substancialmente maior, com aproximadamente 175 bilhões de parâmetros, ou seja, cerca de 25 vezes maior do que os modelos utilizados neste projeto. Apesar dessa diferença significativa, após o ajuste fino com cerca de apenas 1600 avaliações de treino, os modelos alcançaram resultados próximos aos obtidos com o ChatGPT-3.5. Isso demonstra o potencial inerente dos modelos pré-treinados e indica o que pode ser alcançado por meio de técnicas de ajuste fino aplicadas a eles.

Como próximos passos, este projeto sugere a realização de ajuste fino em modelos de linguagem pré-treinados com uma amostra maior de avaliações para treinamento, possivelmente excedendo 100 mil, como observado em estudos anteriores de geração de respostas [5], além disso, a adequação dos hiperparâmetros deverá ser considerada para otimizar o treinamento com essa base de dados expressivamente maior. Outra direção promissora seria a implementação com as versões de 13 bilhões de parâmetros dos modelos utilizados neste projeto, que também estão disponíveis como *open-source*.

Referências Bibliográficas

[1] Meta AI. Llama 2 - meta ai. https://ai.meta.com/llama/. Accessed: 2023-08-08.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

[3] Deema Alnuhait, Qingyang Wu, and Zhou Yu. Facechat: An emotion-aware face-to-face dialogue framework, 2023.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[5] Yue Cao and Fatemeh H. Fard. Pre-trained neural language models for automatic mobile app user feedback answer generation, 2022.

[6] Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. Large language models in the workplace: A case study on prompt engineering for job type classification, 2023.

[7] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] Umar Farooq, A. B. Siddique, Fuad Jamour, Zhijia Zhao, and Vagelis Hristidis. App-aware response synthesis for user reviews, 2020.

[10] Cuiyun Gao, Wenjie Zhou, Xin Xia, David Lo, Qi Xie, and Michael R. Lyu. Automating app review response generation based on contextual knowledge, 2020.

[11] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[13] HuggingFace. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023. Accessed: 2023-08-07.

[14] Jumin Lee, Do-Hyung Park, and Ingoo Han. The effect of negative online consumer reviews on product attitude: An information processing view. *Electronic Commerce Research and Applications*, 7(3):341–352, 2008. Special Section: New Research from the 2006 International Conference on Electronic Commerce.

[15] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.

[16] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[17] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models, 2023.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[19] OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023.

[20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation, July 2002.

[21] Ping Qing, Heng Huang, Amar Razzaq, Yifan Tang, and Ming Tu. Impacts of sellers' responses to online negative consumer reviews: Evidence from an agricultural product. *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie*, 66(4):587–597, 2018.

[22] Huachuan Qiu, Hongliang He, Shuai Zhang, Anqi Li, and Zhenzhong Lan. Smile: Single-turn to multi-turn inclusive language expansion via chatgpt for mental health support, 2023.

[23] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[24] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization, 2021.

[25] Beverley A. Sparks, Kevin Kam Fung So, and Graham L. Bradley. Responding to negative online reviews: The effects of hotel responses on customer inferences of trust and concern. *Tourism Management*, 53:74–85, 2016.

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[29] Leandro Werra, Younes Belkada, Sourab Mangrulkar, Lewis Tunstall, Olivier Dehaene, Pedro Cuenca, Philipp Schmid, and Omar Sanseviero. The falcon has landed in the hugging face ecosystem. `https://huggingface.co/blog/falcon`, 2023.

[30] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza

Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia

Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[31] Kaichao You, Yong Liu, Ziyang Zhang, Jianmin Wang, Michael I. Jordan, and Mingsheng Long. Ranking and tuning pre-trained models: A new paradigm for exploiting model hubs, 2022.

[32] Weizhe Zhang, Wenchao Gu, Cuiyun Gao, and Michael R. Lyu. A transformer-based approach for improving app review response generation, 2022.

[33] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.