Relatório de Prospecção

GT Sokrates.ai

Plataforma Socrática de Conhecimento

Geraldo Xexéo

9/7/2024

1. Estado da Arte

A principal tecnologia envolvida no desenvolvimento do Sokrates.ai, e que possibilita toda a sua capacidade de inovação, é conhecida como LLM, para *Large Language Models*. Além disso, outras tecnologias serão necessárias para a implementação, porém não trazem especificamente nenhum desafio, como linguagens de programação e de interface (Python, HTML5) e bancos de dados.

Associada à tecnologia de LLM estão algumas técnicas que também serão discutidas como RAG, Prompt Engineering e Agentes Múltiplos.

Ressalta-se que o estado da arte é altamente influenciado pela indústria, não sendo restrito à academia, e que nesse relatório as referências do tipo URL serão feitas por meio de *footnotes*.

1.1. O que são LLM

Large Language Models são modelos de aprendizado de máquina treinados em grandes quantidades de texto para compreender e gerar linguagem natural. Eles são baseados na tecnologia de Transformers, que utiliza mecanismos de atenção para processar e gerar texto de forma eficiente e contextualizada (Vasmani et al., 2017). De maneira mais geral, são formadas por um grande algoritmo determinístico, basicamente descrito na forma de um modelo com milhões de parâmetros, que, dada uma sequência, é capaz de gerar uma distribuição de probabilidade para a sequência a seguir, sendo uma evolução dos modelos seq2seq (Sutskever et al. 2014). Um algoritmo baseado em técnicas de aleatoriedade, então, escolhe uma cadeia, em função de parâmetros como temperatura, entre outros. Uma das melhores explicações do seu funcionamento é feita por Wolfram[1].

A utilização desses modelos são essenciais para a criação de agentes conversacionais, uma vez que interagem de forma natural e eficaz com o usuário, de maneira a promover uma melhor experiência de aprendizado (Bender et al., 2021; Brown et al., 2020).

Esses modelos são baseados em algumas ideias como Transferência de Aprendizado, estruturas de codificação e decodificação, vetores de contexto e atenção (Bahdanau et al., 2015) e no forte uso de operações de vetores por meio de GPUs.

Em especial, a atual geração de LLMs é baseada na arquitetura de *transformers*. Um transformer é uma arquitetura de rede neural introduzida por Vaswani et al. (2017), que se distingue por seu uso de mecanismos de atenção para melhorar a eficiência e a eficácia do processamento de sequências de texto. Em vez de processar as palavras de forma sequencial, como nas RNNs (Redes Neurais Recorrentes), os *transformers* utilizam mecanismos de auto-atenção para relacionar todas as palavras de uma sequência entre si

[1] https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

simultaneamente. Isso permite que o modelo capture dependências a longa distância com maior facilidade.

Um LLM é, na prática, uma rede neural profunda, treinada com uma grande quantidade de textos em tarefas genéricas, como previsão do próximo token, ou previsão de um token aleatório em uma sequência, e também, possivelmente, treinada ou refinada com tarefas específicas, como tarefas de classificação ou tradução. A partir de modelos iniciais como os de Vasnani et al. (2017) e Devlin et al. (2019), redes cada vez maiores foram sendo criadas.

O estado da arte é atualmente considerado estar entre duas LLMs, a ChatGPT 4o, da Open AI[2], e a Claude 3[3], da Anthropic (ainda não disponível no Brasil). Pouco se sabe sobre a estrutura interna do ChatGPT 4 ou Claude 3.5, os maiores modelos e com melhor desempenho, na data deste relatório. Essa posição de líder do mercado, porém, sofre constante desafio, tanto por parte de modelos abertos como Llama, da Meta (ex-Facebook), como por modelos fechados como o Gemini da Google.

1.2. Como são usadas

As LLMs são capazes de realizar uma variedade de tarefas de linguagem natural, como tradução para diferentes idiomas, resumo de textos, respostas a perguntas, assistentes virtuais, sistemas de recomendação, análise de sentimentos, tendo sido usadas como base para softwares amplamente utilizados como o ChatGPT, Claude 3.4 e o Gemini (Rane et al., 2024).

Atualmente empresas utilizam modelos mais simples para a criação de chatbots para atendimento de clientes. Esses chatbots são úteis em diversas tarefas, desde o esclarecimento de dúvidas até a realização de processos de garantia ou devolução, sabendo lidar com as especificidades da empresa e podendo ser disponibilizados e utilizados ininterruptamente por 24 horas (Nascimento, 2023).

O treinamento de modelos para LLMs é uma tarefa de custo alto, exigindo máquinas com GPUs de grande capacidade de memória. Muitos treinamentos têm sido realizados em máquinas especiais, produzidas especialmente pela NVidia. Na grande maioria das aplicações são utilizados modelos pré-treinados, abertos ou não. Esses modelos podem ser refinados para uma tarefa específica. No entanto, devido à grande generalidade dos modelos pré-treinados, o refinamento nem sempre leva a bons resultados (Devlin et al., 2018). Além disso, no caso de LLMs fechadas, esse refinamento pode ser complexo ou viável apenas a um custo elevado.

Três são as formas básicas de usar uma LLM já treinada dentro de uma aplicação: a primeira é em máquina própria, a segunda em uma máquina na nuvem, como no serviço SageMaker, paga pelo tempo que está ligada, e a terceira por meio de chamadas de API, que normalmente cobram pela quantidade de tokens na entrada e na saída. De todas as formas, os modelos são normalmente chamados por meio de APIs e de um comando em forma de

[2] https://chatgpt.com/
[3] https://claude.ai/

texto, que é a própria sequência de entrada, conhecido como *prompt.* O prompt pode ser estruturado em várias partes, incluindo exemplos, comandos, contexto e papel, dependendo da aplicação e da API utilizada. Exemplos fornecem amostras de entradas e saídas esperadas, comandos especificam as ações a serem realizadas, o contexto fornece informações adicionais relevantes para a tarefa, e o papel define o comportamento ou a perspectiva que o modelo deve adotar, como, por exemplo, pedindo para que "aja como" determinado especialista.

Para o uso eficaz e eficiente de LLMs, porém, várias outras técnicas devem ser utilizadas. Algumas, são de baixa complexidade técnica, como leitura de arquivos com formatos específicos. Outras, porém, estão em pleno desenvolvimento com novidades semanais.

1.3. Uso de RAG

RAG, ou "Retrieval-Augmented Generation" (Geração Aumentada por Recuperação), é uma técnica que combina a geração de texto baseada em modelos de linguagem com a recuperação de informações relevantes de uma base de dados ou de um corpus de documentos (Lewis et al. 2020; Hu & Lu, 2024).

Esta abordagem permite que os modelos de linguagem gerem respostas mais precisas e contextualizadas, em relação a um conjunto específico de documentos, especialmente em cenários onde a informação necessária pode não estar completamente coberta pelos dados de treinamento do modelo.

No contexto de RAG, o processo genérico envolve três etapas principais (Lewis et al. 2020):

1. Indexação: Nesta fase, um conjunto de documentos é dividido em trechos que são indexados usando diferentes técnicas provenientes da área de Recuperação da Informação e Aprendizado de Máquina. Dois modelos comuns são o uso do modelo vetorial ou o uso de embeddings.
2. Recuperação: Nesta fase, um sistema de recuperação de informações (geralmente um modelo de busca ou um indexador) é utilizado para buscar documentos ou trechos de texto que são relevantes para a consulta ou o prompt fornecido pelo usuário. Esses documentos recuperados servem como uma base de conhecimento adicional para o modelo de geração de texto.
3. Geração: A partir da recuperação das informações relevantes, é construído um contexto que é usado, junto com outras informações, na forma de um *prompt* estendido, para o modelo de geração de texto, no caso uma LLM. Assim, essas informações são usadas para produzir uma resposta ou um texto que é mais preciso e informativo. A presença dos documentos recuperados ajuda a ancorar a geração do texto em fatos e dados específicos, aumentando a relevância e a precisão da resposta gerada. Isso é muitas vezes também chamado de aprendizado no contexto, mas não é realmente um processo de aprendizado.

Nos últimos meses foram apresentadas na literatura várias formas de realizar RAG: com técnicas tradicionais de Recuperação da Informação, com técnicas de *embedding* e mesmo com o uso de LLMs no processo (Costa & Karypis, 2024; Hu & Lu, 2024). Em fevereiro de 2024, a Microsoft lançou um modelo acompanhado de código chamado GraphRAG[4] que pode ser considerado o estado da arte no tema.

O uso de RAG é particularmente vantajoso em aplicações onde a precisão da informação é crucial, como em assistentes virtuais, sistemas de perguntas e respostas, e plataformas de aprendizado, como o Sokrates.AI. Ao integrar a recuperação de documentos com a geração de texto, o Sokrates.AI pode fazer perguntas, gerar contra-exemplos e fornecer informações mais consistentes e detalhadas, contextualmente relevantes e baseadas em dados atualizados, melhorando assim a experiência de aprendizado dos usuários.

1.4. Uso de Prompt Engineering

Os agentes conversacionais dependem de prompts bem formulados para interagir de forma eficaz com os usuários. No contexto do Sokrates.AI, a engenharia de prompts será composta das etapas de criação, teste e refinamento, uma vez que os mesmos guiam as interações entre os agentes e os usuários, assegurando que as respostas e interações geradas sejam relevantes, com linguagem acessível e permitam uma comunicação eficiente e eficaz com os usuários, permitindo assim, que sejam úteis para o processo de aprendizagem e o pensamento socrático (Thanasi-Boçe & Hoxha, 2024).

Os prompts serão elaborados de forma a estimular o pensamento crítico e a reflexão, características do pensamento socrático. Além de perguntas base, também serão realizados questionamentos sobre as respostas do usuário, a fim de incentivar a análise, a reflexão e motivar a autoavaliação do pensamento (Misnevs, 2023).

A engenharia de prompts também envolve a personalização das interações com base no perfil e nas respostas anteriores do usuário. Isso garante que os agentes forneçam feedbacks adaptativos e contextualmente relevantes, aprimorando a experiência de aprendizagem. A utilização de técnicas de processamento de linguagem natural (NLP) permite que os agentes compreendam e respondam adequadamente às variações na linguagem dos usuários, tornando as interações mais naturais e efetivas (Marvin et al., 2023).

A engenharia de prompts também visa economizar no tamanho do prompt, de forma a diminuir o custo por chamada nas APIs pagas por token. O que acontece é um equilíbrio entre custo e desempenho, e sua influência no negócio (Xexéo et al. 2024)

Finalmente, mais recentemente, estudos mostram que prompts longos, que são chamados também de contextos longos, não são tratados igualmente em toda sua extensão (Liu et. al. 2023).

---

[4] https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/

Esta também é uma área em constante inovação, como por exemplo o uso de aprendizado de máquina na criação do prompt (Su et al. 2022) e na compactação do prompt[5].

1.5. Uso de Cadeias de Processamento

As cadeias de processamento e os agentes múltiplos são técnicas avançadas que permitem a execução de tarefas complexas por meio da coordenação de múltiplos agentes de software. Esses agentes podem realizar uma série de atividades, desde a recuperação de informações até a geração de respostas contextualmente relevantes, melhorando significativamente a eficácia e a eficiência de sistemas, o que se aplica ao Sokrates.ai (Wang et al., 2024; Bessinger & Bryson, 2023).

Uma cadeia de processamento[6] envolve a decomposição de uma tarefa complexa em etapas menores e sequenciais, onde cada etapa é executada por um agente especializado. No contexto do Sokrates.AI, isso pode incluir:

- Recepção da Pergunta: Um agente recebe a pergunta do usuário e realiza uma análise inicial para identificar palavras-chave e contextos relevantes.
- Recuperação de Documentos: Um segundo agente utiliza essas palavras-chave para buscar documentos relevantes em uma base de dados ou na internet, utilizando técnicas de recuperação de informação (RI) ou embeddings (Costa & Karypis, 2024).
- Processamento de Contexto: Outro agente analisa os documentos recuperados e extrai as informações mais relevantes, criando um resumo ou ponto focal para a resposta.
- Verificação de Respostas Propostas: outro agente verifica a veracidade das respostas encontradas e trata eventuais alucinações.
- Geração de Resposta: Finalmente, um agente LLM utiliza o contexto criado para gerar uma resposta detalhada e precisa, ajustada ao perfil do usuário.

Cada um desses agentes pode ser treinado e ajustado para otimizar sua função específica, garantindo que o sistema como um todo opere de maneira coesa e eficiente. Além disso, os agentes podem empregar diferentes LLMs.

1.6. Agentes Múltiplos

A utilização de agentes múltiplos permite a implementação de funcionalidades que se complementam, proporcionando uma maior robustez ao sistema. Os agentes múltiplos podem ser especializados em diferentes áreas, como linguagem natural, análise de dados, e interação com o usuário[7] (Yulun et al. 2023).

---

[5] https://www.microsoft.com/en-us/research/blog/llmlingua-innovating-llm-efficiency-with-prompt-compression/

[6] https://www.langchain.com/

[7] https://medium.com/@kyeg/multi-agent-vs-single-agent-a72713812b68

Por exemplo, em um sistema como o Sokrates.AI, poderíamos ter, entre outros:

- Agente de Conversação: Especializado em entender e gerar linguagem natural.
- Agente de Recuperação: Focado em buscar e classificar informações relevantes.
- Agente de Análise: Capaz de realizar análises profundas de dados, identificando padrões e insights importantes.
- Agente de Personalização: Responsável por adaptar as respostas com base no histórico e perfil do usuário.

Esses agentes trabalham de forma colaborativa, comunicando-se entre si para trocar informações e otimizar o fluxo de trabalho. Isso permite que o Sokrates.AI promova uma experiência de aprendizado mais personalizada, eficiente e eficaz.

Recentemente, agentes mais abstratos foram propostos por Moura[8], na plataforma Crew.ai. Nessa plataforma, os agentes são descritos de forma independente e colocados em uma equipe de trabalho (*crew*) que pode funcionar em diferentes estilos de cooperação. Como ainda é um projeto em desenvolvimento, atualmente eles podem ser colocados em fila, controlados pela AI principal, por exemplo o ChatGPT, ou hierarquicamente, com um agente sendo o controlador dos demais. Não é feita programação direta, são especificados apenas prompts e contextos para as LLMs, mas funcionalidades específicas podem ser detalhadas para os agentes.

1.7. Perspectivas para o futuro

O campo dos modelos de linguagem de grande escala (LLMs) e dos agentes conversacionais está em constante evolução, apresentando um grande potencial para avanços significativos no futuro próximo. Um dos objetivos é a criação de uma Inteligência Artificial Geral (AGI, do inglês Artificial General Intelligence), que seria capaz de realizar qualquer tarefa intelectual que um ser humano pode realizar (Ge et al., 2024).

Os avanços em hardware e algoritmos de aprendizado de máquina estão acelerando o desenvolvimento de LLMs cada vez mais poderosos. Tecnologias emergentes, como a computação quântica, também podem desempenhar um papel crucial no avanço das capacidades dos LLMs e na realização de AGI. (Mohammad et al., 2023).

2. Comparativo Tecnológico

As LLMs devem ser analisadas pelo seu impacto econômico no negócio. Isso envolve o seu custo, o seu desempenho e ainda fatores naturais do negócio, tanto positivos como n
egativos. Por exemplo, quando uma LLM é usada para determinar uma sugestão a um comprador em função do seu carrinho de compras atual, além do custo de usar a LLM para calcular a sugestão, há o benefício da sugestão

[8] https://github.com/joaomdmoura/crewAI

ser aceita, mas também há o risco da sugestão tirar o comprador da compra. Isso foi analisado pelo coordenador do projeto em (Xexéo et al. 2024).

Esses custos são influenciados pelo tamanho do contexto, pelo tamanho da respostas e pelo tamanho do modelo. Modelos mais baratos são normalmente menores, e podem oferecer, algumas vezes, desempenho similar a modelos mais caros, principalmente para tarefas específicas (Xavier, n.p.).

Em um trabalho ainda não publicado (Xavier, n.p.), fizemos o levantamento de várias LLMs, analisando seu desempenho e custo, em relação a tarefa de eliminação de toxicidade (uma tarefa que deve fazer parte do Sokrates.ai como um agente específico). Nesse trabalho, analisamos as LLM disponíveis no Brasil, incluindo várias de código aberto, e as principais LLMs fornecidas pela OpenAi.

Levantamos os seguintes modelos candidatos a teste, e depois escolhemos variações específicas para uma melhor cobertura.

Tabela 1. Modelos candidatos (iniciais)

Família	Modelos	Lançamento
Gemma	*Gemma-2B-Instruct, Gemma-7B-Instruct*	02/2024
Llama-3	*Llama3-8b-Instruct*	04/2024
Mistral	*Mistral-7B-Instruct-v0.3*	10/2023
Falcon	*Falcon-7b-Instruct*	06/2023

Tabela 2. Modelos candidatos (extensão)

Modelos	Número de Parâmetros	Lançamento
Falcon-40B-Instruct	*40 bilhões*	06/2023
GPT-3.5-turbo-0125	*180 bilhões*	01/2024
GPT-4-0125-preview	*1,76 trilhões*	01/2024
GPT-4o	*1,76 trilhões*	05/2024
Mixtral-7x8B-Instruct-v0.1	*47 bilhões*	12/2023
Llama-3-70B-Instruct	*70 bilhões*	04/2024

A esta coleção ainda é possível somar o modelo da Google (Gemini), porém não foi analisado seu desempenho.

Para avaliar as ferramentas foi usada a tarefa de identificação de toxidade com 16 tipos de prompts diferentes.

O gráfico a seguir, de (Xavier, n.p.) apresenta resultados de custos x acurácia para as diferentes LLMs avaliadas.

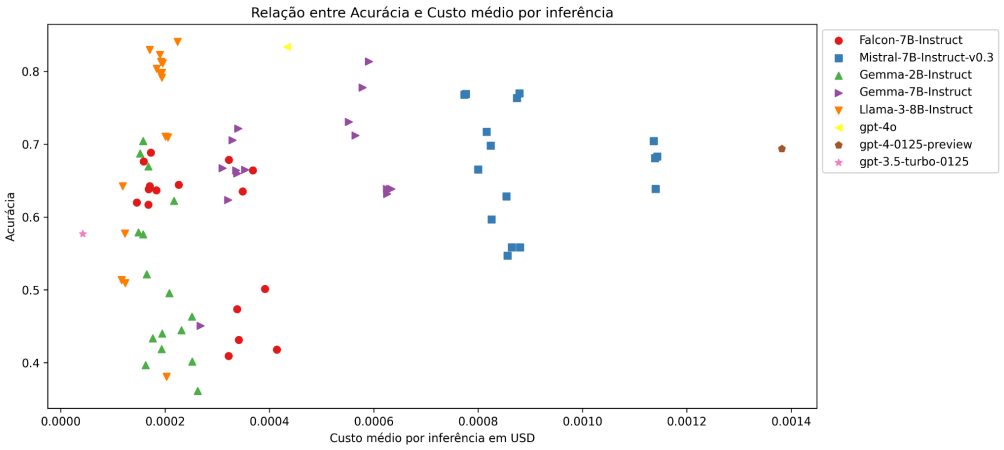


Figura 1. Avaliação de diferentes LLM segundo 16 prompts

Já a tabela a seguir mostra os custos dos modelos avaliados.

Tabela 3. Custos por inferência dos modelos

Modelo	Prompt	Custo por Inferência ($)
Llama-3-70B-Instruct	zero-shot-none	0.025283
gpt-4-0125-preview	zero-shot-none	0.001382
Mistral-7B-Instruct-v0.3	cot-zero-shot-Toxic	0.001143
Gemma-7B-Instruct	few-shot-none	0.000633
gpt-4o	zero-shot-none	0.000434
Falcon-7B-Instruct	cot-few-shot-Not-Toxic	0.000414
Gemma-2B-Instruct	cot-zero-shot-both	0.000262
Llama-3-8B-Instruct	cot-zero-shot-none	0.000224
gpt-3.5-turbo-0125	zero-shot-none	0.000042

Na verdade, a opção de LLM, uma das mais importantes, deve ser feita ao longo de todo o projeto. Primeiro, porque é o custo dominante. Segundo porque é altamente volátil, principalmente como novos lançamentos. Por exemplo, o custo do gpt-3.5-turbo-0125 foi dividio por 10 quando foi lançado o GPT-4. Há a expectativa de um GPT-5 muito mais poderoso no prazo de 1 ou 2 anos.

Além disso, a estrutura prevista do Sokrates.ai permite usar várias LLMs ao mesmo tempo, com um equilíbrio de custo e desempenho, e impacto no negócio, de acordo com necessidades específicas das tarefas.

Outros produtos existentes que devem ser levados em conta são o Gemini[9], ainda não avaliado pelo grupo de pesquisa, e o Claude (ainda não disponível no Brasil)[10].

2.1. Plataformas de desenvolvimento

No desenvolvimento de soluções baseadas em LLMs, como o Sokrates.AI, a escolha da plataforma de desenvolvimento é crucial para

[9] https://gemini.google.com/
[10] https://www.anthropic.com/news/claude-3-5-sonnet

garantir a eficiência, escalabilidade e adaptabilidade do sistema. Entre as principais plataformas de desenvolvimento destacam-se as soluções oferecidas pela Amazon, Microsoft e pela LangChain, além de novas ferramentas emergentes como a CrewAI. Abaixo, apresentamos uma análise inicial de cada uma dessas plataformas.

2.2. Plataforma da Microsoft (baseada na OpenAI)

A Microsoft oferece um conjunto robusto de ferramentas e serviços através de sua plataforma Azure, que suporta o desenvolvimento de modelos de linguagem de grande escala (LLMs). Entre os serviços mais relevantes estão:

- Azure Machine Learning (AML): Facilita o treinamento e a implantação de modelos de aprendizado de máquina em larga escala. AML oferece suporte para modelos pré-treinados e customização de LLMs, além de integração com outras ferramentas de dados e análise da Azure.
- Azure Cognitive Services: Inclui uma variedade de APIs para processamento de linguagem natural (NLP), como análise de sentimentos, tradução de texto, e serviços de Q&A. Essas APIs podem ser integradas aos sistemas para aumentar a funcionalidade e a precisão das respostas geradas.
- Azure Databricks: Um serviço de análise de dados otimizado para o Apache Spark, que permite a preparação e análise de grandes volumes de dados, essenciais para o treinamento de LLMs.
- Azure Kubernetes Service (AKS): Oferece um ambiente gerenciado para a implantação de contêineres, o que é útil para escalar modelos de linguagem e garantir alta disponibilidade e resiliência.

Essas ferramentas, quando combinadas, oferecem uma infraestrutura robusta para o desenvolvimento e a operação de soluções de LLM, garantindo escalabilidade e desempenho.

2.3. Amazon SageMaker (baseada em modelos abertos e comerciais)

Amazon SageMaker é um serviço completamente gerenciado que permite aos desenvolvedores e cientistas de dados criar, treinar e implantar modelos de machine learning rapidamente. Algumas de suas características mais importantes são:

- Treinamento de Modelos: SageMaker facilita o treinamento de modelos em grande escala, utilizando a infraestrutura de computação poderosa da AWS. Ele oferece instâncias com GPUs de alta performance, como as instâncias P3 e P4, que são ideais para treinamento de LLMs.
- Implementação e Gerenciamento: SageMaker oferece ferramentas para a implementação e gerenciamento de modelos, permitindo que os desenvolvedores coloquem seus modelos em produção de forma rápida e escalável. Isso inclui o SageMaker

Endpoint, que pode ser utilizado para servir previsões em tempo real.
- Notebook Instances: Facilita a experimentação e o desenvolvimento de modelos através de notebooks Jupyter integrados, que são pré-configurados com bibliotecas populares de machine learning.

Com o Amazon SageMaker é possível usar modelos Hugging Faces e LangChain.

2.4. LangChain (baseada em modelos abertos)

LangChain é uma coleção de APIs e modelos que se destaca por sua capacidade de integrar diversas bibliotecas e ferramentas de processamento de linguagem natural. Ela é especialmente útil para a construção de pipelines de NLP e sistemas de recuperação e geração de informações. As principais características incluem:

- Integração de Bibliotecas: LangChain permite a integração de várias bibliotecas de NLP, como Hugging Face Transformers, spaCy, e NLTK, facilitando a combinação de diferentes abordagens e técnicas.
- Pipeline Modular: A plataforma suporta a criação de pipelines modulares, onde diferentes etapas do processamento de texto podem ser facilmente configuradas, testadas e ajustadas. Isso inclui etapas de pré-processamento, modelagem, e pós-processamento.
- Suporte para RAG: LangChain oferece suporte nativo para técnicas de Retrieval-Augmented Generation (RAG), permitindo a combinação eficiente de recuperação de informações e geração de texto.

2.5. A flexibilidade e a capacidade de integração de LangChain tornam-na uma escolha atraente para desenvolvedores que buscam uma solução customizável e eficiente para projetos de LLM.

Langchain é open source, criado pela empresa Langchain que oferece serviços adicionais como debugging, traces, etc[11].

2.6. CrewAI

CrewAI[12] é uma plataforma de desenvolvimento (API) relativamente nova, de alto grau de abstração, mas que tem ganhado destaque devido à sua abordagem inovadora e ao uso extensivo de código aberto. As características principais incluem:

- Foco em Colaboração: CrewAI é projetada para facilitar a colaboração entre equipes de desenvolvimento, oferecendo

[11] https://www.langchain.com/
[12] https://www.crewai.com/

ferramentas que suportam o trabalho conjunto em projetos complexos de NLP.
- Código Aberto: A plataforma adota uma filosofia de código aberto, permitindo que desenvolvedores adaptem e estendam suas funcionalidades conforme necessário. Isso promove a inovação e a personalização de soluções.
- Ferramentas Avançadas de NLP: CrewAI integra ferramentas avançadas de NLP, incluindo suporte para modelos de ponta, técnicas de RAG, e engenharia de prompts, além de uma interface amigável para configuração e gestão de pipelines.

A adoção de CrewAI pode ser particularmente vantajosa para equipes que buscam uma solução flexível e colaborativa, que permita rápida adaptação às necessidades específicas do projeto.

3. Comparativo Mercadológico

Decidimos inicialmente comparar nosso produto com dois tipos de soluções no mercado: produtos usados em instituições e organizações de ensino que auxiliam na gestão de aprendizado (Learning Management Systems - LMS) e produtos e serviços que servem como apoio adicional para os alunos, além do fornecido pela sua instituição de ensino principal.

3.1. LMS (Learning Management Systems)

Os sistemas de gerenciamento de aprendizado são ferramentas cruciais para instituições educacionais, facilitando a administração, documentação, acompanhamento, relatórios e entrega de cursos educacionais.

Dois dos principais LMS são o Google Classroom e o Moodle.

- Google Classroom: Uma plataforma gratuita oferecida pelo Google, que integra o G Suite para Educação. Google Classroom permite a criação, distribuição e avaliação de tarefas de maneira simplificada. A plataforma é amplamente adotada devido à sua integração com outros serviços Google, como Google Docs, Google Drive e Google Meet.
- Moodle: Um sistema de gerenciamento de aprendizado de código aberto utilizado globalmente. Moodle é altamente customizável e possui uma vasta gama de plugins e integrações que permitem adaptar a plataforma às necessidades específicas das instituições. É conhecida por sua robustez e flexibilidade, além de ser uma solução escalável para instituições de todos os tamanhos.

Os LMS são plataformas que permitem ao professor organizar um curso, fornecer material, fornecer algumas facilitações, mas não ajudam o aluno a estudar. Tudo depende da capacidade do professor criar e organizar o conteúdo. Porém, isso é de certa forma o serviço básico a ser prestado.

Uma boa forma de implementar o Sokrates.ai seria dentro de um desses ambientes, transformandos em um ambiente inteligente de estudo.

3.2. Plataformas de Apoio ao Estudo

Essas plataformas são utilizadas por alunos para complementar seus estudos, oferecendo recursos adicionais que podem ser acessados a qualquer momento.

Dois exemplos destacados no mercado brasileiro são o Responde Aí e o Descomplica.

- Responde Aí: Uma plataforma online que oferece vídeo-aulas, resumos teóricos e listas de exercícios para estudantes de engenharia e outros cursos técnicos. O Responde Aí é focado em resolver dúvidas pontuais e fornecer material de revisão acessível e de qualidade.
- Descomplica: Uma das maiores plataformas de educação online no Brasil, oferecendo cursos preparatórios para o ENEM, vestibulares, concursos públicos e apoio escolar. O Descomplica se destaca pela vasta quantidade de conteúdo, incluindo aulas ao vivo, videoaulas gravadas, apostilas e simulados.

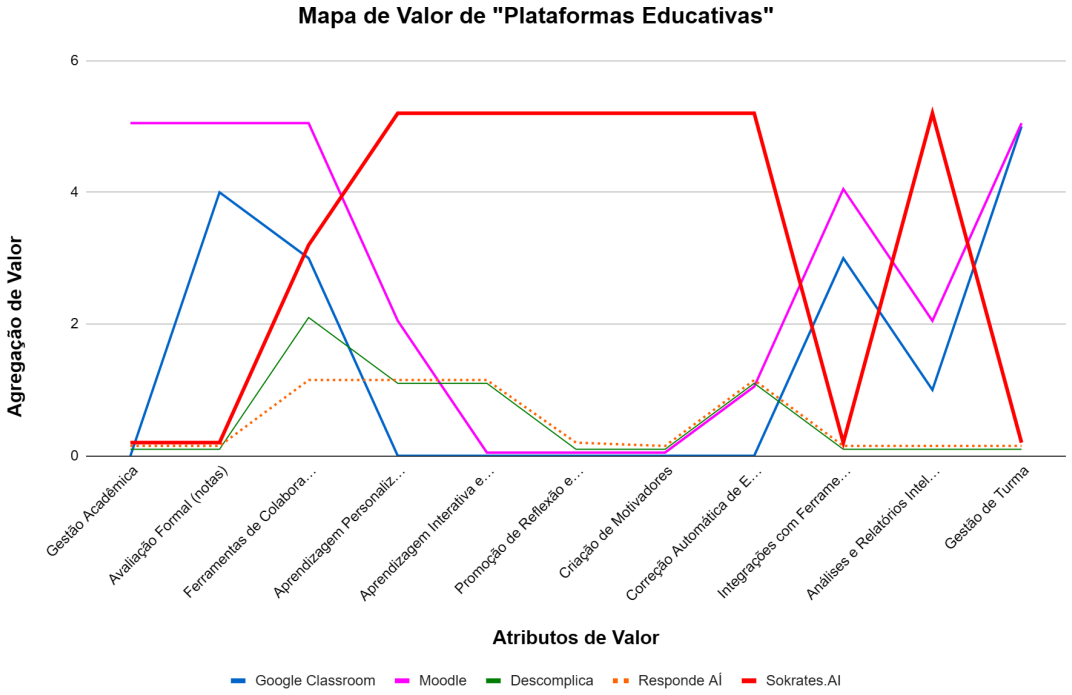
Empresas como Responde Aí e Descomplica são ao mesmo tempo a maior ameaça de concorrência com o Sokrates.ai, quanto uma possível opção de parceria ou mesmo de aquisição da tecnologia ou da empresa, já que o negócio delas, em resumo, parece ser o mesmo do Sokrates.ai, ajudar o aluno a estudar para um objetivo específico.

Ambas as empresas, porém, dependem da criação de conteúdo por conteudistas, e apresentam esse conteúdo de forma direta. O conhecimento do mercado, por meio de conversas que temos tido, mostra que há um grande interesse de empresas desse mercado na correção automática de exercícios, que seria um dos serviços possivelmente propostos pelo Sokrates.ai.

Uma versão preliminar da Matriz de Valor baseada na Estratégia do Oceano Azul, busca posicionar o Sokrates.ai frente aos atributos de valor das plataformas Google Classroom, Moodle, Descomplica e Responde AÍ.

	Foco no Aprendiz						Foco no Educador				
Atributo de Valor Plataforma	Gestão Acadêmica	Avaliação Formal (notas)	Ferramentas de Colaboração	Aprendizagem Personalizada	Aprendizagem Interativa e Socrática	Promoção de Reflexão e Pensamento Crítico	Criação de Motivadores	Correção Automática de Exercícios	Integrações com Ferramentas Específica	Análise s e Relatórios Inteligentes	Gestão de Turma
Google Classroom	0	4	3	0	0	0	0	0	3	1	5
Moodle	5	5	5	2	0	0	0	1	4	2	5
Descomplica	0	0	2	1	1	0	0	1	0	0	0
Responde AÍ	0	0	1	1	1	0	0	1	0	0	0
Sokrates.AI	0	0	3	5	5	5	5	5	0	5	0

A Matriz de Valor pode ser visualizada na forma do Mapa de Valor a seguir.



Os atributos de valor analisados bem como o grau de impacto de cada atributo na criação de valor proporcionada pelo Sokrates.ai frente a diferentes plataformas serão refinados no decorrer do projeto e à luz do processo de “Customer Discovery, conforme proposto por Steve Blank e Bob Dorf (2012).

4. Bibliografia (Ordem Alfabética)

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of ICLR.

Bender, Emily M.; Gebru, Timnit; McMillan-Major, Angelina; Mitchell, Margaret. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. 2021. p. 610-623.

Blank, Steve; Dorf, Bob. The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company. In: K&S Ranch Press, 2012.

Brown, Tom B.; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; Sastry, Girish; Askell,, Amanda et al. Language Models are Few-Shot Learners. In: Advances in neural information processing systems. 2020. p. 1877-1901.

Costas Mavromatis, & George Karypis. (2024). GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning.

Ge, Y., Hua, W., Mei, K., Tan, J., Xu, S., Li, Z., & Zhang, Y. (2024). OpenAGI: When LLM meets domain experts. Advances in Neural Information Processing Systems, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Kim, W. Chan; Mauborgne, Renée. A Estratégia do Oceano Azul: Como Criar Novos Mercados e Tornar a Concorrência Irrelevante. In: Harvard Business Review Press, 2011.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Advances in Neural Information Processing Systems (pp. 9459–9474). Curran Associates, Inc.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, & Percy Liang. (2023). Lost in the Middle: How Language Models Use Long Contexts.

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023, June). Prompt engineering in large language models. In International conference on data intelligence and cognitive informatics (pp. 387-402). Singapore: Springer Nature Singapore.

Misnevs, B. (2023). Socratic method for prompt engineering–new teaching design. Step into the Future, 18(2), 10-11.

Mohammad, A. F., Clark, B., Agarwal, R., & Summers, S. (2023, July). LLM/GPT Generative AI and Artificial General Intelligence (AGI): The Next Frontier. In 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE) (pp. 413-417). IEEE.

Nascimento, O. L. D. (2023). Estudo comparativo entre modelos transformers aplicados ao desenvolvimento de chatbots (Master's thesis, Universidade Federal do Rio Grande do Norte).

Rane, N., Choudhary, S., & Rane, J. (2024). Gemini versus ChatGPT: applications, performance, architecture, capabilities, and implementation. Performance, Architecture, Capabilities, and Implementation (February 13, 2024).

Ilya Sutskever, Oriol Vinyals, & Quoc V. Le. (2014). Sequence to Sequence Learning with Neural Networks.

Thanasi-Boçe, M., & Hoxha, J. (2024). From ideas to ventures: building entrepreneurship knowledge with LLM, prompt engineering, and conversational agents. Education and Information Technologies, 1-57.

Vasmani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N.; Kaiser, łUkasz; Polosukhin, Illia. Attention is all you need. In: Advances in neural information processing systems. 2017. p. 5998-6008.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, & Igor Mordatch. (2023). Improving Factuality and Reasoning in Language Models through Multiagent Debate.

Yucheng Hu, & Yuxing Lu. (2024). RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing.

Geraldo Xexéo, Filipe Braida, Marcus Parreiras, & Paulo Xavier. (2024). The Economic Implications of Large Language Model Selection on Earnings and Return on Investment: A Decision Theoretic Model.