Universidade Federal de Pernambuco
Centro de Informática

Graduação em Engenharia da Computação

Serenpit Editor: Experiência de co-criação humano-computador de textos para internet utilizando GPT-3

Thiago Augusto dos Santos Martins

Trabalho de Graduação

Recife, Pernambuco
19 de maio de 2022

Universidade Federal de Pernambuco
Centro de Informática

Thiago Augusto dos Santos Martins

Serenpit Editor: Experiência de co-criação humano-computador de textos para internet utilizando GPT-3

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. Filipe C. A. Calegario*

Recife, Pernambuco
19 de maio de 2022

Dedico esse trabalho à minha mãe e ao meu pai. Edjane e Eliseu, sem o apoio de vocês eu não teria chegado aqui.

Agradecimentos

Primeiramente, gostaria de agradecer a minha família pelo suporte e valor dado a educação durante toda a minha vida.

Agradeço também aos meus amigos e amigas que nunca me deixaram na mão e sempre me apoiaram ao longo do caminho. Aos amigos e amigas da UFPE, obrigado pelas noites mal dormidas, trabalhos de última hora e risadas ao longo do caminho.

Agradeço aos professores e professoras que de tudo me ensinaram durante esses 5 anos de universidade. Especialmente, ao professor Dr. Filipe Calegário pelo suporte e direcionamento da criação deste trabalho. Obrigado também aos que disponibilizaram o seu tempo para participar das entrevistas deste trabalho.

Gostaria de agradecer também aos que contribuíram na minha jornada fora da universidade. Todos que me impactaram em hackathons, projetos de extensão, projetos de pesquisa, startups, voluntariado, e nas empresas que passei.

Arte pra mim não é produto de mercado. Podem me chamar de romântico.
Arte pra mim é missão, vocação e festa.
—ARIANO SUASSUNA

Resumo

Os humanos já utilizaram diferentes formas de comunicação ao longo do tempo, sejam elas textuais, auditivas, visuais ou audiovisuais. Um canal que cresce a cada ano é a produção textual para internet. Esse processo, normalmente envolve mais de um autor na co-criação dos textos. Entretanto, essa não é a única forma de co-criação existente para autores. Ultimamente, a interação de autores e IAs de propósito geral vêm se intensificando. Usuários escrevem *prompts* e as IAs dão respostas para eles. Existem modelos de inteligência artificial que conseguem produzir textos semelhantes aos de linguagem natural, como o GPT-3 da OpenAI. Nesse trabalho vamos entender sobre o estado da arte de geração automática de textos utilizando inteligência artificial e realizar um experimento de co-criação humano-IA para criação de textos para internet, utilizando o formato de blog como base.

Palavras-chave: GPT-3; Criatividade Computacional; Co-criação textual; Inteligência Artificial

Abstract

Humans have used different forms of communication over time, whether textual, audio, visual or audiovisual. A channel that grows every year is textual production for the internet. This process usually involves more than one author in the co-creation of texts. However, this is not the only form of co-creation that exists for authors. Lately, the interaction of authors and general-purpose AIs has been intensifying. Users write prompts and AIs give them answers. There are artificial intelligence models that can produce texts similar to natural language, such as OpenAI's GPT-3. In this work we will understand the state of the art of automatic generation of texts using artificial intelligence and perform an experiment of human-AI co-creation to create texts for the internet, using blog format as a basis.

Keywords: GPT-3; Computational Creativity; Textual co-creation; Artificial Intelligence

Sumário

CAPÍTULO 1

Introdução

Seja na escrita de um roteiro para vídeos no Youtube, na criação de um *blog post*, ou numa postagem de influencers em uma rede social, existe o envolvimento e co-criação entre humanos para revisar e garantir a qualidade daquele texto. Os principais apps utilizados na atualidade[Agr21], se beneficiam e utilizam de ferramentas audiovisuais e primariamente envolvem contar histórias. Essas histórias, são criadas a partir de conteúdos e roteiros, que podem ser organicamente gerados ou previamente definidos e pensados para melhor atingir suas audiências.

Atualmente, essa co-criação entre humanos não é a única forma de co-criação existente para esses autores. Uma técnica que vem se popularizando é o desenvolvimento de Inteligências Artificiais (IAs) de propósito geral, que permitem usuários escreverem *prompts* e as IAs terão a resposta para eles. Entre eles, temos o GPT-3 da OpenAI [Ope21], Megatron da Microsoft e Nvidia [AK21] e M6 do Alibaba [LMY+21]. Esses modelos vêm sendo utilizados em diferentes contextos, como: reescrita de textos, criação de assistentes virtuais, geração automática de códigos, entre outras.

Ferramentas autorais, como o Rytr [Rytb], Writesonic [Wri] e Jasper [jasa] utilizam IAs para melhorar o resultado de textos aplicados em diferentes casos de uso: texto para blogs, nome de marcas, geração de anúncios digitais, emails, entre outros. As chamadas *AI Copywriter*s e *AI Writer Assistants* melhoram o tempo de escrita em diferentes línguas e conseguem ser muito flexíveis no seu uso. Essas ferramentas têm diferentes *tiers* de uso, desde os níveis gratuitos a ferramentas de uso corporativo. Além disso, não é necessário entender tecnicamente como as IAs funcionam para poder usufruir de seus benefícios.

Essas ferramentas, tem seus limites de utilização, devido aos casos de uso que são implementados por elas. É necessário entender a necessidade dos usuários e implementar interações com as IAs de propósito geral que solucionem esses problemas. Mesmo que essas plataformas já estejam resolvendo alguns problemas, nem todos os usuários tem acesso a essas plataformas. Ainda existem limitações de linguagem e custo, que impedem o acesso de qualquer pessoa.

Outro ponto importante, é a possibilidade não só de geração de conteúdo, mas de análise e apoio a escrita como um todo. Sugestões de sintaxe e semântica, podem também estar nesse processo. As ações tomadas pelas IAs não necessitam ser somente de geração, mas também de edição, remoção e ampliação.

Considerando os temas acima, esse trabalho tem dois objetivos. O primeiro é entender o estado da arte para geração automática de texto e utilizar o modelo GPT-3 a criação de uma ferramenta de co-criação de textos para internet (primariamente *blog posts*) que possa agilizar o tempo de escrita desses conteúdos. Para criação dessa plataforma, fizemos uma análise de *Blue Ocean* [Atr22] sobre as atuais plataformas para co-criação textual, e definimos os principais

benefícios para a nossa audiência de escritores criativos.

Além disso, o trabalho busca avaliar os aspectos de usabilidade, suporte a criatividade e experiência do usuário ao utilizar essa plataforma. Uma pesquisa qualitativa foi feita com usuários da audiência proposta, para identificar como a proposta da plataforma performaria no Questionário de Experiência do Usuário(UEQ) [LHS08] e na Escala de usabilidade do sistema(SUS) [Lew18a].

1.1 Motivação

Steven Johnson, no seu livro *De onde vêm as boas ideias* [Joh11], fala sobre as características de ambientes que estimulam a criatividade, e levanta a pergunta de como reproduzí-los em diferentes momentos e locais do nosso cotidiano. O possível adjacente, redes líquidas e exaptação são alguns dos temas trazidos pelo livro. De acordo com o autor, o possível adjacente é uma característica de ambientes onde boas ideias surgem, onde os limites para criação e interação entre pessoas é muito maior; as redes líquidas são locais ondem informações se propagam com maior facilidade; e a exaptação é a capacidade de adaptação e utilização de ideias e produtos em outros meios, que não havia sido pensados inicialmente.

Mas, o tema abordado que mais se destacou para mim foi a Serendipidade. Em suas palavras,

> *A serendipidade completa uma intuição ou abre uma porta para o possível adjacente que não havíamos percebido. [...] A serendipidade requer colisões e descobertas improváveis, mas também algo em que ancorá-las.* [Joh11, p. 91]

A Serendipidade traz o papel do acaso, coincidências e das possibilidades de encontrar algo próximo que nos ajude a criar algo novo. Seja conectando diferentes áreas do conhecimento, ou nos expondo a diferentes contextos. Essa é uma das diferentes formas, em que a criatividade se apresenta e estimula a combinação de diferentes ideias.

A motivação desse trabalho surge da vontade em aumentar a serendipidade para criadores de conteúdo textual. Mesmo que esse conteúdo seja transpilado para as mais diversas formas de audiovisual, o seu formato textual, estruturado e de roteiro pode ajudar a aumentar esse possível adjacente.

Além disso, um dos maiores problemas que pessoas com atividades criativas encontram é o conhecido *Writers Block* [sta21] e *Creative Block* [McG19]. Nesse problema, as pessoas não conseguem produzir conteúdo, ou produzem um conteúdo que não as deixam felizes, seja porque não está ao mesmo nível de outras entregas já feitas ou é "mais do mesmo". Para desbloquear a criatividade, muitos vão dar uma volta, tomar um café, conversar com amigos, alongar, entre várias outras formas de espairecer.

A serendipidade catalisada pode ser uma forma de ajudar a desbloquear a criatividade dos escritores. Por isso, as ferramentas e benefícios escolhidos para a plataforma de co-criação desse trabalho tomam como problema principal o bloqueio criativo; e tem como seus principais benefícios a geração e modificação de conteúdo a partir de *inputs* do usuário.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver e avaliar uma ferramenta de suporte à criatividade na co-criação de textos para internet, utilizando o GPT-3.

São objetivos específicos do projeto:

- Realizar um estudo sobre o estado da arte para geração automática de textos utilizando inteligência artificial.
- Desenvolver uma plataforma de co-criação textual que use dos modelos estudados para auxiliar na escrita de textos para internet.
- Criar um experimento de teste da plataforma com pessoas que escrevam conteúdo para internet para avaliar usabilidade, experiência de uso e o suporte à criatividade da ferramenta.

Quanto a estrutura e organização dos capítulos desse trabalho:

- **No capítulo 2**: Falamos sobre o estado da arte dos modelos geradores de linguagem natural, plataformas de co-criação textual, e os desafios para desenvolvimento e uso desses modelos.
- **No capítulo 3**: Explicamos qual foi a metodologia seguida pelo trabalho para criação do experimento de co-criação e avaliação das pesquisas individuais.
- **No capítulo 4**: Apresentamos a técnica utilizada para definição das *features* e benefícios do experimento.
- **No capítulo 5**: Especificamos o desenvolvimento da plataforma e a interface do usuário. Além da estrutura para as pesquisas qualitativas e quantitativas.
- **No capítulo 6**: Compartilhamos os resultados quantitativos e qualitativos e levantamos pontos de discussão sobre os dados.
- **No capítulo 7**: Concluímos o trabalho pontuando aprendizados e pontos de melhoria. Como também, elencamos possíveis trabalhos futuros na área deste trabalho.

CAPÍTULO 2

Estado da Arte

Existem diversos tipos de modelos de inteligência artificial sendo utilizados para geração de textos. Alguns dos modelos são a GTP-3, o Megatron e o M6. Esses modelos se destacam por dois fatores: a quantidade de parâmetros utilizados e a quantidade de dados de treinamento. O GPT-3 [BMR+20], por exemplo, tem 175 bilhões de parâmetros e utilizou diversos datasets para o seu pré-treinamento. O dataset com maior impacto no pré-treinamento do GPT-3, o *Common Crawl* [com], tem 410 bilhões de tokens e constituiu 60% do dataset desse modelo.

Pela constituição desses modelos, com a sua grande quantidade de parâmetros e datasets, o poder computacional para treiná-los também é alta. Por isso, muito desses modelos não são 100% gratuitos. Outros modelos, com uma menor quantidade de parâmetros se diferenciam dos mais robustos, pela sua acessibilidade e custo. Custo de desenvolvimento é um dos desafios relacionados com esses modelos, no entanto, algumas críticas ao uso final e tipos de *outputs* [OD20] se tornam relevantes com esses tipos de modelos.

Não somente os modelos de IA impactam a criação de conteúdo, mas também como as ferramentas de co-criação utilizam desses modelos tão potentes. Para validar o uso dessas ferramentas com o usuário final e seus casos de uso, revisamos algumas plataformas de co-criação de texto para entender quais benefícios o usuário tem a utilizá-las.

2.1 Modelos de Linguagem Autoregressivos

Modelos de linguagem autoregressivos [Ho19] são modelos sequenciais, que predizem valores futuros através de valores passados, sem a necessidade dos dados recorrentes como nas RNNs (Recurrent Neural Networks) [She20]. Esses modelos autoregressivos são uma boa alternativa ao uso de RNNs para dados sequenciais e de GANs (Generative Adversarial Networks) [GPAM+14] para geração de conteúdo. Os modelos autoregressivos são uma boa alternativa ao uso de RNNs, pois os inputs do passados são adicionados como um estado qualquer do modelo, enquanto nas RNNs, os inputs passados são interpredados como uma camada intermediária.

2.1.1 GPT-3

GPT-3 significa Generative Pre-trained Transformer 3. Ele é um modelo de deep learning criado pela empresa OpenAI com 175 bilhões de parâmetros e treinado com mais de 400 bilhões de tokens. Esse modelo, desenvolvido em Maio de 2020, é capaz de criar textos que parecem ser escritos por humanos dado um *input* textual ao modelo.

GPT-3 foi treinado utilizando uma combinação de datasets que contém dados de páginas da internet, livros, e todo o conteúdo da Wikipedia disponível até o momento [Coo22]. Os datasets utilizados foram os:

- Common Crawl: 410 bilhões de tokens
- WebText2: 19 bilhões de tokens
- Books1: 12 bilhões de tokens
- Books2: 55 bilhões de tokens
- Wikipedia: 3 bilhões de tokens

O GPT-3 funciona como um modelo autoregressivo transformador. Modelos transformadores são modelos sequence-to-sequence (S2S) [Ver22] que produzem uma sequência de texto dado um texto de *input*. As suas principais funcionalidades são relacionadas a responder perguntas, resumo de textos, tradução de textos e geração automática de textos dado um *prompt*. Um *prompt* nada mais é do que uma frase descrevendo ao modelo a ação que ele deve tomar.

A OpenAI disponibiliza diferentes tipos de modelo, com diferentes habilidades e preços. Os preços são baseados em tokens, a representação base de cada palavra gerada por uma requisição a API[Ope21]. Os modelos podem ser utilizados no seu modo base, pré-treinados e sem modificação. Há também a possibilidade de utilizar esses modelos no modo *fine tuning*, em que o usuário pode treinar o modelo a partir dos seus próprios *datasets*.

Dada a sua estrutura generalista e de conhecimento de diferentes áreas, o GPT-3 tem a capacidade de realizar diferentes atividades na área da linguagem. Alguns exemplos:

- **Completar**: Gerar textos a partir de um *prompt*. Com um exemplo deste *prompt*: "gere um texto para minha loja de sapatos", temos a saída: "Oi! Seja bem-vindo à loja de sapatos. Aqui você encontrará os melhores sapatos para todas as ocasiões. Tenha um ótimo dia!". Esse modelo pode ser utilizado para uma gama extensa de aplicações.
- **Busca semântica**: Dado um conjunto de documentos, um score de semelhança semântica é retornado para uma query. Uma query é um conjunto de tokens que podem se relacionar com os documentos. Por exemplo, seja um conjunto de documentos definido por arquivos JSON (JavaScript Object Notation), com os documentos e a query abaixo:

  Documentos:

  - `{"text": "puppy A is happy", "metadata": "emotional state of puppy A"`
  - `{"text": "puppy B is sad", "metadata": "emotional state of puppy B"`

  Query: `"happy"`

  O resultado da busca semântica seria o documento `"document": "puppy A is happy"`. Esse exemplo de resposta foi retirado da documentação da OpenAI API [Ope21].

- **Ajuste fino**: Você pode treinar os modelos existentes com seus próprios dados.
- **Classificação**: Retorna *labels* para uma *query*, a partir de um conjunto de documentos classificados pelo usuário. Não há a necessidade de ajuste fino do modelo.
- **Perguntas e Respostas**: Uma *feature* interessante para produtos que tem uma base de conhecimento. Dado um conjunto de documentos para o modelo, ele consegue responder perguntas feitas a base de conhecimento.

2.1.2 Modelos Open-Source

Alguns dos modelos que contém uma quantidade menor de parâmetros comparado com o GPT-3 estão com suas licenças open-source habilitadas. Esses modelos foram utilizados para entender as possibilidades dos modelos de linguagem treinados com bilhões de parâmetros.

2.1.2.1 GPT-2

Esse modelo foi desenvolvido pela OpenAI, em fevereiro de 2019, [RWC+19] para entender as capacidades dos modelos de linguagem, a partir de diferentes quantidades de parâmetros nos modelos. Eles utilizaram o *dataset* WebText [RWC+19].

O último modelo treinado contém 1.5 bilhão de parâmetros, enquanto outras versões foram treinadas com 124 milhões, 355 milhões e 774 milhões de parâmetros.

O seu uso principal foi para entender os comportamentos, vieses e os limites de modelos geradores de linguagem em alta escala.

Já que os datasets de treino foram primariamente da internet, vieses de gênero, raça e religioso podem aparecer nos seus *outputs*. Além disso, o modelo não distingue realidade de ficção. Um outro problema é a capacidade do modelo de gerar conteúdo tóxico, ofensivo e fake news. Por isso, a equipe da OpenAI não recomenda o uso desse modelo com usuários finais, a não ser a critério de pesquisa.

Mesmo assim, a equipe ainda compartilha que alguns casos como assistência para escrita, escrita criativa para textos ficcionais e entretenimento podem ser bons casos de uso para essa ferramenta, cujo código está liberado no Github da OpenAI [1].

2.1.2.2 GPT-J-6B

Este é um outro modelo com licença open-source, desenvolvido por Ben Wang [Wan21]. Esse modelo têm 6 bilhões de parâmetros e foi treinando usando o The Pile [GBB+21] um dataset com 825 GB de tokens. Esse *dataset* é diverso e contem informação sobre diferentes domínios. Essa característica para o dataset é o que auxilia na capacidade de generalização desses modelos.

O modelo GPT-J-6-B pode ser testado nessa [2]página de demo. Na página de demo, o modelo permite a modificação de dois parâmetros importantes, top-p e temperatura. Esses parâmetros também estão disponíveis em outros modelos, como o GPT-3.

[1] https://github.com/openai/gpt-2

[2] https://6b.eleuther.ai

- **TOP-P**: controla o quão randômico será o texto gerado. Onde o seu valor de 0 a 1, irá controlar a distribuição de probabilidade de palavras comuns do vocabulário do modelo. 0 sendo selecionando as palavras menos comuns possíveis e 1 as palavras mais comuns do vocabulário.

- **Temperatura**: relacionado a entropia do sistema. 0 significa que a resposta sempre será a mais próximas dos exemplos do dataset em que o modelo foi treiando, enquanto 1 diminui a previsibilidade do sistema.

2.1.2.3 GPT Neo

Com o objetivo de ter as mesmas funcionalidades do GPT-3 e alguns diferenciais, o modelo GPT Neo utiliza da estrutura de modelos transformadores na sua rede [BGW+21]. Um dos diferenciais desse modelo é a adição de uma camada *Mixture-of-Experts* que aumenta o desempenho do modelo na modelagem de linguagem e tradução automática [SMM+17]. É possível treinar o modelo utilizando suas próprias configurações. Há alguns modelos da GPT Neo pre-treinados com o *dataset* The Pile e disponíveis para uso. Os dois modelos disponíveis utilizaram 1.3 bilhão de parâmetros e 2.7 bilhões de parâmetros.

Durante o desenvolvimento da GPT Neo, o time identificou a necessidade de compatibilidade para treinamento com GPU, além de TPUs. Por isso, a GPT NeoX foi desenvolvida com o objetivo de treinar um modelo com 175 bilhões de parâmetros com a sua nova estrutura. Atualmente, o modelo GPT-NeoX-20B [BBH+22] utiliza essa infraestrutura, treinado utilizando o The Pile e com 20 bilhões de parâmetros.

2.1.3 Modelos futuros

Desde o lançamento do GPT-3 em 2020, vários outros modelos de linguagem para fins generalizados têm surgido. A LifeArchitect.ai fez um apurado [3] sobre os últimos modelos lançados e os próximos que estão por vir. O PaLM [NC22] do Google Research conta com 540 bilhões de parâmetros e foi treinando com *datasets* multi-línguas, livros, Wikipedia e códigos do Github. Além disso, o GPT-4 já é esperado, mas até o momento da escrita desse documento não se tem tantas informações sobre sua quantidade de parâmetros e datasets.

Com o aumento da quantidade de parâmetros do PaLM, percebeu-se a evolução com relações a algumas atividades. Contexto, causa e efeito, e raciocínio foram alguns dos pontos que diferenciam esse modelo dos demais. Além disso, conclui-se que para geração de códigos de linguagem de programação, quanto maior a quantidade de parâmetros do modelo, maior a sua eficiência na geração de código.

Além do PaLM existem outros modelos na lista, como o OPT-175B da Meta AI com 175 bilhões de parâmetros [ZDZ], o Luminous da Aleph Alpha com 200 bilhões de parâmetros e o Chinchilla da DeepMind com 70 bilhões de parâmetros.

Na figura 2.1, podemos ver o crescimento na quantidade de modelos e nas suas quantidades de parâmetros. Cada um desses modelos tem suas particulares com relação a linguagem,

[3] Acessar o apurado em: https://docs.google.com/spreadsheets/d/1O5KVQW1Hx5ZAkcg8AIRjbQLQzx2wVaLl0SqUuir9Fs/editgid=1158069878 .

conjuntos de treinamento, e se o modelo é público ou privado. Dessa forma, podemos perceber que o desenvolvimento desses modelos está aumentando consideravelmente com o tempo.

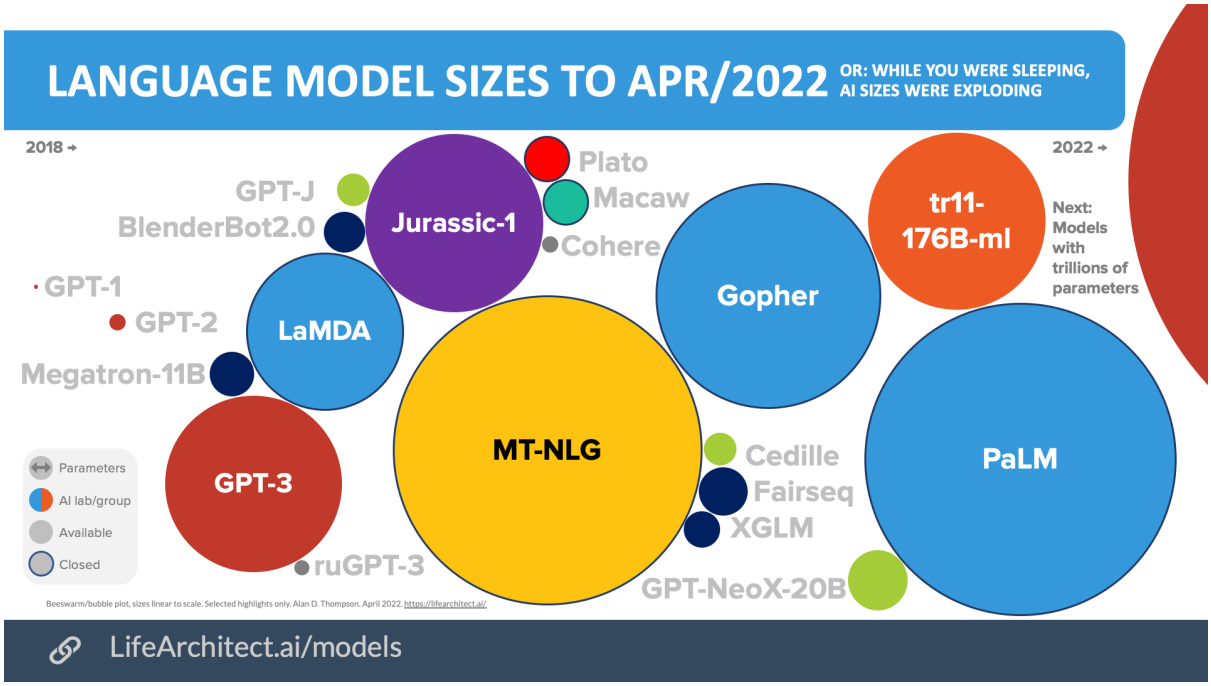

Figura 2.1 Estado atual dos modelos geradores de linguagens. O Eixo X é a data de lançamento e o tamanho das círculos a quantidade de parâmetros do modelo.. Infográfico desenvolvido pela LifeArchitect.ai [5].

2.2 Plataformas de co-criação

Os modelos de linguagem potencializam seus resultados ao mostrarem valor para os usuários finais. O principal valor, é tornar acessível os modelos de linguagem para usuários gerais, e não só usuários avançados, como desenvolvedores e pesquisadores. Nesse acesso, o valor é refletido em casos de uso que as plataformas disponibilizam e solucionam problemas reais dos usuários.

Esses valores são mostrados em diversas plataformas, que utilizam do modelo base dos modelos generalizados para resolverem problemas específicos. Além do *playground* da OpenAI onde usuários podem escrever *prompts* e visualizarem os resultados, existem plataformas com casos de uso definidos e que simplificam a vida de diversas pessoas que necessitam escrever no seu dia a dia.

2.2.1 Rytr

O Rytr se define como uma *AI Writer Assistant*. Com diferentes casos de uso, a plataforma consegue ter uma audiência distinta e benefícios que auxiliam a escrita, análise e correção dos

textos.

Ela está disponível em diferentes linguas, português incluída, e tem um *free tier* de até 5000 palavras geradas por mês. Rytr utiliza GPT-3 como o seu backend para a geração de textos [Ryta].

O diferencial dessa plataforma é a diversa quantidade de casos de uso. Ela contém mais de 30 casos de uso, como geração de seções para blog, copywrite para diferentes plataformas de anúncio digital, descrição de produtos, edição de texto, entre outras. Além disso, a plataforma ainda permite modificar o tom do texto gerado, como casual, formal e urgente. Como também, retornar diferentes variantes para um determinando caso de uso, e você escolhe com qual continuar.

Uma limitação dessa plataforma é o custo para utilização. Após 5000 caracteres gerados por mês, o seu nível gratuito acaba. E os planos acima do gratuito começam em 9 dólares por mês, independente de quantos caracteres você utilizar. Um critério de melhoria poderia ser o pagamento *pay-as-you-go*, onde a cobrança só existe para a quantidade de tokens gerados.

2.2.2 Jasper

A Jasper também se posiciona como uma *AI Writer Assistant*. Ela também conta com diversos casos de uso, mas sua audiência é o público empresarial, com casos de uso focados em vendas, SEO e escrita de copywrite. A plataforma não tem um nível gratuito [jasb].

Os casos de uso das duas plataformas são muito parecidos, mas essa plataforma conta com mais de 50 casos de uso. Ela também utiliza GPT-3 como o back-end para a geração de textos.

É interessante perceber que ambas a Jasper e Rytr usam o mesmo modelo gerativo de texto, mas que seus casos de uso são diferentes e talvez alguma seja mais eficiente do que a outra. A eficiência desses modelos depende de seus casos de uso: pode ser a originalidade da descrição, como nos casos de Marketing; o tempo de resposta do modelo para os casos em geral; ou a diversidade de casos de uso existentes que para resolução de problemas.

O diferencial nesse mercado parece se encaminhar para o *fine tuning* das plataformas, os casos de uso em específico para sua audiência e o custo de uso dessas.

Diferente do Rytr, o Jasper não tem um plano gratuito. Seus planos começam a partir de 29 dólares ao mês, para a geração de 20 mil palavras. Outro problema que usuários podem enfrentam ao utilizar o Jasper e o Rytr é a originalidade dos conteúdos gerados pela IA, com relação a necessidade do usuário. Não existe um ajuste fino dentro dessas plataformas, então o usuário tem que se contentar com o tipo de resposta que as plataformas dão. Já que as plataformas que definem o parâmetros dos modelos geradores de linguagem.

2.3 Desafios dos Modelos de Linguagem Natural

Em modelos de aprendizagem de máquina é necessário treinar, validar e colocar o modelo em produção. Além das dificuldades base que desenvolvedores desses modelos artificial enfrentam, outras dificuldades específicas aparecem para os modelos de linguagem generalizados. Alguns desses desafios foram verificados durante a fase de desenvolvimento do experimento e entrevistas individuais.

2.3.1 Custo de utilização

Primeiramente, mesmo que os modelos já estejam pré-treinados, ainda existe um custo para sua utilização. A OpenAI disponibiliza uma API para acesso do GPT-3, com 18 dólares por 3 meses (ver custos de uso na tabela 5.1). Entretanto, após o termino dos 3 meses não existe um *free tier* da API.

Então, poderíamos pensar em utilizar um dos modelos *open source* já que não existem "custos"relacionados a eles. Mesmo com as estruturas dos modelos em licença *open source* e com alguns já pré-treinados, ainda é necessário disponibilizar esses modelos para a sua ferramenta e usuários finais. Essa disponibilização, mesmo que de baixo escala, pode acabar gerando custos para o desenvolvedor. Por isso, muitos desenvolvedores acabam recorrendo a plataformas que contém *free tiers* que auxiliam no desenvolvimento de suas plataformas, como AWS [aws] e Google Colab [gcp].

2.3.2 Integridade e cautela

Sobre quais assuntos esses modelos geradores de textos podem falar? Eles podem falar com todo tipo de audiência, incluindo crianças? Eles podem dar instruções sobre sistemas críticos? E sobre saúde? Como um modelo gerador de texto parecido com textos humanos, é necessário cautela quanto ao uso final dessa ferramenta. Eventualmente, textos que deveriam ser fictícios, podem parecer reais, basta lembrar que partes do *dataset* de treino contém informações da internet, que não necessariamente são fatos.

Num artigo para a MIT Technology Review, Gary e Ernest [MD20] citam uma fala de seu colega Summers-Stay sobre a GPT.

> *GPT is odd because it doesn't 'care' about getting the right answer to a question you put to it. It's more like an improv actor who is totally dedicated to their craft, never breaks character, and has never left home but only read about the world in books. Like such an actor, when it doesn't know something, it will just fake it. You wouldn't trust an improv actor playing a doctor to give you medical advice.*

O ponto que o autor traz mostra o cuidado que devemos ter ao utilizar os resultados do GPT-3. Mesmo que o texto tenha sido gerado pelo modelo, não existe uma garantia de que ele não terá alguma informação incorreta ou alguma conclusão inesperada. Por isso, a co-criação e a participação do humano nesse processo é tão importante.

Depois de utilizar a plataforma da OpenAI por um tempo, de fato, em alguns momentos o modelo lembra de técnicas de improviso. Uma das técnicas que mais se utiliza em grupos de improviso, é a *Yes, and?* [McD19]. Nessa técnica, você nunca refuta o que foi dito ou mostrado numa cena anterior, você sempre continua a partir de onde a história está. A comparação com o GPT-3 se destaca quando alimentamos *propmts* ao modelo. Ele tende a completar o que já foi dito e a ideia principal, mantendo o estilo e criando a história que lhe for pedida. Não existe uma *flag* para diferenciar se o texto é fictício ou baseado em fatos.

2.3.3 Limitações para produção

Visto os possíveis problemas que uso do GPT-3 podem gerar, o time da OpenAI fez um compromisso com segurança e responsabilidade do uso da API [6]. Aplicações que forem ser expostas para o público em geral necessitam passar por uma revisão de uso pelo time da OpenAI para estarem de acordo com as políticas de uso e segurança. A critério de experimentação e pesquisa, não é necessário passar por essa revisão para o uso da API.

Essa camada de proteção é importante para garantir a segurança dos usuários finais, mas pode gerar uma fricção no desenvolvimento das aplicações. Além de que essa é uma política de uso definida pela OpenAI, o que pode ser diferente de políticas de uso de outras empresas também. Os critérios para definir se é necessário passar pela revisão de uso pela plataforma se encontram abaixo:

- Mais de 10 usuários utilizam a plataforma.
- Qualquer pessoa na web pode utilizar a sua plataforma.
- Você está cobrando para pessoas utilizarem sua plataforma.
- Você gasta mais de 250 dólares por mês.

2.3.4 Autoria

Um outro ponto de desafio é relacionado a autoria dos textos gerados pelo modelo. Esses textos são de autoria do humano que gerou o *prompt*? É uma autoria compartilhada entre o GPT-3 e o humano? Será que os desenvolvedores do GPT-3 também entram nesse grupo de autoria?

Para solucionar esse problema, perguntei ao GPT-3. Afinal, o modelo se propõe a responder qualquer tipo de pergunta. O diálogo com o GPT-3 se encontra abaixo:

THIAGO AUGUSTO : De quem é a autoria e licença dos textos gerados pelo GPT-3?

GPT-3 : Os textos gerados pelo GPT-3 são de autoria da OpenAI e estão sob a licença MIT.

THIAGO AUGUSTO : Posso utilizar esse texto no meu Trabalho de Graduação?

GPT-3 : Sim, você pode utilizar os textos gerados pelo GPT-3 em seu Trabalho de Graduação, desde que cite a fonte.

Como podemos ver, o próprio GPT-3 define que a autoria é da OpenAI sob licença MIT. A licença MIT basicamente permite o uso do conteúdo gerado por softwares livre e pagos [mit22]. Entretanto, será que esse texto gerado é verídico?

Esse desafio da veracidade do texto gerado apareceu durante as entrevistas individuais e pôde ser exemplificado na interação com a OpenAI. Será que de fato a licença dos textos gerados é a MIT? Ou será que o GPT-3 só está seguindo com o *Yes, and?* e nos dando uma resposta coerente?

Embora a reposta do GPT-3 fosse a licença MIT, ao olhar nos termos e usos da API da OpenAI [7], no tópico 2.(c), podemos ver a definição da autoria do conteúdo gerado:

[6] https://beta.openai.com/docs/usage-guidelines

[7] https://openai.com/api/policies/terms/

> *2. Using the APIs (c) Copyright. OpenAI will not assert copyright over Content generated by the API for you or your end users.*

Em vista disso, podemos utilizar os conteúdos gerados pela API da OpenAI, mas é necessário tomar cuidado com os seus resultados, como já havíamos relatado na seção 2.3.2.

CAPÍTULO 3

Metodologia

Neste capítulo, apresentamos três etapas para o atingir os objetivos de criação da ferramenta de co-criação de textos para internet e avaliação da experiência do usuário ao utilizar essa plataforma. As três etapas são:

- Exploração e definição das *features* e benefícios da experiência
- Desenvolvimento do Experimento Web
- Definição do Roteiro e Avaliação das Pesquisas

Na etapa de exploração e definição das *features*, a técnica Blue Ocean foi utilizada para identificar oportunidades e necessidades dos usuários finais.

Quanto ao desenvolvimento do experimento, foi tomada a decisão de utilizar ferramentas *web* focando em um processo mais simples. Para a escolha das tecnologias foi levado em consideração a velocidade de desenvolvimento, completude das ferramentas para o experimento e o menor custo possível que não comprometesse o experimento. Além disso, através dessa ferramenta, podemos visualizar a experiência dos usuários remotamente por compartilhamento de tela.

Quanto à entrevista, avaliamos traços qualitativos e quantitativos com uma entrevista semi-estruturada. Nesse formato, pontos importantes para avaliação são levantados durante a entrevista e interação do usuário com a experiência, mas o caminho pode ser diferente a depender do caminho que o usuário tomar. Um conjunto de ferramentas de análise e escala foram utilizados durante a pesquisa para avaliar a experiência geral e qualitativa dos entrevistados.

3.1 Pesquisa de Experiência do Usuário

A experiência do usuário é uma grande parte das plataformas de co-criação humano-computador. Uma das formas de avaliar essa experiência é realizar pesquisas que possam medir os resultados em aspectos relevantes a experiência e usabilidade.

O *User Experience Questionnaire (UEQ)* e o *System Usability Scale (SUS)* são dois questionários que validam a experiência de produtos interativos e aspectos clássicos de usabilidade. O SUS cobre a usabilidade do sistema, enquanto o UEQ cobre a experiência do usuário com o sistema. Dentre os aspectos avaliados estão atratividade, eficiência, criatividade e necessidade. O importante de utilizar essas pesquisas é de poder comparar, dentro de uma escala, como o artefato desempenha em relação a outros da mesma categoria.

3.1.1 System Usability Scale (SUS)

O SUS é o questionário padronizado de usabilidade mais utilizado no mundo [Lew18b]. O questionário valida a experiência do usuário em 3 aspectos: efetividade, eficiência e satisfação. Ele é um questionário mais rápido do que o UEQ, mas os dois se complementam em suas avaliações. A sua simplicidade facilita a pesquisa e não se torna algo cansativo para o usuário.

A pontuação do SUS pode ir de 0 a 100, com 68 pontos sendo a sua média. Seu resultado é calculado com perguntas numa escala de 1 a 5, que avaliam se o usuário atingiu o objetivo desejado e se tinha os recursos pra isso.

3.1.2 User Experience Questionnaire (UEQ)

O *UEQ* foi criado com o propósito de ser uma ferramenta de baixo custo e eficiente para medir quantitativamente a experiência do usuário (UX) [LHS08]. A área de UX é uma área subjetiva, onde usuários expõe suas experiências baseados tanto em experiências anteriores, como a experiência atual. Vieses e a não padronização das pesquisas podem influenciar no entendimento da experiência dos usuários. Nesse artigo, vamos utilizar a ideia do benchmark desenvolvido por Schrepp, Hinderks e Jörg Thomaschewski [SHT17] para avaliar o resultado das pesquisas individuais.

Basicamente o questionário avalia a experiência em 6 aspectos: atratividade, clareza, eficiência, confiabilidade, estimulação e inovação. As perguntas do questionário avaliam qualidades pragmáticas e hedônicas. Qualidades pragmáticas são aquelas baseadas no objetivo final do usuário; e as qualidades hedônicas na experiência e não no objetivo final.

Dentre os aspectos, Atratividade representa uma escala com a impressão geral da experiência pelo usuário. Estimulação e inovação representam qualidades hedônicas, enquanto as outras três representam as qualidades pragmáticas.

As notas do UEQ vão de -3 a 3, em cada uma das qualidades. Sendo -3 terrivelmente ruim e +3 extremamente bom. Valores entre -0.8 e 0.8 representam uma avaliação neutra. Valores acima de 0.8 uma avaliação positiva, e valores menos de -0.8 uma avaliação negativa.

CAPÍTULO 4

Blue Ocean

O Blue Ocean é uma técnica para atingir mercados e usuários que estão crescendo ou que ainda não foram contestados [Atr22]. O contrário do Blue Ocean, seria o Red Ocean. No Read Ocean, o mercado já é saturado com várias soluções e não há espaço para mais uma solução similar, ou pelo menos, é muito mais difícil desenvolver uma nova solução nesse tipo de oceano.

A oportunidade de identificar novas demandas e se diferenciar no mercado não contestado se torna mais atraente pela menor competição com outras empresas. Além disso, essa diferenciação e iniciativa pode definir como esse novo mercado irá atuar, precificar e até quais são os benefícios mais importantes para os usuários desse mercado. Alguns produtos que podemos destacar, que se aproveitaram do Oceano Azul, são a Netflix, Uber e iTunes. Ambos os produtos/empresas, resolveram um problema, através de novidades tecnológicas e de logística. Nos três casos, os problemas resolvidos já existiam, mas as novas soluções resolviam o problema com maior praticidade e qualidade.

Para esse trabalho, o objetivo é utilizar a técnica da Curva de valor, uma das ferramentas do Blue Ocean, para identificar lacunas no processo de criação e co-criação de texto para definição de benefícios diferenciados para os usuários desses mercados.

4.1 Curva de Valor

Levando em conta os critérios de comparação, a estratégia de curva de valor [KM14] consegue definir diferentes perfis e entender o posicionamento de produtos concorrentes. O objetivo de utilizar a curva de valor junto a ideia do Oceano Azul é de encontrar uma proposta de valor única, ao comparar a curva proposta com a curva dos concorrentes.

Nesse trabalho, para cada um dos benefícios identificados na seção 4.1.2 um valor de 0 a 10 será dado para cada um dos produtos analisados durante a exploração 4.1.1.

A avaliação dos produtos analisados podem ser verificados na imagem 4.1. A comparação entre os produtos foi colocada no mesmo ambiente, já que o mesmo usuário pode recorrer a ambos os produtos para solucionar problemas similares. Mesmo assim, a proposta do Serenpit Editor, encaixa-se dentro das *AI Writer Assistants*, que será o principal grupo para comparação para a proposta da curva de valor.

Podemos ver a curva de valor definida na figura 4.2. A principal ideia dessa proposta é facilitar a escrita, através de novos casos de uso para geração de templates. Além disso, usar dos próprios casos de uso para destacar ajustes gramaticais, como os benefícios vistos nas *AI Grammar Assistants*.

A comparação da curva de valor do Serenpit Editor com um representante das outras cate-

gorias pode ser visto na figura 4.3. A comparação da curva proposta com as outras *AI Writer Assistants* pode ser vista na figura 4.4.

O Serenpit Editor se destaca em relação as outras *AI Writer Assistants* pelos conteúdos gerados por templates. Mesmo que as outras soluções tenham uma quantidade maior de templates, as aplicações do Serenpit Editor focam na criação de textos para internet, em específico escrita de blog post. Quanto a comparação com as *Grammar Assistants*, o diferencial se apresenta pela capacidade de gerar textos. Por último, as *APIs* não são tão acessíveis para todos os usuários, por isso na nossa proposta, teremos um ambiente para interação direta com usuários, sem a necessidade de saber que existe uma API por trás.

Benefício	Serenpit Editor	Rytr	Jasper	Writer	Linguix	Grammarly	Open AI	Riku
Free Tier	10	5	0	10	10	10	5	0
Preço	5	6	6	6	0	3	5	6
Conteúdo gerado por prompt	10	10	10	0	0	0	10	10
Conteúdo gerado de templates	5	10	10	0	0	0	10	0
Reescreve textos	10	10	10	5	5	5	10	0
Criar caso de uso da IA	0	10	0	0	0	0	10	0
Usar diferentes IAs para formular tex	0	0	0	0	0	0	10	10
Plataforma para escrita de textos	10	10	10	10	10	10	0	0
Disponível em Português	10	10	-	0	0	0	10	10
Avalia Erros Gramaticais	10	0	0	10	10	10	0	0
Monitora a qualidade da escrita	0	0	0	10	10	10	0	0

Figura 4.1 Score atribuido a cada benefício dos produtos analisados e da solução proposta.

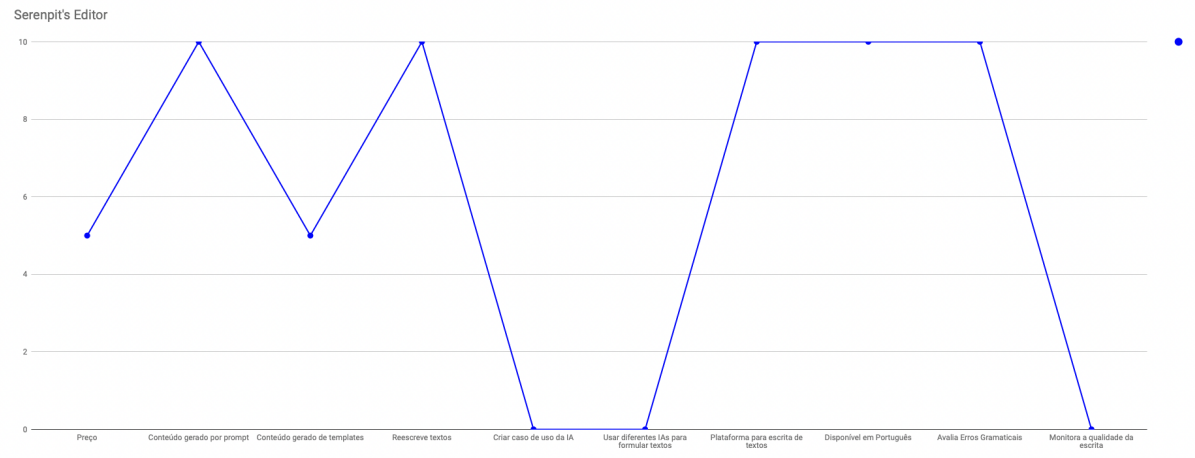

Figura 4.2 Curva de valor proposta para o Serenpit Editor.

4.1.1 Exploração

Além das plataformas mencionadas na seção 2.2, outras plataformas de *AI Writer Assistants* foram exploradas e avaliadas. Um outro ponto importante dessa exploração, é verificar mer-

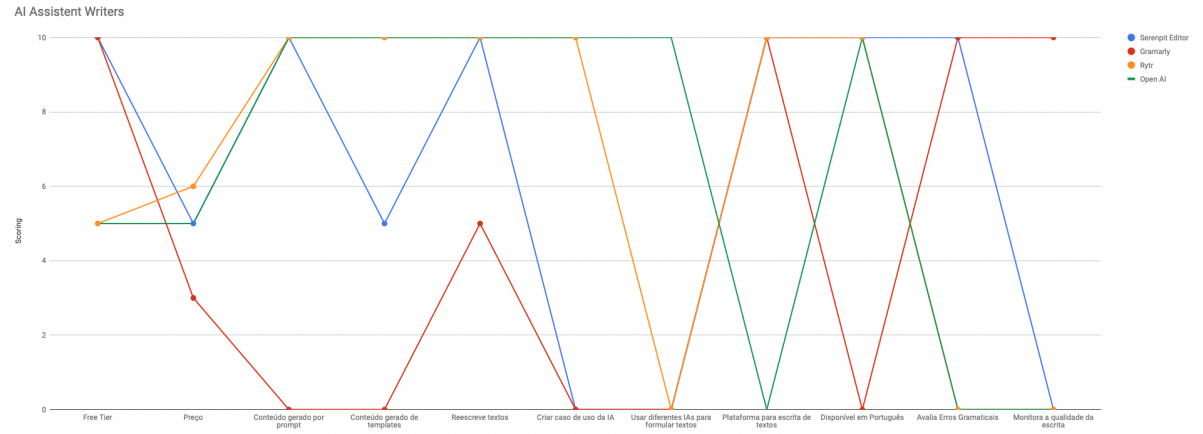

Figura 4.3 Comparação curva de valor do Serenpit Editor com um representante de cada categoria de produto explorada: *AI Writer Assistants*, *Grammar Assistants* e APIs geradoras de linguagem natural.

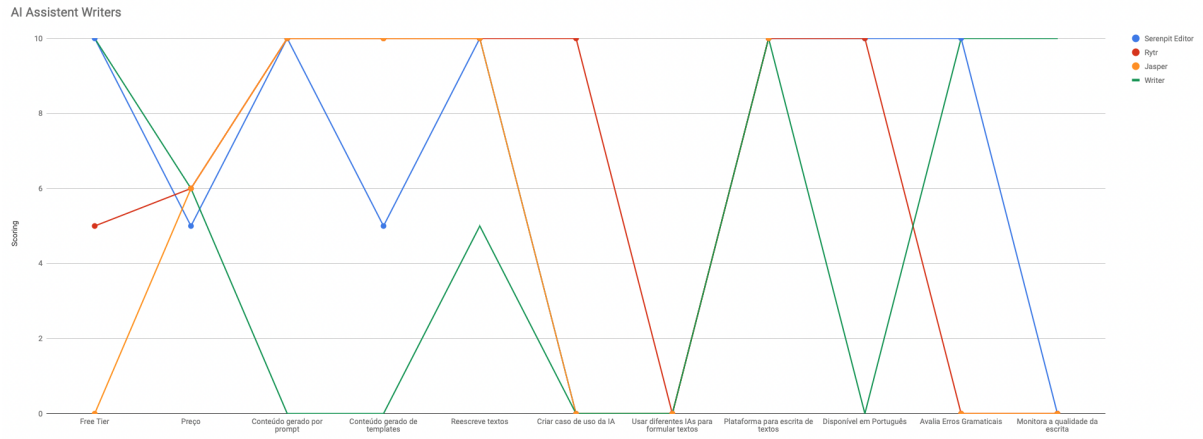

Figura 4.4 Comparação curva de valor do Serenpit Editor com os produtos da categoria *AI Writer Assistants*

cados próximos aos das *Writer Assistants*, para identificar outras oportunidades. Os outros dois grupos explorados foram o de *AI Grammar Assistants* e de APIs de modelos geradores de linguagem natural, como a OpenAI.

A diferença das *Grammar Assistants* para as *Writer Assistants* é o seu foco muito maior em avaliar textos já escritos e providenciar sugestões de mudança. Elas tem uma maior aplicação em organizar os textos e prover *feedback* para os escritores, do que geração de textos automáticos e desbloqueio de criatividade.

O critério para exploração dessas plataformas foi de acessar as demonstrações disponíveis (nas que estavam disponíveis) e analisar as *features* destacadas nas suas *landing pages*. Na exploração foram analisados os produtos, nas três categorias:

- *AI Writer Assistants*: Rytr, Jasper, Writer [wri22].

- *AI Grammar Assistants*: Linguix [lin], Grammarly [gra].
- APIs geradoras de linguagem natural: OpenAI, Riku [rik].

4.1.2 Features e Benefícios

Da exploração, percebemos que as *AI Writer Assistants* são normalmente um conjunto de duas soluções diferentes: um editor de texto + acesso abstraído às APIs geradoras de linguagem natural. Soluções como: re-escrever um texto, gerar parágrafos a partir de temas, e até *prompts* em aberto para as plataformas. Na figura 4.5, podemos verificar algumas das *features* analisadas nesse tipo de assistente. Um outro detalhe, são as aplicações de uso para determinados nichos, como conteúdos para divulgação de marketing, *Copywriting* e *SEO* para páginas web.

Em paralelo, na exploração das *AI Grammar Assistants*, notamos a aparição de estatísticas de qualidade do texto, sugestões de melhorias no texto e correção de erros gramaticais. Essas plataformas assistentes também têm os seus próprios editores de texto, mas é importante destacar que elas também contam com extensões para *browsers* que encontram o usuário onde ele estiver escrevendo. Como o seu objetivo é auxiliar na correção e melhoria de textos, essas extensões podem dar sugestões dentro de qualquer caixa de texto dentro de páginas web.

As APIs, dada a sua natureza, têm um público muito mais focado em *power users*, como desenvolvedores, pesquisadores e usuários avançados. Mesmo com esse foco de usuários, as APIs contam com *playgrounds*, para teste e exploração de suas features. Mesmo assim, as APIs também tem diferentes casos de uso, como as *Writer Assistants*, para mostrar a sua capacidade e o que pode ser feito com elas.

Dado a exploração dessas ferramentas, elencamos os benefícios mais importantes para a análise do mercado e identificação do nosso oceano azul. Esses benefícios são:

- **Free Tier**: Se existe uma camada gratuita de teste e quais features estão disponíveis.
- **Preço**: Custo de utilização da plataforma e dos seus diferentes níveis de uso.
- **Conteúdo gerado por *prompt***: Se existe a possibilidade de interagir com a plataforma através de prompts.
- **Conteúdo gerado de templates**: Se através de palavras-chave, ou um tema, é possível gerar conteúdos textuais. Por exemplo, criar uma história de três parágrafos a partir de cinco palavras aleatórias.
- **Reescreve textos**: Aperfeiçoamento, transformar os textos em formais ou informais, reduzir frases extensas.
- **Criar caso de uso da IA**: A partir de dados do usuário, criar um novo caso de uso. Classificação de conteúdos ou exemplo de resposta para *prompts* são alguns exemplos.
- **Usar diferentes IAs para formular textos**: Se existe a possibilidade de utilizar outras IAs, além do GPT-3.
- **Plataforma para escrita de textos**: Se existe um editor de texto para usuários finais.

- **Disponível em Português**: Não só a plataforma utilizada pelos usuários, mas também as features das IAs.
- **Avalia Erros Gramaticais**: Além de avaliar, propõe ajustes sintáticos.
- **Monitora a qualidade da escrita**: Juntamente com a avaliação dos erros gramaticais, propõe sugestões e melhorias para a escrita. Seja as modificações propostas a nível gramatical, ou no estilo do texto.

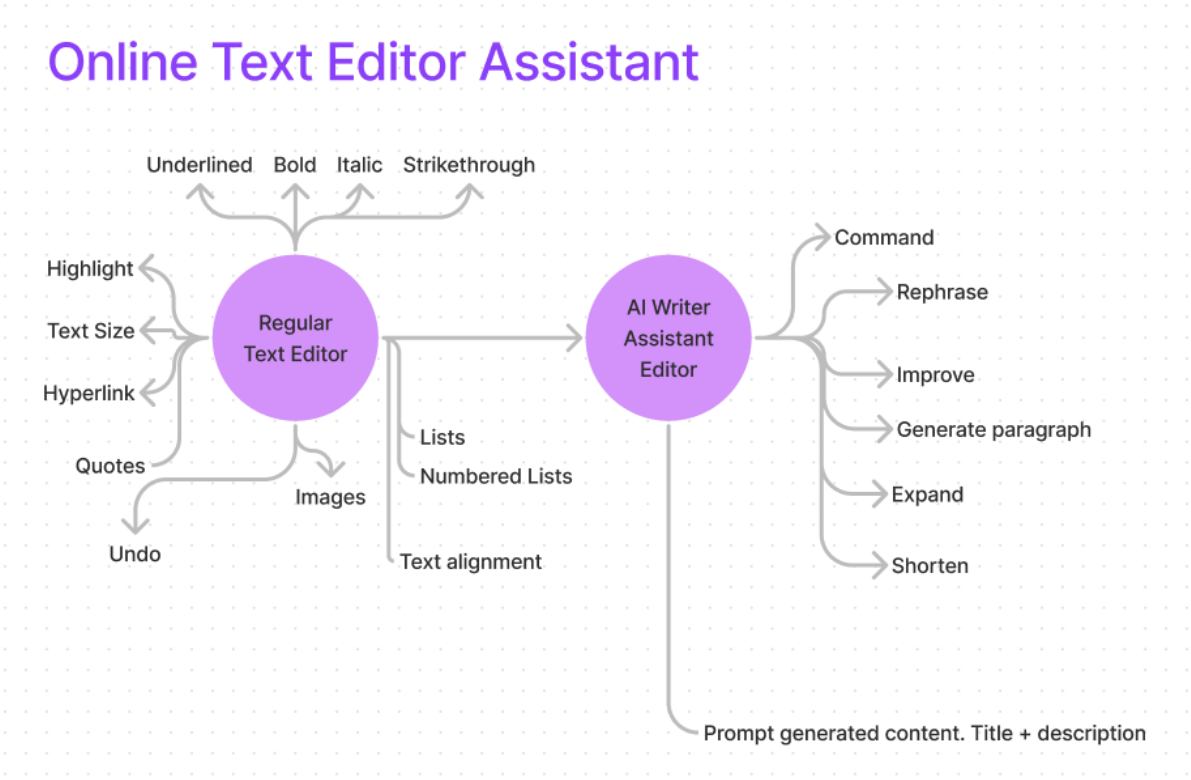

Figura 4.5 Exploração e documentação das features encontradas nas *AI Writer Assistants*.

CAPÍTULO 5

Desenvolvimento da Ferramenta

Para simplificar a criação do experimento, foi tomada a decisão de criar a plataforma de co-criação utilizando tecnologias web. Além de conhecimento prévio nas tecnologias abaixo, a utilização de web em detrimento de mobile se deu pela possibilidade de atingir mais usuários para as pesquisas individuais e pela possibilidade de atualizar a plataforma para ser responsiva no futuro.

5.1 React

React é um framework desenvolvido pelo Facebook Open Source para facilitar o desenvolvimento de interfaces do usuário interativas [rea]. React tem uma estrutura para criação e reutilização de componentes, como também uma comunidade ativa que conta com várias bibliotecas e componentes *open source*.

Além da facilidade para criação de interfaces do usuário, o *framework* funciona de forma declarativa, simplificando o fluxo de desenvolvimento web. HTML, CSS e Javascript encontram-se num mesmo ambiente e conseguem conversar de forma mais fácil. Outro conceito importante do *framework* são os estados. Esses estados criam um ambiente em que a interação com os componentes reage as ações do usuário e podem salvar suas decisões.

Neste projeto, foi utilizado para desenvolvimento do frontend da aplicação. A interação com o backend também foi feita utilizando ferramentas do Javascript em conjunto com os estados dos componentes React.

5.1.1 Chakra UI

Chakra UI é uma biblioteca de componentes React *open source* que simplifica a criação de blocos e interfaces para os usuários [cha]. Essa biblioteca conta com componentes modulares, simples e acessíveis. Ao invés de customizar os componentes HTML do zero, a base para botões, *containers*, *pop-ups*, títulos, parágrafos, entre outros componentes, já estão disponíveis. É só utilizá-los diretamente.

Outros benefícios da biblioteca são a facilidade e possibilidade de customização dos componentes, a partir de um só arquivo de configuração; os componentes já seguem os padrões de acessibilidade *WAI-ARIA*; os componentes são customizados para terem múltiplos temas, como temas escuros e claros. Não obstante, a comunidade tem um papel fundamental para que essa biblioteca seja tão famosa e utilizada. Vários kits iniciais e plataformas de exemplo utilizam o Chakra, o que trás um maior apoio ao encontrar desafios utilizando a biblioteca.

Neste projeto, foi utilizado para construção de todos os componentes utilizados pelo React. Os componentes base do Chakra facilitaram o desenvolvimento da aplicação e adicionaram os benefícios de acessibilidade e temas para os componentes.

5.1.2 Draft.js

Draft.js é um *framework* para editor de texto feito em React [dra]. A biblioteca foi desenvolvida pelo Facebook Open Source e conta com um editor declarativo, que facilita o uso em aplicações React. O desenvolvedor não precisa se preocupar em como o *input* da área é manipulado, ou como seleções são feitas. O framework é como uma API funcional e todas as aplicações comuns de um editor de texto foram abstraídas.

Bem como o Chakra UI, esse framework também conta com uma forte comunidade que cria *templates* em cima da estrutura inicial. Esses *templates* da comunidade já conta com diversas aplicações básicas de um editor de texto: editor de estilos, seleções de texto, *tooltips*(textos que aparecem ao passar o mouse por cima de algo), e estados do texto salvos. Esse *framework* é altamente customizável e extensível.

Neste projeto, foi utilizado para criar o componente editor de texto. Além de ser uma biblioteca compatível com React, ela adicionou a possibilidade de estilização e de adicionar os textos gerados pelo GPT-3.

5.2 Node.js

Node.js é um ambiente para execução de código javascript no lado do servidor [Nod]. Para a criação do experimento deste trabalho, Node.js será utilizado para a interação com a OpenAI API. Por utilizar a mesma semântica de Javascript, Node.js facilita a experiência do desenvolvedor front-end, com a utilização de uma mesma linguagem para o frontend e backend.

Neste projeto, foi utilizado para implementar as funções que interagem com o GPT-3 e são hospedadas na Netlify.

5.3 Netlify

Netlify é um ecossistema de ferramentas que num único workflow permitem o deploy de aplicações web [net]. Netlify é baseado nas soluções da Amazon Web Services (AWS) [aws], mas que de forma abstraída simplificam a vida e o tempo para construções de aplicações web para os desenvolvedores.

Algumas das soluções da mais famosas da Netlify são as Content Delivery Network (CDNs) [cdn] para *hosting* das aplicações finais usadas pelos usuários; funções *serverless* que podem ser invocadas pelo *frontend*; além dos *webhooks* existentes na plataforma, que facilitam os fluxos de *Continuous Integration/Continuous Delivery* (CI/CD) [SABZ17]. Inclusive, a netlify contém um generoso *free tier* para *build* de aplicações e de *bandwith*.

Neste projeto, foi utilizado para hospedar a aplicação web e as funções serverless que interagem com o GPT-3. A hospedagem da aplicação utiliza um dos serviços de CDN da Netlify e

contam com um serviço de monitoramento da saúde tanto da aplicação geral, como das funções *serverless*.

5.4 OpenAI API

A OpenAI API fornece o acesso ao GPT-3. Diversas empresas utilizam a API da OpenAI diretamente, ou indiretamente pelas plataformas vistas na seção 2.2. Cada um dos modelos disponíveis na plataforma tem o seu tempo de resposta, mas que ainda assim são tempos de resposta baixo. A API é escalável e flexível, podendo ser treinada para atingir casos de uso em específico com o *fine tuning*, ou utilizando dos modelos base para solução dos problemas.

Dentro da plataforma da OpenAI enconta-se um *playground*, onde os usuários podem começar a testar *prompts* e casos de uso como: correção gramatical; cálculo de complexidade de tempo de um algoritmo; classifação de tweets; explicação de código; transpilador de javascript para python; criador de perguntas para entrevista; descrição de produtos; entre vários outros.

Dentro do *playground*, e nas chamadas da API, é possível modificar alguns parâmetros que influenciam no *output* do modelo. Os parâmetros mencionados na seção 2.1.2.2 também estão disponíveis por aqui, mas além deles temos:

- **Quantidade máxima de tokens**: o limite máximo de tokens gerados pela resposta do modelo. O máximo pode varia a depender da *engine*[1] utilizada, mas o valor vai de 2048 a 4000 tokens. Onde 1000 tokens são aproximadamente 750 palavras.
- **Penalidade de frequência**: o quanto é penalizado pela repetição de novos *tokens* no texto gerado. O valor vai de 0 a 2, onde 0 é não existir penalização. Essa configuração diminui a probabilidade de repetição de palavras.
- **Penalidade de presença**: o quanto é penalizado pela aparição de um *token* já existente no texto gerado. O valor vai de 0 a 2, onde 0 é não existir penalização. Essa configuração diminui a probabilidade de repetição de ideias.
- **Melhor de X**: Gera diversas respostas ao pedido do servidor e escolhe a melhor. O máximo de opções para comparação é 20 e o *default* é 1 opção.

O modelo de precificação da OpenAI é *pay as you go*, você só paga se usar e o quanto usar. Na tabela 5.1, podemos ver o custo de utilização das *engines* no modelo base e do mesmo modelo em seu modo avançado. Além disso, a OpenAI disponibiliza $18 para testes da plataforma por 3 meses.

[1]Uma engine é um modelo de linguagem natural do GPT-3. A API disponibiliza diferentes tipos de *engines*, com capacidades computacionais diferentes, a depender do caso de uso dos usuários. Mais em: https://beta.openai.com/docs/engines .

Modelo	Custo Requisição Modelo Base *	Custo Treinamento *Fine Tuning* *	Custo Requisição *Fine Tuning* *	Benefícios
Ada	$0.0008	$0.0004	$0.0016	Mais rápido
Babbage	$0.0012	$0.0006	$0.0024	Atividades simples
Curie	$0.0060	$0.0030	$0.0120	Muito capaz e mais rápido que o Davinci
Davinci	$0.0600	$0.0300	$0.1200	Mais poderoso

Tabela 5.1 Modelos disponíveis na API da OpenAI. *Custo por 1k de tokens.

5.5 Desenvolvimento do Experimento

O experimento é acessado através de um *web browser* e conta com um *backend* que se comunica com o GPT-3 através da OpenAI API. O usuário acessa o experimento através de uma rota definida no meu blog pessoal [2].

Através da análise realizada no Blue Ocean e a curva de valor, alguns benefícios para o usuário se destacaram. Esses benefícios pertencentes as *features* existentes na curva de valor do Serenpit Editor foram selecionadas como parte do experimento.

5.6 Definição das features

Dada a exploração, e necessidades dos usuários para escrita de textos, propomos as seguintes *features* para a plataforma:

- **Editor de texto**: Área para produção textual. Tanto o usuário pode escrever novos conteúdos, como também as features de co-criação interagem nessa área. Além disso, conta também com estilização do texto: negritos, tamanho da fonte, itálico, sublinhado e estilo de código de programação.
- **Ações rápidas**: Essas ações podem ser tomadas ao selecionar qualquer texto dentro da área de edição de texto.
  - **Expandir**: A partir do texto selecionado, expandir a ideia geral daquele texto.
  - **Encurtar**: Às vezes, um parágrafo, ou frase está muito extenso. A ideia de encurtar é transformar o texto selecionado em algo mais objetivo.
  - **Aperfeiçoar**: Essa ação não, necessariamente, aumenta o texto selecionado. O objetivo é transformar o texto selecionado em algo mais formal.

[2] https://serenpit.com/editor

  - **Criar parágrafo**: Algumas frases contém ideias que poderiam se transformar num próprio parágrafo. Esas ação expande frases e cria novos parágrafos.

- **Ações avançadas**: Essas ações são tomadas a partir de *inputs* dos usuários. Cada uma das ações espera diferentes informações do usuário, podendo ser simples frases, múltiplas informações, ou uma lista de dados.

  - **Escrever seu *prompt***: Aceita uma instrução genérica para o GPT-3. Funciona como uma conexão direta a OpenAI API.
  - **Criar um blog post**: Através de um tema inicial é criado uma estrutura de blog, preenchida com dados sobre o conteúdo.
  - **Listar tópicos**: A partir de um tema, uma lista enumerada é criada para o usuário.
  - **Descrição do Produto**: Duas informações são necessárias para essa ação: o nome do produto e uma descrição inicial do produto.
  - **Conte uma história**: Com uma lista de, no mínimo, 5 elementos, uma história é criada.
  - **Resumir para uma criança**: Um texto complexo é resumido para que uma criança possa entender.

- **Múltiplos documentos**: Normalmente, em plataformas de edição de texto, usuários podem criar diferentes conteúdos em diferentes documentos. Essa *feature* foi criada para estar de acordo com a experiência de usuários em outras plataformas, no qual isso já existe.

  - **Salvar automaticamente**: Por algum motivo o usuário pode atualizar a página, perder acesso a internet, ou abrir uma nova aba. Em todos esses casos, se não existir o salvamento automático, a experiência pode ser prejudicada. Por isso, a necessidade de salvar o texto enquanto ele está sendo escrito.
  - **Documento de demonstração**: *Onboarding* é a introdução de usuários a um novo fluxo ou produto. Numa nova plataforma, é importante ter algum direcionamento se o usuário não souber por onde seguir. Por isso, um documento pré-populado com conteúdos já demonstra a utilização da plataforma.

- **Configuração**: Seguindo as diretrizes da própria OpenAI, essa plataforma por estar em uma URL pública, deveria passar por um critério de qualificação para ser pública para usuários em geral. Por isso, uma chave de utilização é adicionada as configurações para o uso completo da plataforma.

5.7 Interface de Usuário (UI)

Criamos a interface para o usuário no formato desktop. A página inicial pode ser vista na figura 5.1 e acessado na página do Serenpit Editor[3].

[3]https://serenpit.com/editor

Por estar dentro do meu blog pessoal, a interface já usufrui de três *features* prévias: página em inglês e português; modo *dark mode* (figura 5.2); e a roleta da serendipidade. A roleta da serendipidade encontra um blog post publicado em `serenpit.com` e redireciona o usuário para lá.

Na plataforma, do lado esquerdo está o editor de texto, com as opções de estilização e logo abaixo, as ações rápidas. No lado direito, temos as abas de ações avançadas, documentos e configurações.

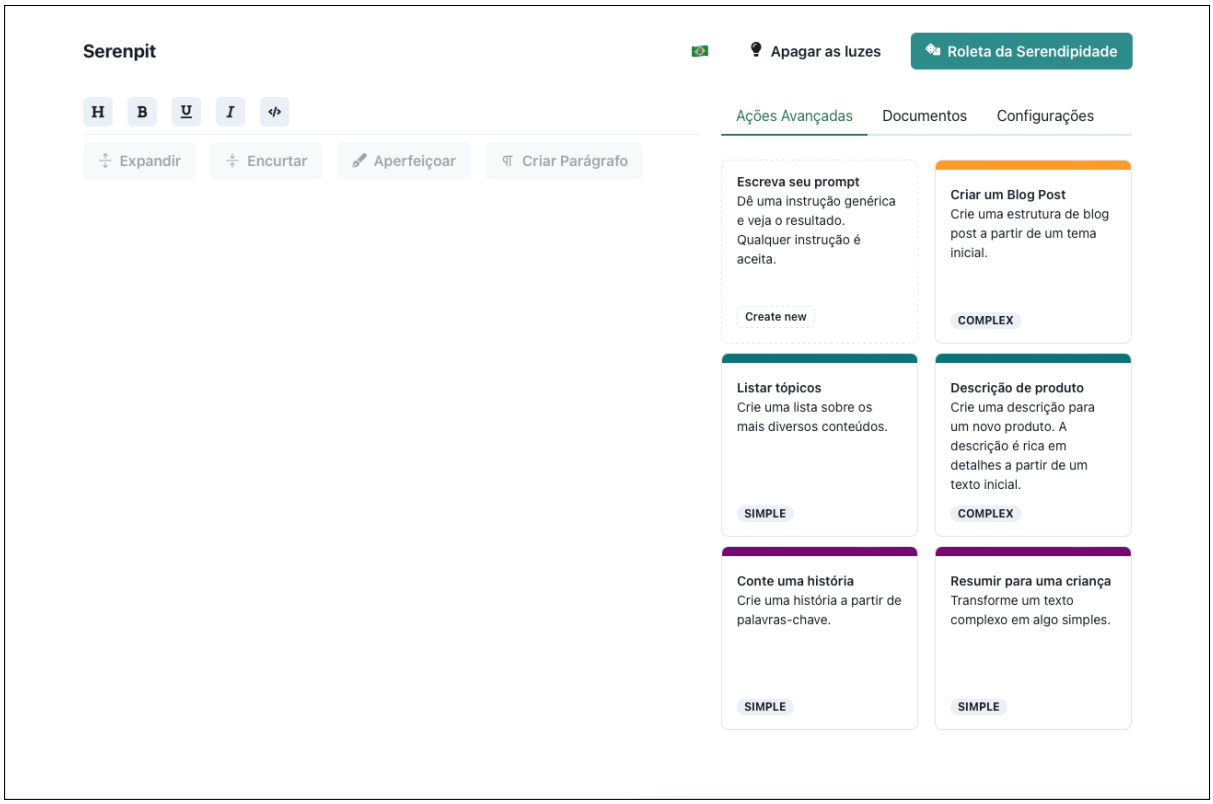

Figura 5.1 Página inicial do experimento.

5.7.1 Ações rápidas

As ações rápidas são executadas após a seleção de algum texto dentro do editor de texto. Ao receber a resposta do servidor, o novo texto atualizado é adicionado também ao editor de texto.

5.7.2 Documentos

A aba de documentos dá acesso a dois documentos: um documento que não contém nenhum conteúdo, e um outro que já contem um modelo de *blog post* com o tema "Como negociar salários em tecnologia". O segundo documento é utilizado como demonstração de textos da plataforma. O *blog post* foi gerado com a própria ação avançada "Criar um Blog Post". A aba

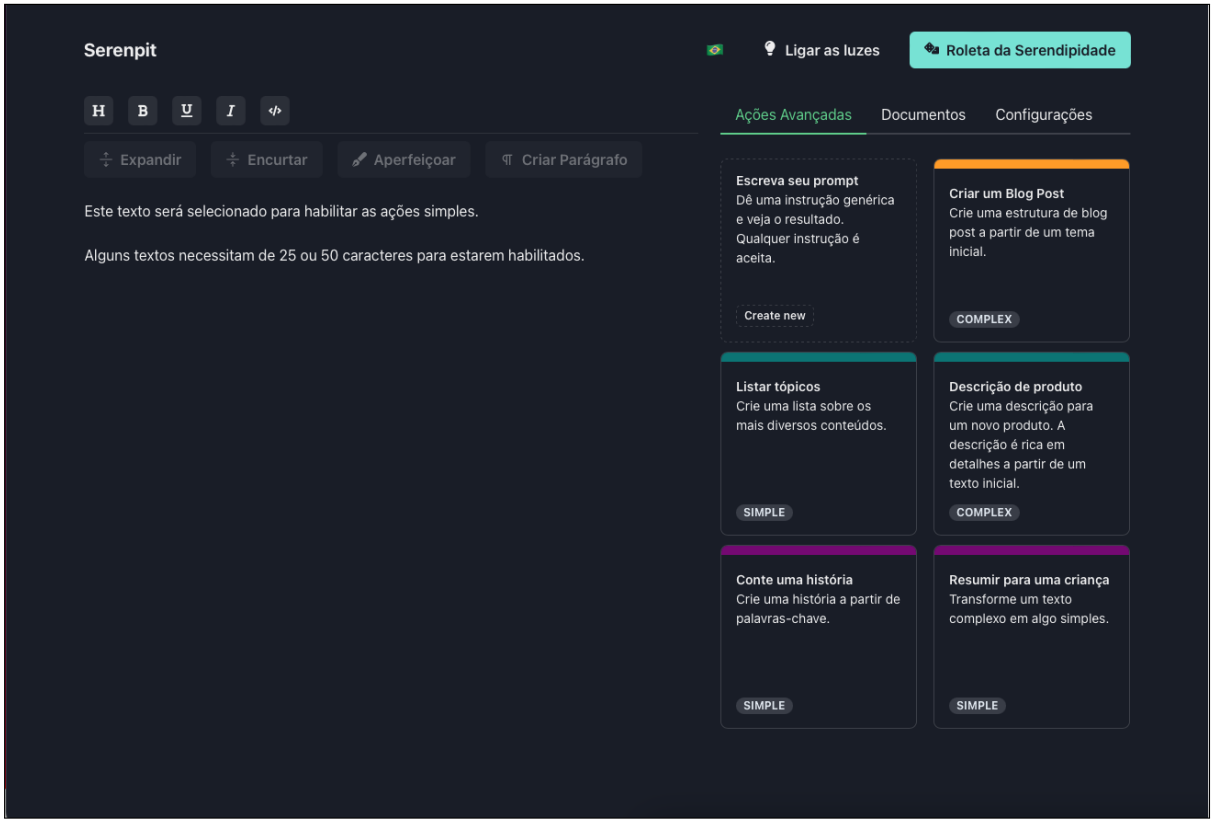

Figura 5.2 Página inicial do experimento em *dark mode*.

de configurações pode ser vista na figura 5.2. O documento de demonstração pode ser visto na figura 5.4.

5.7.3 Ações Avançadas

As ações avançadas aparecem num modal após selecionadas. Após o conteúdo ser preenchido, o resultado da API é adicionado ao campo do editor de texto. Cada uma das ações tem diferentes campos para serem preenchidos. "Escreva seu prompt", "Criar um Blog Post", "Listar Tópicos"e "Resumir para uma criança"recebem textos corridos, mas que o modo de escrever influencia o resultado final da API. Em cada uma das ações avançadas, o *placeholder* do campo de texto dá um exemplo de como interagir com aquela ação. Um exemplo pode de ação avançada pode ser visto na figura 5.5.

Na ação "Descrição de produto"existe mais um campo de texto para adicionar o nome do produto. Enquanto que, a ação "Conte uma história"é a interação mais diferente dos modals. Essa ação recebe uma lista de palavras-chave para gerar a história. A UI pode ser vista na figura 5.6.

Figura 5.3 Aba de Documentos.

5.7.4 Quantidade mínima de caracteres

Fizemos diversos testes internos antes de realizar as entrevistas com os usuários. O principal problema verificado durante esses testes foram as respostas que não realizavam a ação pedida. Por exemplo, ao utilizar a ação "Expandir", o texto era retornado em inglês; ou ao utilizar a função "Aperfeiçoar"o mesmo texto era retornado. Para contornar esse problema, adicionamos uma quantidade mínima de caracteres para a utilização das Ações Rápidas e Avançadas.

Em algumas ações, a quantidade de caracteres foi maior do que em outras, devido a necessidade de mais informações para a tomada de ação. No caso de "Encurtar", não faz sentido encurtar um texto que já fosse curto, por isso o mínimo de 50 caracteres. Enquanto que para "Expandir"o mínimo necessário é de 25 caracteres.

No caso das Ações Avançadas, a limitação de caracteres foi introduzida na quantidade escrita dentro dos campos de *input*. Ao mesmo tempo que nas Ações Rápidas, a limitação foi implementada a partir das palavras selecionadas do texto. Enquanto não existir a quantidade mínima de palavras selecionadas, os botões não são habilitados. Para ajudar na explicação de como esses botões são habilitados, um *tooltip* foi adicionado ao passar o mouse em cima dos botões. Podemos visualizar os botões habilitados e o tooltip na figura 5.7.

5.8 Backend OpenAI

A *engine* text-davinci-001 foi a selecionada para consumir o GPT-3 pela OpenAI. Essa foi a engine escolhida devido a sua capacidade computacional e de resolver tarefas, mesmo que tenha menos contexto para realizar as ações. Foi utilizado o *entrypoint* de *completion* da API, que server para resolver tarefas variadas. Os parâmetros utilizados da engine foram os *default* do modelo, menos a quantidade máxima de tokens, que foi modificada para 1200. O backend foi implementado em Node.js e hospedados nas funções lambda da Netlify.

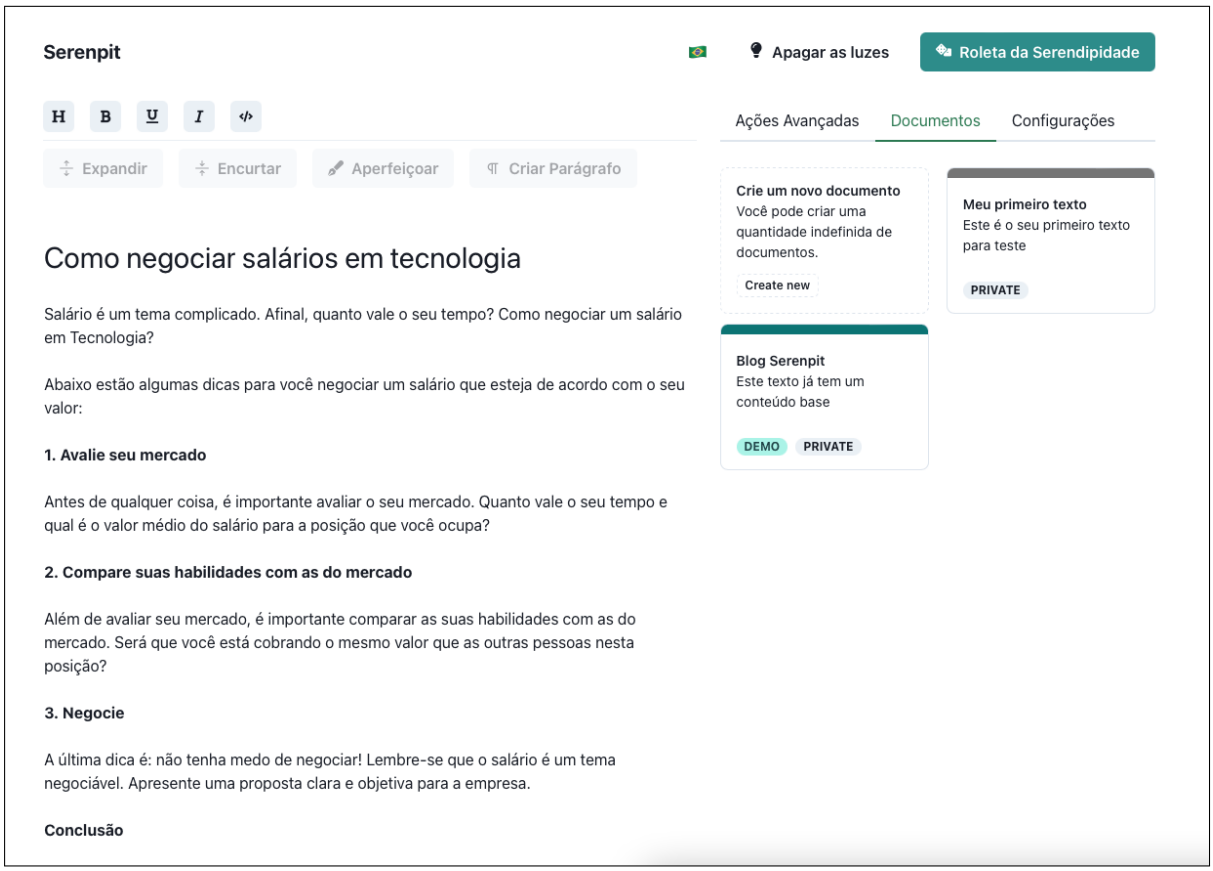

Figura 5.4 Página inicial do experimento com o documento de demonstração aberto.

A princípio, queríamos utilizar o máximo possível de tokens para enriquecer a resposta para os usuários. Entretanto, quanto maior a quantidade de tokens gerados, maior o tempo de resposta da API. Como o backend que acessava a API da OpenAI foi das funções lambda da Netlify, não podíamos ultrapassar 10 segundos de tempo de processamento, já que esse é o limite definido pela Netlify. Se ultrapassar 10 segundos, a função na Netlify retorna um erro de *timeout* por padrão. Por isso, escolhemos utilizar uma quantidade menor de tokens.

5.8.1 Segurança chave de API

Para seguir a política de uso da OpenAI API, a plataforma está disponibilizada ao público, mas para utilizar qualquer uma das ações é necessário adicionar uma chave na aba de configurações, ver figura 5.8. Essa chave, não é a chave de API que OpenAI disponibiliza para uso, mas sim uma chave, ao qual checamos no lado do servidor para validar o usuário. Essa chave é então compartilhada com os usuários no momento das pesquisas individuais.

Ambas as chaves de acesso a OpenAI e a chave pública dos usuários é armazenada nas variáveis de ambiente da Netlify. A chave pública é checada em todas as chamadas ao backend, como vemos no listing 5.1. Se a chave não for válida, é retornado um erro de autenticação para o frontend.

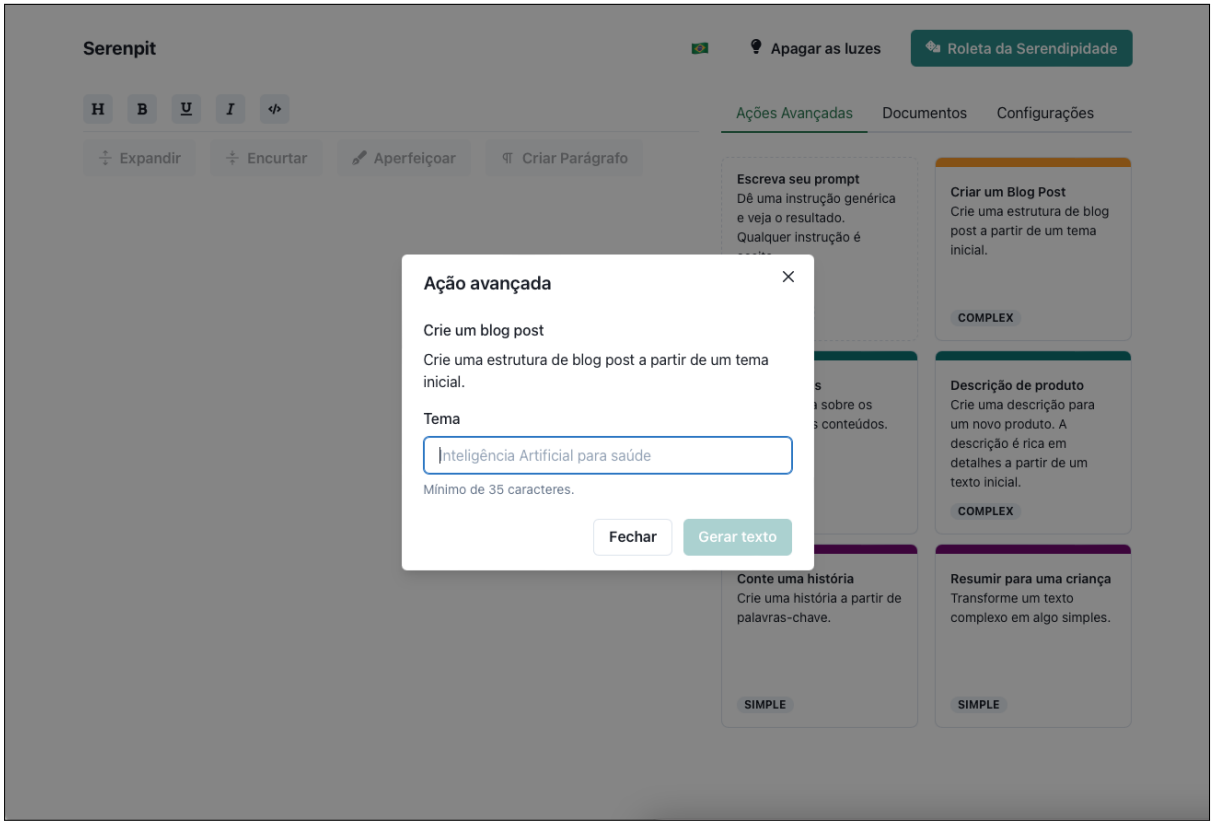

Figura 5.5 Ação avançada "Criar um Blog Post".

```
1 if (apiKey != process.env.OPENAI_PUBLIC_API_KEY) {
2   return {
3     statusCode: 401,
4     body: 'API Key is not authorized',
5   }
6 }
```

Listing 5.1 Código de proteção ao acesso público em Node.js.

5.8.2 *Prompt Programming*

Para cada uma das ações, diferentes prompts são enviados para a OpenAI. Nessa etapa, testamos diferentes frases para entender como o GPT-3 interpretava as requisições e se suas respostas estavam coerentes. A ação de conversar com o GPT-3 e ver seus resultados está alinhada com um novo paradigma de desenvolvimento, o *Prompt Programming* [RM21]. Nesse modelo de desenvolvimento, plataformas podem usufruir da interação via linguagem natural com esses modelos geradores de texto.

Como fazemos o pedido para o GPT-3 pode influenciar na sua resposta. Durante os testes, principalmente da ação "Criar um blog post", precisamos testar diferentes frases até chegarmos num resultado esperado. Explicitamente dizemos para o GPT-3 gerar três seções e uma con-

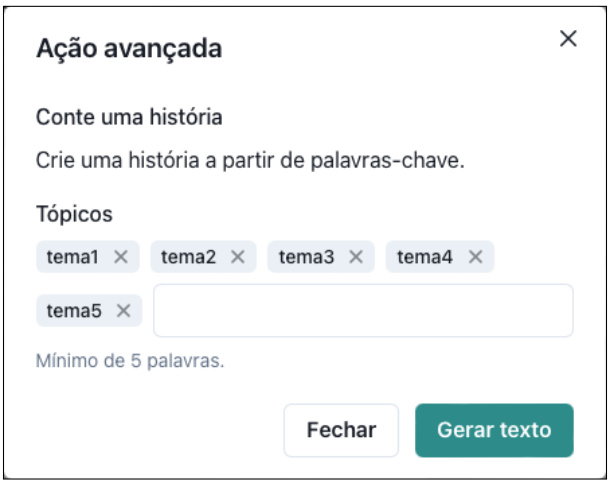

Figura 5.6 Ação avançada "Conte uma história".

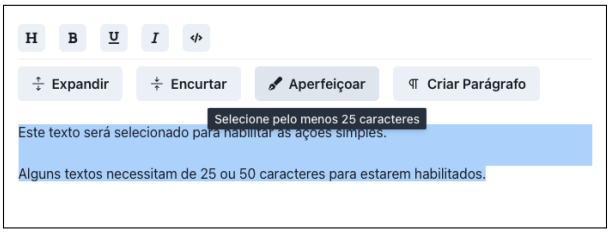

Figura 5.7 Botões das Ações Rápidas habilitados e o *tooltip* ao passar o mouse pela ação "Aperfeiçoar".

clusão, com o texto escrito no formato markdown. Desta forma, o GPT-3 tem uma saída mais constante e uniformizada, ao invés de dizermos "gere um blog post com o tema X", onde o resultado poderia ser de estruturas diferentes a cada chamada.

Para padronizar a chamada da API no backend, o formato escolhido foi de adicionar os *inputs* do usuário, aos prompts testados para interação com o GPT-3. Podemos ver como esse tratamento é feito no listing 5.2. O frontend envia uma requisição a função lambda, que modifica o prompt a depender do caso de uso, e ao final envia a requisição para a OpenAI API.

```
let max_tokens = 1200;
switch (completionEvent) {
  case "EXPASION":
    prompt = `expandir a frase "${prompt}"`;
    break;
  case "IMPROVE":
    prompt = `melhorar a frase "${prompt}"`;
    break;
  case "PARAGRAPH":
    prompt = `criar um paragrafo para "${prompt}"`;
    break;
  case "SHORTEN":
    prompt = `diminuir a frase "${prompt}"`;
    break;
  case "BLOG_POST":
```

Figura 5.8 Aba de configurações.

```
16      prompt = `crie um blog post em formato markdown com 3 seções e 1
            conclusão, com o tema "${prompt}"`;
17      break;
18    case "LIST":
19      prompt = `faça uma lista de cinco itens sobre ${prompt}`;
20      break;
21    case "PRODUCT_DESCRIPTION":
22      const { name, description } = prompt;
23      prompt = `criar uma descrição do produto ${name} com a descrição mais
            detalhada a partir de "${description}"`;
24      break;
25    case "STORY":
26      const { tags } = prompt;
27      prompt = `crie uma história a partir das palavras-chave "${tags.join()}
            "`;
28      break;
29    case "EXPLAIN_TO_CHILD":
30      prompt = `simplifique esse texto para que uma criança de 5 anos possa
            entender "${prompt}". use palavras simples.`;
31      break;
32    default:
33      break;
34  }
35  const completion = await openai.createCompletion("text-davinci-001", {
36    prompt,
37    max_tokens
38  });
39
40  return {
41      statusCode: 200,
42      body: completion.data.choices[0].text,
43  };
```

Listing 5.2 Exemplo de prompts tratados no backend em Node.js. As requisições são enviadas a OpenAI e a resposta retornada ao frontend.

5.9 Pesquisa Qualitativa e Quantitativa

Para análise da experiência dos usuários, fizemos uma rodada de entrevistas com o objetivo de extrair informações sobre o suporte à criatividade do nosso experimento. Primeiramente, os usuários participaram de uma entrevista síncrona, com acesso a plataforma. Ao final, os entrevistados responderam dois formulários com as perguntas do SUS e UEQ.

A análise quantitativa da experiência se deu através das respostas desses formulários. A análise qualitativa se deu através da nossa percepção de uso da plataforma pelos entrevistados, respostas das perguntas do roteiro e feedbacks relatados durante o uso da plataforma e nos formulários.

5.10 Roteiro da Entrevista

O roteiro da entrevista seguiu o modelo de entrevista não-estruturada com um teste de usabilidade. As entrevistas não-estruturadas se parecem com uma conversa, mas sempre tem o mesmo objetivo final. O entrevistador tem um roteiro das perguntas a serem feitas, mas pode perguntá-las em diferentes ordens a depender do caminho da conversa. O entrevistado tem espaço para dar seus *feedbacks* e opiniões, e o entrevistador vai acompanhando e entendendo quais as próximas perguntas e passoas para extrair as informações pertinentes para a pesquisa. Quanto ao teste de usabilidade, o usuário tem objetivos explícitos a serem realizados dentro de um artefato e o pesquisador acompanha, sem intervenções, a interação usuário-artefato. Ao final, perguntas abertas são feitas para avaliar a experiência.

Neste trabalho, o roteiro da entrevista foi criado no formato de um teste de usabilidade e após a entrevista não-estruturada. Num conjunto de 30 minutos, os usuários são expostos a diferentes atividades, estando com acesso a plataforma. Depois disso, algumas perguntas são feitas, e é dado o espaço para os usuários compartilharem suas opiniões. Ao final, os entrevistados respondem os dois formulários relativos ao SUS e UEQ. Podemos verifica o roteiro da entrevista abaixo:

1. Apresentação da entrevista e pedido para compartilhar tela, se o usuário se sentir confortável. Os entrevistados são também avisados que podem falar sempre que quiserem durante a entrevista e experiência.

2. Em um intervalo de 15 minutos, o usuário interage com a plataforma. Esse tempo pode se estender em algumas etapas, e ser mais curta em outras, a depender do usuário.

   (a) 3 minutos: Explorar a plataforma livremente.

   (b) 5 minutos: Escrever um texto para um blog. Esse blog post será o primeiro post de um blog ficcional do entrevistado. O tema é de escolha livre.

   (c) 5 minutos: Utilizar pelo menos uma Ação Avançada.

   (d) 2 minutos: Utilizar pelo menos uma Ação Rápida.

3. Ao final da experimentação, realizamos cinco perguntas para avaliar a experiência com o artefato:

(a) Para você, o que deveria ser eliminado do artefato (tem hoje, mas deveria não ter mais)?

(b) Para você, o que deveria ser reduzido do artefato (tem hoje, mas deveria ter menos)?

(c) Para você, o que deveria ser mantido ao artefato (tem e deveria continuar do jeito que está)?

(d) Para você, o que deveria ser ampliado do artefato (tem hoje, mas deveria ter mais)?

(e) Para você, o que deveria ser adicionado ao artefato (não tem hoje, mas deveria passar a ter)?

Essas perguntas foram adaptadas do processo do Blue Ocean.

4. Compartilhamos o link para os dois formulários de avaliação, do SUS e UEQ.

5.11 Entrevistas e Análise

As entrevistas foram feitas num intervalo de uma semana. Após as entrevistas, fizemos uma análises sobre as gravações e anotações feitas durante as entrevistas.

Cada um dos entrevistados já teve experiências criando textos para internet, seja conteúdo textual ou audiovisual. A avaliação qualitativa desses candidatos é importante para entender a percepção com a plataforma e o sentimento enquanto a utiliza.

5.11.1 Processo de Entrevistas Individuais e Análises

Cada entrevista foi feita individualmente utilizando a plataforma Google Meets. Se o entrevistado se sentisse confortável, a gravação da utilização da plataforma seria feita e o compartilhamento de tela também.

Durante a análise, revisitamos as gravações das entrevistas e anotações feitas durante o processo. Temas e dificuldades semelhantes apareceram em diferentes entrevistas, por isso agrupamos os *feedbacks* e opiniões de uso para identificar os pontos positivos e negativos da plataforma.

5.11.1.1 Perfil dos participantes

Entrevistamos cinco usuários, os seus perfis estão descritos na tabela 5.2. Todos os participantes foram avisados da confidencialidade da sua contribuição e os que consentiram, a gravação da entrevista foi realizada. O tempo destinado para a interação da plataforma era de 15 minutos, mas a maioria ultrapassou esse tempo. Seja porque o entrevistado estava entrando numa área mais tortuosa e queríamos entender até que ponto ele chegaria ou o interesse do próprio entrevistado em continuar utilizando a plataforma.

5.11.1.2 Agrupamento

Mesmo as entrevista tendo diferentes caminhos, alguns temas foram recorrentes. Durante a análise, algumas categorias de *feedback* e percepções se destacaram. Usabilidade, entendi-

	P1	P2	P3	P4	P5
Formação	Engenharia da Computação	Engenharia da Computação	Ciência da Computação	Engenharia da Computação	Engenharia da Computação
Ocupação	Desenvolvedor	Desenvolvedor	Desenvolvedor	Desenvolvedor	Desenvolvedor
Nacionalidade	Brasileiro	Brasileiro	Brasileiro	Brasileiro	Brasileiro
Duração Interação plataforma	n/a	17:35	23:36	22:47	23:31

Tabela 5.2 Perfil dos entrevistados.

mento das ações, ações mais usadas e ações não usadas foram alguns desses grupos.

Além das entrevistas, os formulários do SUS e UEQ tinham campos com espaço para comentários e sugestões, dos quais alguns entrevistados preencheram.

CAPÍTULO 6

Resultados e Discussão

Este capítulo tem o objetivo de compartilhar os resultados quantitativos e qualitativos das avaliações individuais. O resultado do SUS e UEQ foram sumarizados através das respostas dos entrevistados via formulários. Através dos agrupamentos dos *feedbacks* e análises das entrevistas, chegamos aos resultados qualitativos. Além do compartilhamento dos resultados, as respostas e valores encontrados são discutidos ao longo do texto.

6.1 Resultado das pesquisas individuais

Para cada um dos modelos de avaliação da experiência do usuário, aplicamos as fórmulas definidas pelos modelos para o cálculo de suas métricas.

6.1.1 SUS

Na tabela 6.1, podemos ver as respostas para as 10 perguntas do SUS. As perguntas se encontram abaixo e para cada uma das perguntas o usuário responde de 1 a 5, sendo 1 "Discordo totalmente"e 5 "Concordo totalmente". O SUS é um modelo mais rápido de ser respondido e com uma única métrica para comparação. O resultado visto na tabela é de 82.5 pontos, o que é um bom resultado para a usabilidade, já que a média da escala é de 68 pontos.

- Q1: Eu acho que gostaria de usar este sistema frequentemente.
- Q2: Eu achei o sistema desnecessariamente complexo.
- Q3: Eu achei que este sistema foi fácil de usar.
- Q4: Eu acho que eu precisaria do suporte de uma pessoa com conhecimentos técnicos para ser capaz de usar este sistema.
- Q5: Eu achei que várias funções neste sistema estão bem integradas.
- Q6: Eu achei que existiram várias inconsistências neste sistema.
- Q7: Eu imagino que a maioria das pessoas aprenderia a usar este sistema rapidamente.
- Q8: Achei o sistema muito complicado de usar.
- Q9: Me senti confiante usando o sistema.

- Q10: Eu precisaria aprender muitas coisas antes que eu pudesse me sentir a vontade em usar este sistema.

E.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score SUS	Score Final SUS
P1	3	2	5	1	4	3	5	1	4	1	33	82.5
P2	5	1	4	1	4	1	4	1	4	1	36	90
P3	3	1	5	1	5	2	4	1	5	1	36	90
P4	1	1	5	1	1	5	5	1	4	1	27	67.5
P5	4	2	4	1	3	2	4	3	4	2	29	72.5
Média											33	82.5

Tabela 6.1 Respostas do entrevistados(E.) do formulário SUS.

6.1.2 UEQ

Utilizando a ferramenta de benchmark [SHT17] para sumarizar a pesquisa, temos o resultado das 26 perguntas feitas para os entrevistados. Para cada área de análise do UEQ, temos 4 perguntas relativas a área, isso é feito para que o resultado daquela área seja a média das 4 perguntas feitas. Na tabela 6.2 vemos as respostas de cada um dos entrevistados. A média, variância e desvio padrão de cada um dos itens das perguntas podem ser vistos na tabela 6.3. O agregado das escalas pode ser visto na tabela 6.4. Além disso, a média das qualidades pragmáticas e hedônicas podem ser vistas em 6.5.

Na prática, os valores observados do UEQ [SHT17] ficam entre -2 e 2, sendo os valores acima de 0.8 uma avaliação positiva e abaixo de -0.8 uma avaliação negativa. Os valores entre -0.8 e 0.8 são neutros. Podemos ver na figura 6.1 o comparativo dos valores das escalas e o nível de cada uma.

Dos resultados das entrevistas, a Atratividade foi a escala com a maior média, de 1.5, enquanto Inovação teve a menor média, mas a maior variância. Com relação às qualidades, a qualidade hedônica, relacionada a experiência do usuário, se mostrou um pouco acima do nível de neutralidade, para o lado positivo com a média de 0.85. Enquanto a qualidade pragmática, com o foco no objetivo final, ficou dentro da neutralidade com média de 0.53.

	Itens																								
E.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
P1	6	5	1	1	2	7	7	3	2	1	5	1	6	7	7	7	4	2	1	7	3	6	1	3	3	7
P2	7	7	7	7	7	7	7	2	6	7	4	6	6	6	6	6	6	6	6	6	4	6	1	1	4	7
P3	6	6	2	1	3	5	6	6	6	2	5	2	6	5	5	5	4	5	6	2	3	6	6	3	2	6
P4	3	4	3	6	1	4	6	1	1	6	3	2	6	1	1	2	1	6	7	1	7	2	1	1	1	5
P5	6	5	7	4	7	5	5	4	2	2	6	1	4	6	4	6	2	2	1	1	3	6	2	2	2	6

Tabela 6.2 Respostas dos entrevistados(E.) do formulário UEQ. Cada um dos itens de 1 a 26 se relacionam diretamente aos itens da tabela 6.3.

Item	Média	Variância	Desvio Padrão	No.	Esquerda	Direita	Escala
1	1.6	2.3	1.5	5	Desagradável	Agradável	Atratividade
2	1.4	1.3	1.1	5	Incompreensível	Compreensível	Clareza
3	0.0	8.0	2.8	5	Criativo	Sem criatividade	Inovação
4	0.2	7.7	2.8	5	De Fácil aprendizagem	De difícil aprendizagem	Clareza
5	0.0	8.0	2.8	5	Valioso	Sem valor	Estimulação
6	1.6	1.8	1.3	5	Aborrecido	Excitante	Estimulação
7	2.2	0.7	0.8	5	Desinteressante	Interessante	Estimulação
8	-0.8	3.7	1.9	5	Imprevisível	Previsível	Confiabilidade
9	0.6	5.8	2.4	5	Rápido	Lento	Eficiência
10	0.4	7.3	2.7	5	Original	Convencional	Inovação
11	0.6	1.3	1.1	5	Obstrutivo	Condutor	Confiabilidade
12	1.6	4.3	2.1	5	Bom	Mau	Atratividade
13	1.6	0.8	0.9	5	Complicado	Fácil	Clareza
14	1.0	5.5	2.3	5	Desinteressante	Atrativo	Atratividade
15	0.6	5.3	2.3	5	Comum	Vanguardista	Inovação
16	1.2	3.7	1.9	5	Incómodo	Cómodo	Atratividade
17	0.6	3.8	1.9	5	Seguro	Inseguro	Confiabilidade
18	-0.2	4.2	2.0	5	Motivante	Desmotivante	Estimulação
19	-0.2	8.7	2.9	5	Atende as expectativas	Não atende as expectativas	Confiabilidade
20	-0.6	8.3	2.9	5	Ineficiente	Eficiente	Eficiência
21	0.0	3.0	1.7	5	Evidente	Confuso	Clareza
22	1.2	3.2	1.8	5	Impraticável	Prático	Eficiência
23	1.8	4.7	2.2	5	Organizado	Desorganizado	Eficiência
24	2.0	1.0	1.0	5	Atraente	Feio	Atratividade
25	1.6	1.3	1.1	5	Simpático	Antipático	Atratividade
26	2.2	0.7	0.8	5	Conservador	Inovador	Inovação

Tabela 6.3 Média, Variância e Desvio Padrão para cada uma das perguntas feitas no questionário UEQ.

Escalas UEQ (Média e Variância)		
Atratividade	1.500	0.57
Clareza	0.800	1.23
Eficiência	0.750	1.66
Confiabilidade	0.050	1.73
Estimulação	0.900	0.86
Inovação	0.800	2.39

Tabela 6.4 Média e Variância das Escalas UEQ

Qualidades Pragmáticas e Hedônicas	
Atratividade	1.50
Qualidade Pragmática	0.53
Qualidade Hedônica	0.85

Tabela 6.5 Média da Atratividade, Qualidades Pragmáticas e Qualidades Hedônicas

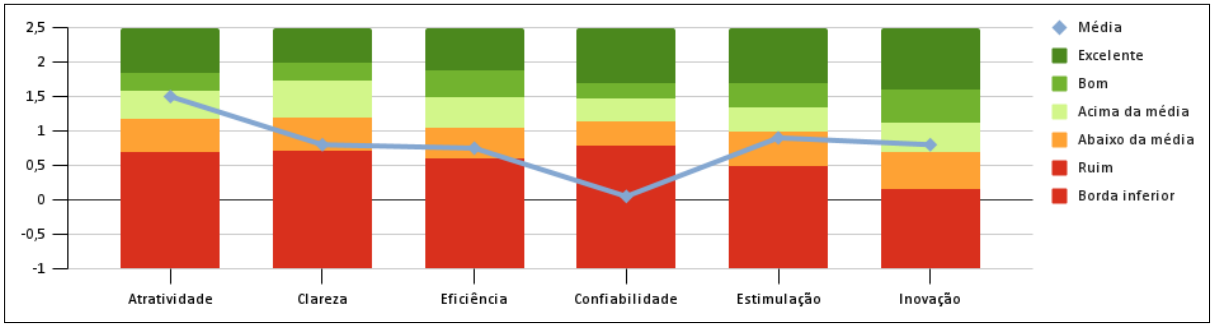

Figura 6.1 Aba de configurações.

6.1.3 Agrupamento

Da análise das entrevistas, dividimos os grupos de *feedback* em Usabilidade, Resultado das Ações e Features não utilizadas.

6.1.3.1 Usabilidade

A experiência do usuário está atrelada ao artefato e as interações possíveis com ele. O resultado das ações e o conteúdo são muito importantes, mas algumas erros básicos de usabilidade podem acabar influenciando negativamente no resto da experiência. Os pontos de melhoria e problemas identificados na a plataforma se encontram abaixo:

- Dentro de outra plataforma: por estar dentro do meu blog pessoal, algumas funções acabaram levando usuários para um caminho fora do artefato principal. Por exemplo, alguns entrevistados clicaram no botão "Roleta da Serendipidade"e não entenderam para onde estavam sendo redirecionados.

- Falta de feedbacks de confirmação: o campo de *input* da chave de API pública em configurações não retornava uma confirmação após ser adicionado, enquanto que outros campos tinham esse retorno. A não padronização entre elementos do mesmo tipo dentro da plataforma confundiu os usuários.

- Estilização do texto: editores de texto existem na internet desde o seu início, por isso, os entrevistados já estavam acostumados com atalhos e ferramentas que existem nos mais diversos tipos de editores de texto. O fluxo de estilização, por não seguir o padrão de outros editores, acabou não sendo tão direto. Aqui existe uma oportunidade de se adequar aos padrões de atalhos e *features* de outros editores.

- Onboarding: ao entrar na plataforma alguns entrevistados se sentiram perdidos. Com a quantidade de ações e possibilidades na plataforma, o tempo de exploração foi necessário para que pudessem entender como que ela funcionava. Além disso, a única descrição das Ações Rápidas é o seu próprio nome, então não fica muito claro qual o resultado final dessas ações, em comparação com as Ações Avançadas.

6.1.3.2 Resultado das Ações

Durante o objetivo de criar o seu primeiro blog, o entrevistado P3 submeteu o prompt "Detecção de Fake News através de processamento natural de linguagem."na ação avançada "Criar um Blog Post". A ação retornou o esperado texto estruturado de blog post no tema pedido, mas o que chamou a atenção do entrevistado foi uma frase dentro da introdução do texto: "Segundo o relatório do Pew Research, 74% dos americanos acreditam que existem notícias falsas na imprensa.".

Essa frase conta com uma citação a uma pesquisa, mas não existe uma fonte. O usuário abriu um aba no seu navegador e foi procurar esse relatório. O usuário fez a pesquisa em português e inglês, mas não a encontrou.

O exemplo acima é bem parecido com o que relatamos na seção 2.3.4. Os dados gerados pelo GPT-3 podem ser até verdadeiros, mas sem a fonte da informação fica difícil acreditar, principalmente se você já tiver visualizado outros momentos em que as ações não eram tão confiáveis. O que lembra outro ponto importante: conclusões lógicas. Em alguns momentos a estrutura de parágrafos seguia o formato, se "A"e "B", então "C". Entretanto, não existia uma conclusão lógica para "C", mas gramaticalmente e sintaticamente a frase estava correta.

Além disso, os resultados das ações do GPT-3 são não-determinísticos, ou seja, dada uma certa entrada ele pode dar diferentes resultados para ela. O que pode atrapalhar ou ajudar na co-criação de textos. Em alguns momentos, a ferramenta ajudou, pois criou diferentes vertentes para um mesmo prompt. Em outros momentos, ela atrapalhou por não conseguir reproduzir o mesmo output para um pedido anterior.

Uma outra dificuldade foi a de respostas de ações, mas que não seguiam a descrição delas. A depender do prompt, algumas ações não retornavam o que estava sendo pedido. Em alguns momentos, o texto era retornado em inglês, e em outros nada mudou, como no uso de Ações rápidas.

As ações de "Criar um Blog Post", "Listar Tópicos", "Resumir para uma criança"e "Conte uma história"foram as ações que tiveram os melhores resultados e que os entrevistados ficaram mais animados com.

Por último, uma dificuldade conjunta de usabilidade e resultado das ações foi a da inserção do resultado das ações no texto. Quando uma ação é executada, o texto resultante é inserido no local onde o cursor estava antes de pedir a execução da ação. Mas, quando o texto do participante começa a ficar grande, a dificuldade de identificar onde o texto da ação foi adicionado aumenta. Um pedido de diversos participantes foi de identificar quais eram as mudanças após a execução das ações.

6.1.3.3 Features não utilizadas

A aba de documentos foi praticamente inutilizada. O único momento em que essa aba foi utilizada, foi durante o momento de exploração da plataforma, e mesmo assim, o seu uso ainda não estava 100% compreensível. Outras *features* que não foram muito utilizadas foram as Ações Rápidas, e mesmo quando utilizadas, os seus resultados não foram tão expressivos. Os usuários acabavam gastando uma quantidade considerável de tempo para identificar o que foi modificado no texto após a execução da ação.

Além da aba de documentos, o texto de base não foi utilizado. Talvez uma melhor abordagem para demonstração da plataforma seja um tutorial *step-by-step*, como o tempo que foi utilizado para fazer a exploração durante a entrevista. Como também, a adição de uma *feature* para destacar as modificações propostas pelo GPT-3 ao adicionar no texto.

CAPÍTULO 7

Conclusão

Através das avaliações realizadas, percebemos que a usabilidade do sistema encontra-se com valores positivos para os usuários, mesmo que existam oportunidades para melhoria. Numa experiência de co-criação com o GPT-3, não somente a interação com o GPT-3 é importante, mas todo o sentimento e experiência utilizando o artefato.

O GPT-3 é uma ferramenta muito poderosa e consegue resolver diversos problemas. Porém, ainda existem muitos casos em que as suas respostas não necessariamente são coerentes, corretas e baseadas em fatos. Se houvesse a adição de referências ou links nas informações compartilhadas pelo GPT-3, o grau de confiança na plataforma e nos resultados poderia aumentar.

Para os entrevistados, a ferramenta Serenpit Editor e suas *features* se mostraram auxiliar no suporte a criatividade. As Ações Avançadas desempenharam um papel fundamental no início e desenvolvimento de seus textos. Por ser não-determinística, o GPT-3 cria a possibilidade de novos caminhos e dá chance ao acaso. A combinação do não-determinístico com resolução geral de problemas, faz o GPT-3 ser uma ótima parceira de co-criação textual.

A plataforma de desenvolvimento teve como seu escopo o mínimo necessário para realizar o experimento, e mesmo assim algumas *features* não foram utilizadas durante as avaliações. Isso mostra a importância dos testes e as conversas com usuários, para identificar o que de fato é o mínimo necessário para a plataforma.

A formalização das avaliações com dados quantitativos do SUS e UEQ ajudam a dar valor ao que os usuários compartilharam qualitativamente. A adição de uma entrevista semi-estruturada com um teste de usabilidade trouxe resultados importantes para identificar pontos fortes e fracos da plataforma.

Mesmo que os conteúdos gerados pelo GPT-3 sejam parecidos com linguagem natural, algumas de suas conclusões e construções do pensamento nos fazem lembrar que aquele texto não foi gerado por um humano. O modelo não resolve todos os problemas, mas consegue dar um ótimo suporte no processo de co-criação. Além disso, existe uma grande oportunidade para implementar soluções que avaliem a escrita e proponha mudanças através do GPT-3.

Para possíveis trabalhos futuros existem diversas oportunidades a serem desenvolvidas: Comparação entre os modelos de linguagem natural, GPT-3, PaLM e até o próximo grupo de modelos na casa de trilhões de parâmetros; Identificar quais são os casos de uso mais importantes para a co-criação de texto online; e a co-criação de outros tipos de conteúdo para internet, como imagens, áudio e vídeo.

Referências Bibliográficas

[Agr21] Lucas Agrela. Estes são os dez aplicativos mais baixados de 2021, 7 2021.

[AK21] Ali Alvi and Paresh Kharya. Using deepspeed and megatron to train megatron-turing nlg 530b, the world's largest and most powerful generative language model, Oct 2021.

[Atr22] Amanda Atrash. The blue ocean strategy summary (with 4 examples), Feb 2022.

[aws] Cloud computing services - amazon web services (aws).

[BBH+22] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022.

[BGW+21] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. If you use this software, please cite it using these metadata.

[BMR+20] 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.

[cdn] What is a cdn? how do cdns work? | cloudflare.

[cha] Chakra ui - a simple, modular and accessible component library that gives you the building blocks you need to build your react applications.

[com] Common crawl.

[Coo22] Kindra Cooper. Openai gpt-3: Everything you need to know, Feb 2022.

[dra]

[GBB+20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[GBB+21] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2021.

[gcp] Google cloud platform - solve your toughest challenges with google cloud.

[GPAM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[gra] Write your best with grammarly.

[Ho19] George Ho. Autoregressive models in deep learning - a brief survey, Mar 2019.

[jasa] Jasper (formerly jarvis) - 1 ai writing assistant.

[jasb] Plans amp; pricing - jasper.

[Joh11] Steven Johnson. *DE ONDE VÊM AS BOAS IDEIAS*. Editora Zahar, 2011.

[KM14] W. Chan Kim and Renée Mauborgne. Charting your company's future, Aug 2014.

[LCLdM05] Pedro Lincoln C. L. de Mattos. A entrevista não-estruturada como forma de conversação: razões e sugestões para sua análise. *Revista de Administração Pública - RAP*, 2005.

[Lew18a] James R. Lewis. The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, 34:577 – 590, 2018.

[Lew18b] James R. Lewis. The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590, 2018.

[LHS08] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. *Lecture Notes in Computer Science*, page 63–76, 2008.

[lin] Linguix free writing assistant.

[LMY+21] Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, and et al. M6: A chinese multimodal pretrainer, May 2021.

[McD19] Scott McDowell. The "yes, and..."approach: Less ego, more openness, more possibility, Feb 2019.

[McG19] Mark McGuinness. 7 types of creative block (and what to do about them), Feb 2019.

[MD20] Gary Marcus and Ernest Davis. Gpt-3, bloviator: Openai's language generator has no idea what it's talking about, Dec 2020.

[mit22] Mit license, Apr 2022.

[NC22] Sharan Narang and Aakanksha Chowdhery. Pathways language model (palm): Scaling to 540 billion parameters for breakthrough performance, Apr 2022.

[net] Develop amp; deploy the best web experiences in record time.

[Nod] Node.js.

[OD20] Liz O'Sullivan and John P. Dickerson. Here are a few ways gpt-3 can go wrong, Aug 2020.

[Ope21] OpenAI. Openai api, Nov 2021.

[Ram21] Nidhi Raman. How content writing is similar to script writing, Jun 2021.

[rea] React – uma biblioteca javascript para criar interfaces de usuário.

[rik] The vault for your a.i. creations.

[RM21] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[Ryta] Rytr. About rytr. `https://rytr.me/about`.

[Rytb] Rytr. A better, 10x faster way to write emails.

[SABZ17] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, 5:3909–3943, 2017.

[She20] Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404:132306, mar 2020.

[SHT17] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. Construction of a benchmark for the user experience questionnaire (ueq). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40, 2017.

[SMM+17] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017.

[sta21] MasterClass staff. What is writer's block? how to overcome writer's block with step-by-step guide and writing exercises (with video), Aug 2021.

[Ver22] Christian Versloot, Feb 2022.

[Wan21] Ben Wang. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[Wri] Writesonic. Ai writer, ai copywriter amp; writing assistant.

[wri22] Writer documentation, Apr 2022.

[ZDZ] Susan Zhang, Mona Diab, and Luke Zettlemoyer. Democratizing access to large-scale language models with opt-175b.

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).