

Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro



Rafael Alves Goiás



UFG

UNIVERSIDADE
FEDERAL DE GOIÁS

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

RAFAEL ALVES GOIÁS

Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado
Financeiro

Goiânia
2025



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): RAFAEL ALVES GOIÁS

Título do trabalho: Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento SIM NÃO¹

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

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

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



Documento assinado eletronicamente por **Rafael Alves Goias, Discente**, em 13/01/2025, às 09:35, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Fernando Marques Federson, Professor do Magistério Superior**, em 15/01/2025, às 16:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089803** e o código CRC **5B4DF198**.

Referência: Processo nº 23070.001598/2025-92

SEI nº 5089803

RAFAEL ALVES GOIÁS

Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

GOIÁS, RAFAEL ALVES

Sistemas Multiagentes baseados em LLMs [manuscrito] : Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro / RAFAEL ALVES GOIÁS. - 2025.
135 f.

Orientador: Prof. Dr. Fernando Marques Federson.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. modelos de linguagem. 3. análise financeira. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

RAFAEL ALVES GOIÁS

Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 17 de dezembro de 2024.



Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)



Prof. Dr. Aldo André Diaz Salazar
Coordenador de TCC do BIA (INF-UFG)



Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)



Prof. Dr. Ivens Gervasio Sene Junior
(INF-UFG)

RAFAEL ALVES GOIÁS

Sistemas Multiagentes baseados em LLMs

Uma Abordagem baseada em Large Language Models para Análise do Mercado Financeiro

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Sistemas Multiagentes baseados em LLMs**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

ABSTRACT

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

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

Goiânia

2025

Minha Jornada

Rafael Alves Goiás

Especialista em: Sistemas Multiagentes baseados em LLMs



Semana 1 e 2

Exploração inicial de LLMs, agentes e sistema multiagentes.

Semana 3

Mapeamento do estado da arte, incluindo ferramentas e frameworks.

Semanas 4 e 5

Implementação inicial de agentes no mercado financeiro e definição do fluxo para predição e geração de relatórios.

Semanas 6 e 7

Documentação e estudo de métricas financeiras para suporte ao fluxo, integração de APIs e aperfeiçoamento do Sistema.

Semana 8 e 9

Conclusão do Sistema, ajustes finais e avaliação de desempenho.

MINHA JORNADA

Nome: Rafael Alves Goiás

Especialidade: Sistemas Multiagentes baseados em LLMs

Objetivo deste documento

Durante o processo da disciplina Residência em IA¹, foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada iniciou com o interesse em Large Language Models (LLMs) e suas aplicações em agentes autônomos e sistemas multiagentes. Na **Semana 1**, estudei os temas propostos em conferências como ICAI 2024 e ICDATA 2024, o que me ajudou a definir a área de atuação para meu programa. A partir daí, entre 29 de julho e 16 de setembro de 2024, dediquei-me à análise de artigos recentes e ao estudo dos conceitos fundamentais da área, buscando compreender tendências e identificar avanços tecnológicos relevantes. Um marco desse período foi o estudo aprofundado do artigo "A Survey on Large Language Model based Autonomous Agents", que ofereceu uma base teórica sólida sobre agentes baseados em LLMs e serviu como alicerce para a continuação da minha jornada. Durante a **Semana 2**, aprofundei-me nos conceitos de agentes e sistemas multiagentes, explorando suas interações e aplicações práticas. Analisei alguns artigos pertinentes e bastante referenciados em várias áreas de aplicação e frameworks como MetaGPT, ChatDev e Generative Agents, que ampliaram minha visão sobre o potencial desses sistemas. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 1**.

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

Na **Semana 3**, concentrei-me na investigação de ferramentas e estudos considerados “estado da arte” em Agentes e Sistemas Multiagentes baseados em Grandes Modelos de Linguagem (LLMs). Minha análise priorizou artigos com mais de 50 citações, repositórios com mais de 200 estrelas e publicações de alto impacto, como: LLM Agents Papers, LLM-Agents-Papers, Awesome LLM-Powered Agent. Também organizei uma árvore conceitual dos tópicos mapeados, que facilitou a visualização dos conhecimentos adquiridos. O mapa consistiu em três ramificações que são: Resolução de problemas, simulação de mundo e benchmarks em diversas frentes. Essa Semana foi fundamental para consolidar minha compreensão sobre as ferramentas mais relevantes e direcionar o próximo passo da pesquisa. Os materiais relacionados a esta Semana podem ser encontrados no **Apêndice 2**.

Na **Semana 4**, avancei na pesquisa e na implementação prática do tema, com foco no mercado financeiro. Desenvolvi um agente básico de forma isolada, sem utilizar ferramentas prontas, com o objetivo de compreender melhor a estrutura e os conceitos fundamentais. Paralelamente, pesquisei sobre simulação de mercado financeiro, analisando como diferentes perfis de investidores reagem a políticas e notícias, com base em empresas chinesas listadas na bolsa de valores na ideia de seguir neste caminho nas próximas etapas. Continuando o desenvolvimento na **Semana 5**, aprofundi os estudos sobre interações multiagentes e simulações aplicadas ao mercado financeiro. Desenvolvi um fluxo de interação multiagente, ainda sem o uso de frameworks, utilizando o agente desenvolvido previamente. Além disso, continuei explorando artigos relacionados à aplicação de agentes no mercado financeiro, extraíndo insights sobre os objetivos e métodos da simulação. Estudei seus conceitos teóricos e práticos, realizei testes iniciais e desenvolvi um código para simulações interativas, fora do escopo proposto previamente, incluindo um exemplo realista. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 3**.

Durante as **Semanas 6 e 7**, finalizei o escopo do trabalho, definindo a área de atuação na resolução de problemas no mercado financeiro, com foco em análises fundamentalistas. O objetivo principal foi desenvolver um sistema multiagente para predição

de mercado e geração de relatórios financeiros, estruturado com agentes especializados para coordenar o fluxo de trabalho, interpretar dados financeiros e realizar análises quantitativas detalhadas. Também aprofundi estudos sobre ferramentas e frameworks relevantes, integrando conceitos teóricos e práticos no desenvolvimento inicial do sistema. Esses esforços resultaram em avanços significativos, consolidando uma base sólida para os próximos passos da pesquisa. Os materiais relacionados a estas Semanas podem ser encontrados no **Apêndice 4**.

Nas últimas **Semanas, 8 e 9**, o trabalho focou em refinar e finalizar o sistema desenvolvido. Realizei ajustes no fluxo automatizado, simplificando entradas e resolvendo problemas técnicos, como a integração com APIs e a formatação de saídas. As melhorias incluíram a inclusão de campos para comparação entre empresas concorrentes, automação da geração de relatórios em PDF e uma metodologia para avaliar a qualidade das análises. A avaliação incluiu métricas de desempenho, como taxa de acerto e precisão, destacando o alto desempenho na previsão de direções, embora com margem de melhoria nas estimativas de amplitude. Por fim, refinei prompts e instruções para melhorar a precisão e padronização das respostas dos agentes. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 5**.

Ao final desta jornada, posso afirmar que o processo foi uma experiência enriquecedora e desafiadora, que me permitiu crescer tanto técnica quanto pessoalmente. Cada etapa trouxe novos aprendizados, mostrando que dedicação e foco são fundamentais para superar desafios e alcançar resultados concretos. Essa trajetória me ensinou a importância de alinhar teoria e prática, adaptando-me às necessidades e refinando constantemente meu trabalho. Saio deste período com uma visão mais clara do meu potencial, motivada a continuar explorando novas possibilidades e confiante de que cada esforço investido contribui para meu desenvolvimento e para resultados de impacto.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Essa semana foi dedicada à exploração e estudos de artigos na minha área de interesse, sendo dividido em 2 etapas:

1. Pesquisa e análise de artigos recentes no campo de Large Language Models, com foco em desenvolvimentos e tendências emergentes. A análise foi feita a partir de dois canais de divulgação amplamente acessíveis: [NLPlanet - Natural Language Processing Community](#) e [NLP Newsletter - DAIR.AI](#), com a coleta de materiais publicada entre os dias 29 de julho de 2024 e 16 de setembro de 2024.

- Resumo do estudo feito: [Resumo artigos - 16/09/24](#)

2. Estudo e pesquisa sobre agentes baseados em LLMs: [A Survey on Large Language Model based Autonomous Agents arXiv:2308.11432v5 \[cs.AI\] 4 Apr 2024](#)

- Resumo do estudo feito: [A Survey on Large Language Model based Autonomous Agents - 1...](#)

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

- Como próxima entrega, pretendo ler e levantar os pontos principais em relação a um outro estudo sobre multi-agentes: [arXiv:2402.01680v2 \[cs.CL\] 19 Apr 2024](#)
- Investigar e analisar em profundidade os recursos e capacidades de alguns agentes específicos apresentados no estudo citado anteriormente. Os modelos são: Generative Agents, ChatDev e MetaGPT
- Procurar e ler referências sobre trabalhos que minimizem as limitações e desafios enfrentados relatados na seção 6 do documento:

[A Survey on Large Language Model based Autonomous Agents - 16/09/24](#)

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

Resumo e estatística dos principais artigos na área de NLP/LLMs de 29/07/2024 - 16/09/2024

A área de processamento de linguagem natural é um campo extenso e complexo que se atualiza muito rapidamente. Todos os dias, novos artigos são lançados, incluindo técnicas novas, modelos otimizados, ferramentas, guias de estudos e várias outras frentes. A ideia por trás desse estudo é ver como e para onde estão caminhando os desenvolvimentos nessa área. Para isso, escolhi dois canais de conteúdo abertos ao público e analisei os principais artigos postados desde o dia 29 de julho de 2024 até 16 de setembro de 2024. Os canais de divulgação são: [NLPlanet - Natural Language Processing community](#) e [NLP Newsletter](#). Ambos são focados apenas na área de NLP e possuem centenas de pessoas que os acompanham semanalmente

NLPlanet - Natural Language Processing community e NLP Newsletter:

Avanços em LLMs:

- [Llama 3.1 \(405B, 70B, 8B\)](#): Meta lança um modelo de código aberto com capacidades multilinguísticas, de raciocínio e uso de ferramentas, com melhorias de treinamento e maior comprimento de contexto.
- [Mistral Large 2](#): Modelo com 123 bilhões de parâmetros e uma janela de contexto de 128k tokens, projetado para gerar código e suportar linguagens de programação e humanas.
- [Llama 4](#): A Meta prevê um aumento de 10x no poder computacional necessário para treinar o Llama 4 em comparação ao Llama 3.
- [Gemma 2 2B](#): Novo modelo de IA da Google, eficiente em classificação de conteúdo e interpretabilidade de modelos.

- GPT-4o mini: Modelo de baixo custo da OpenAI, oferecendo inteligência eficiente com custos reduzidos em comparação ao GPT-3.5 Turbo.

Insights:

- Escalonamento de Modelos: Há uma clara tendência de escalonamento nos modelos LLMs, tanto em termos de número de parâmetros quanto em capacidade computacional.
- Eficiência Computacional: O desenvolvimento de modelos mais eficientes que mantêm ou melhoram o desempenho é uma prioridade, como visto no GPT-4o mini e Gemma 2 2B.
- Comprimento de Contexto: O aumento da janela de contexto permite que os modelos processem sequências maiores, essencial para tarefas complexas como geração de código e compreensão de textos extensos.

Modelos Multimodais e Multilinguísticos:

- [Llama 3 Herd of Models](#): Conjunto de modelos da Meta com 405 bilhões de parâmetros, com suporte para multimodalidade e processamento de imagens, vídeos e fala.
- [xGen-MM \(BLIP-3\)](#): Framework para o desenvolvimento de modelos multimodais avançados.
- [LongLLaVA](#): LLM multimodal que processa grandes volumes de imagens de forma eficiente.
- Mistral releases Pixtral 12B: Modelo multimodal que processa texto e imagens.
- [LLaMA-Omni](#): Modelo de interação de fala eficiente para conversão de fala para texto e vice-versa.

Insights:

- Integração de Múltiplas Modalidades: A incorporação de texto, imagem, vídeo e fala em um único modelo amplia significativamente as aplicações possíveis dos LLMs.
- Comunicação Multilíngue: A capacidade de operar em múltiplas línguas facilita a adoção global e a aplicação em contextos multiculturais.
- Aplicações Práticas: Esses modelos permitem avançar em áreas como tradução automática, legendagem de vídeos e assistentes virtuais mais naturais.

Aplicações Específicas em Domínios Técnicos:

- [AlphaProof e AlphaGeometry 2](#): IAs que atingiram o nível de medalha de prata ao resolver problemas da Olimpíada Internacional de Matemática.
- Medical SAM 2: Aplicação do modelo SAM 2 para segmentação de imagens médicas em 2D e 3D, com avanços em diagnóstico assistido por IA.
- [Chemlactica e Chemma](#): Modelos de linguagem voltados para otimização de moléculas pequenas, com técnicas de amostragem e algoritmos genéticos.
- [Med42-v2](#): Conjunto de LLMs clínicos avançados para consultas médicas.
- [Nvidia StormCast](#): Modelo avançado para previsão de clima em escala mesoscalar.

Insights:

- Especialização de Modelos: Há um movimento em direção a modelos especializados que são altamente eficientes em domínios específicos, como medicina, química e meteorologia.
- Impacto na Pesquisa e Desenvolvimento: Esses modelos podem acelerar descobertas científicas, auxiliar em diagnósticos médicos e melhorar previsões climáticas.

Melhorias em Treinamento e Eficiência de Modelos:

- [Spectra](#): Estudo abrangente sobre modelos de linguagem quantizados e ternários (TriLMs), mostrando que eles podem alcançar desempenho equivalente aos modelos de ponto flutuante FP16, com uso reduzido de memória.
- [Memory-Efficient LLM Training](#): Otimizador para treinar LLMs de forma mais eficiente em termos de memória.
- [Matryoshka-Adaptor](#): Framework para reduzir as dimensões das embeddings de LLMs de forma eficiente.
- [How to Prune and Distill Llama-3.1 8B](#): Guia sobre técnicas de compressão e destilação de modelos.

Insights:

- Otimização de Recursos: A redução de requisitos computacionais e de memória é crucial para tornar os LLMs mais acessíveis e sustentáveis.
- Técnicas de Compressão: Pruning, quantização e destilação são áreas de pesquisa ativa que melhoram a eficiência sem sacrificar significativamente o desempenho.
- Aplicabilidade em Dispositivos Limitados: Modelos mais leves podem ser implementados em dispositivos com recursos limitados, ampliando o alcance da tecnologia.

Novas Ferramentas e Frameworks:

- [RAG Foundry](#): Framework open-source para melhorar modelos de Geração Aumentada por Recuperação (RAG), com foco em uso de fontes de conhecimento diversas.
- [DSPy framework](#): Guia sobre o framework open-source DSPy, que coordena múltiplas chamadas de LLMs para resolver problemas complexos.
- [Transformer Explainer](#): Artigo interativo explicando o funcionamento dos modelos Transformer, com ferramentas para explorar o comportamento do modelo em tempo real.
- [iText2KG](#): Ferramenta para construir gráficos de conhecimento a partir de texto com LLMs.

Insights:

- **Facilitação do Desenvolvimento**: Novas ferramentas e frameworks estão tornando mais fácil para pesquisadores e desenvolvedores trabalharem com LLMs.
- **Transparência e Interpretabilidade**: Ferramentas como o Transformer Explainer ajudam a entender melhor como os modelos funcionam, o que é vital para a confiança e melhoria contínua.
- **Integração de Conhecimento**: Frameworks que incorporam fontes externas de conhecimento podem melhorar a precisão e relevância das respostas dos modelos.

Considerações Legais e Éticas:

- [Perplexity AI enfrentando desafios legais](#): A Condé Nast enviou uma ordem de cessar e desistir ao Perplexity AI, destacando os desafios legais emergentes no uso de dados na IA.
- **Bloqueios de dados na web**: Estudo mostra o aumento de domínios da web que impedem a extração de dados para treinar modelos de IA, possivelmente levando a vieses nos modelos de IA.
- [Grok AI de Elon Musk](#): Modelo poderoso sendo treinado com dados do X, levantando questões de privacidade.
- [AMD está se tornando cada vez mais uma empresa focada em chips de IA](#): Implicações éticas e econômicas dessa mudança.

Insights:

- **Proteção de Dados**: A coleta e uso de dados para treinar modelos de IA estão sob escrutínio legal e ético, o que pode afetar a disponibilidade de dados para pesquisa.
- **Vieses e Equidade**: Restrições de dados podem levar a vieses nos modelos, afetando a justiça e a representatividade.

- Privacidade: O uso de dados pessoais, como os do X (antigo Twitter), levanta preocupações significativas de privacidade.

Aplicações em Desenvolvimento de Software e Programação:

- [Programação orientada a chat \(CHOP\)](#): Exploração de como ferramentas de IA interativas estão transformando a programação, especialmente no refinamento de prompts e no entendimento de código.
- [Replit Agent](#): Assistente de IA para desenvolvimento de software que entende linguagem natural.
- [Amazon Q](#): Assistente de software que economizou 4.500 anos de trabalho de desenvolvedores ao automatizar a transformação de sistemas Java.
- Codestral Mamba: Modelo de IA focado em codificação eficiente, gerenciando sequências longas com inferência em tempo linear e suporte para sequências teóricas ilimitadas.

Insights:

- Automação de Tarefas de Codificação: A IA está sendo usada para automatizar partes do desenvolvimento de software, aumentando a produtividade.
- Interação Natural com Código: Ferramentas que permitem programar usando linguagem natural estão tornando a programação mais acessível.
- Desafios e Oportunidades: Há espaço para pesquisa sobre como essas ferramentas impactam o desenvolvimento de software e quais são os desafios associados.

Melhorias em Raciocínio e Resolução de Problemas:

- ["Chain of Thought" em IAs](#): Exploração do impacto positivo que o planejamento em etapas tem sobre a qualidade das respostas de IA.
- [CoD \(Chain of Diagnosis\)](#): Método que melhora a interpretabilidade de diagnósticos médicos, emulando o raciocínio médico por meio de modelos de linguagem.
- [OpenAI o1-preview](#): Modelo avançado de raciocínio que supera modelos anteriores em tarefas complexas de ciência e matemática.
- [Imitating Language via Scalable Inverse Reinforcement Learning](#): Uso de aprendizado por reforço para refinar LLMs.

Insights:

- Raciocínio em Etapas: Abordagens que emulam o raciocínio humano passo a passo melhoram a precisão e interpretabilidade dos modelos.

- Aprendizado por Reforço: Técnicas avançadas de treinamento, como aprendizado por reforço inverso, estão sendo aplicadas para aprimorar a capacidade de resolução de problemas dos LLMs.
- Aplicações em Educação e Ciência: Modelos com melhores habilidades de raciocínio têm grande potencial em educação, pesquisa científica e suporte à tomada de decisão.

Tendências de Negócios e Investimentos em IA:

- OpenAI enfrenta possíveis perdas: Estimativa de perdas operacionais de US\$ 5 bilhões em 2024, apesar de ganhos substanciais com o ChatGPT.
- Groq levanta US\$ 640 milhões: Financiamento para atender à crescente demanda por inferência rápida de IA, com expansão de sua capacidade de LPUs.
- AMD adquire ZT Systems: Expansão do ecossistema de IA da AMD com foco em infraestrutura.
- OpenAI em negociações para levantar US\$ 11,5 bilhões em uma avaliação de US\$ 150 bilhões: Relevante para o crescimento da indústria de IA.

Insights:

- Mercado em Crescimento: O intenso investimento financeiro indica um mercado de IA em rápida expansão e altamente competitivo.
- Sustentabilidade Financeira: Apesar dos altos custos operacionais, empresas como a OpenAI estão buscando estratégias para manter a sustentabilidade financeira.
- Oportunidades de Pesquisa: Analisar modelos de negócio e estratégias de monetização em IA pode ser um tema relevante para o seu TCC.

Contagem dos artigos e notícias:

Avanços em modelos de linguagem de grande escala (LLMs): 5 artigos

Modelos multimodais e multilinguísticos: 5 artigos

Aplicações específicas em domínios técnicos: 5 artigos

Melhorias em treinamento e eficiência de modelos: 4 artigos

Novas ferramentas e frameworks: 4 artigos

Considerações legais e éticas: 4 artigos

Aplicações em desenvolvimento de software e programação: 4 artigos

Melhorias em raciocínio e resolução de problemas: 4 artigos

Tendências de negócios e investimentos em IA: 4 artigos

Tendências Identificadas:

1. **Expansão de LLMs:** O foco em modelos maiores e mais capazes indica uma corrida por avanços em inteligência artificial geral.
2. **Multimodalidade em ascensão:** A integração de múltiplas formas de dados está se tornando crucial para aplicações mais robustas e versáteis.
3. **Eficiência como necessidade:** A otimização de recursos é uma resposta aos desafios computacionais impostos por modelos cada vez maiores.
4. **Especialização setorial:** Modelos voltados para setores específicos, como saúde e programação, destacam a importância da personalização.
5. **Preocupações éticas e legais crescentes:** O número significativo de artigos nessa área reflete uma conscientização sobre os impactos sociais da IA.

A Survey on Large Language Model based Autonomous Agents

Proposto por vários pesquisadores chineses da Escola de inteligência artificial de Gaoling na universidade de Renmin em Beijing, na China.

A ideia é conhecer melhor a área de agentes, aprofundar nesses critérios, conhecer agentes existentes, limitações e obter ideias restringindo o tema que devo atacar.

Para melhorar a organização deste documento, vou dividi-lo em 6 partes, sendo:

1. Introdução:

O texto fala sobre pesquisas envolvendo agentes autônomos, que são sistemas capazes de tomar decisões sozinhos. Antigamente, esses agentes eram treinados de forma limitada e em ambientes controlados, o que dificultava que eles aprendessem e tomassem decisões como os humanos. Mas, com o surgimento dos LLMs, que são sistemas treinados com uma enorme quantidade de informação da internet, os pesquisadores perceberam que esses modelos poderiam ter uma inteligência parecida com a humana. O artigo faz uma revisão detalhada de como esses agentes autônomos baseados em LLMs estão sendo construídos, como estão sendo usados em diferentes áreas (como ciências e engenharia), como são avaliados e quais são os desafios e as futuras direções de pesquisa nesse campo.

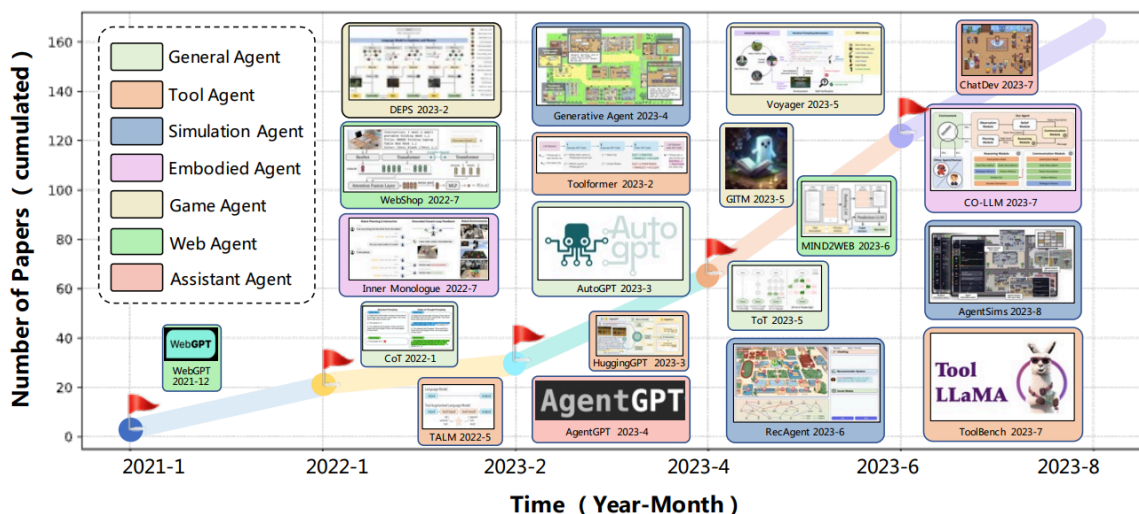


Ilustração sobre os agentes autônomos baseados em LLMs no decorrer do tempo.

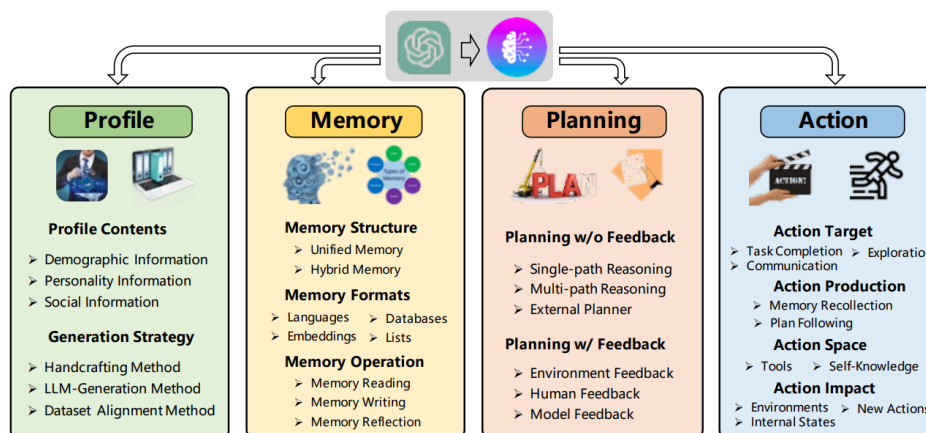
2. Construção dos agentes:

A pesquisa foi feita com base em três aspectos: a construção, aplicação e avaliação de agentes autônomos baseados em LLMs.

Para a construção do agente, focaram em dois problemas:

- (1) como projetar a arquitetura do agente para melhor aproveitar os LLMs, e
- (2) como inspirar e aprimorar a capacidade do agente para completar diferentes tarefas.

Para preencher a lacuna entre os LLMs tradicionais e os agentes autônomos, um aspecto crucial é projetar arquiteturas de agentes racionais para ajudar os LLMs a maximizar suas capacidades. Nessa direção, trabalhos anteriores desenvolveram diversos módulos para aprimorar os LLMs. Assim, foi proposto um framework unificado para resumir esses módulos composta por um módulo de perfil, um módulo de memória, um módulo de planejamento e um módulo de ação.



2.1. Módulo de Perfil:

O módulo de perfil tem como objetivo indicar os perfis dos papéis dos agentes, que geralmente são escritos no prompt para influenciar os comportamentos do LLM. Os perfis dos agentes normalmente abrangem informações básicas, como idade, gênero e carreira, bem como informações psicológicas, refletindo as personalidades do agente, e informações sociais, detalhando os relacionamentos entre agentes.

- Método de Criação Manual: Neste método, os perfis dos agentes são especificados manualmente.

- Método de Geração por LLM: Neste método, os perfis dos agentes são gerados automaticamente com base em LLMs.
- Método de Alinhamento de Conjunto de Dados: Neste método, os perfis dos agentes são obtidos a partir de conjuntos de dados do mundo real. Normalmente, organiza-se a informação sobre humanos reais nos conjuntos de dados em prompts de linguagem natural, e depois usa-se isso para perfilar os agentes.

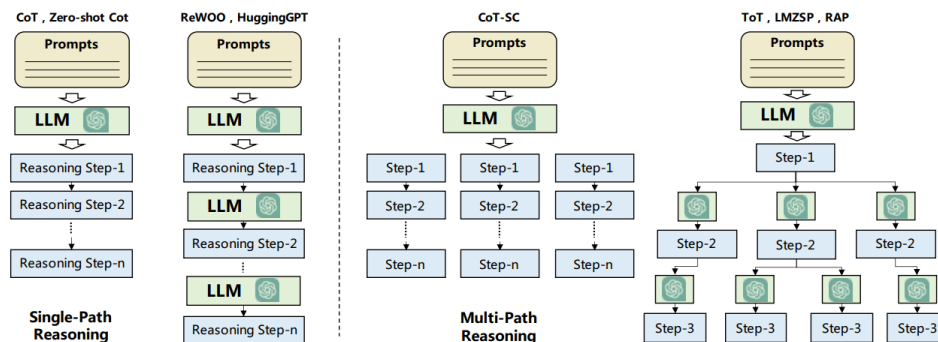
2.2. Módulo de Memória:

O módulo de memória desempenha um papel muito importante no design da arquitetura de agentes. Ele armazena informações percebidas do ambiente e utiliza as memórias registradas para facilitar ações futuras.

- Estrutura de Memória Unificada: Esta estrutura simula apenas a memória de curto prazo humana, geralmente implementada por meio do aprendizado dentro do contexto, e as informações de memória são diretamente escritas nos prompts.
- Estrutura de Memória Híbrida: Essa estrutura modela explicitamente as memórias de curto e longo prazo humanas. A memória de curto prazo armazena temporariamente percepções recentes, enquanto a memória de longo prazo consolida informações importantes ao longo do tempo.
- As memórias podem ser armazenadas em linguagem natural, embeddings, bancos de dados e listas estruturadas.
- Algumas operações na memória como escrita, leitura e reflexão podem existir.

2.3. Módulo de Planejamento:

Quando enfrentam uma tarefa complexa, os humanos tendem a decompô-la em sub-tarefas mais simples e resolvê-las individualmente. O módulo de planejamento tem o objetivo de dotar os agentes dessa capacidade humana, permitindo que eles ajam de forma mais racional, poderosa e confiável.



- Planejamento Sem Feedback: Os agentes não recebem feedback que possa influenciar seus comportamentos futuros após a execução de uma ação.
- Planejamento Com Feedback: Nos métodos anteriores, sem feedback, o planejamento pode ser menos eficaz por várias razões, como a dificuldade de gerar um plano perfeito desde o início e a imprevisibilidade dos ambientes. Por isso, pesquisadores desenvolveram módulos de planejamento onde o agente pode receber feedback após a execução de ações. O feedback pode vir do ambiente, de humanos ou de modelos.
- Raciocínio de caminho único e Raciocínio de caminho múltiplo.

2.4. Módulo de Ação:

O Módulo de Ação é responsável por transformar as decisões do agente em resultados práticos, interagindo diretamente com o ambiente. Ele é influenciado pelos módulos de perfil, memória e planejamento, e pode ser entendido sob quatro perspectivas:

- Meta da Ação: O agente pode realizar ações para diferentes fins, como completar tarefas específicas (ex.: construir algo em Minecraft), se comunicar com outros agentes ou explorar ambientes desconhecidos para expandir seu conhecimento.
- Produção da Ação: As ações podem ser geradas de duas formas: recuperando informações da memória do agente para guiar a ação ou seguindo planos pré-estabelecidos, dividindo tarefas em sub-objetivos.
- Espaço de Ação: Refere-se às ações possíveis que o agente pode tomar, como usar ferramentas externas (APIs, bancos de dados, modelos externos) para aumentar suas capacidades e executar tarefas mais complexas.
- Impacto da Ação: Trata-se das consequências diretas das ações, como o sucesso ou falha da tarefa, mudanças no ambiente ou interações com outros agentes.

3. Principais Agentes levantados no artigo:

Na tabela, os diferentes métodos e estratégias são representados por números, com os seguintes significados:

- Módulo de Perfil:
 - ①: Método manual (handcrafting)
 - ②: Método de geração por LLM
 - ③: Método de alinhamento de dados

- Módulo de Memória:
 - Operação de Memória:
 - ①: Apenas operações de leitura/escrita
 - ②: Operações de leitura/escrita/reflexão

- Estrutura de Memória:
 - ①: Memória unificada
 - ②: Memória híbrida

- Módulo de Planejamento:
 - ①: Planejamento sem feedback
 - ②: Planejamento com feedback

- Módulo de Ação:
 - ①: O modelo não usa ferramentas
 - ②: O modelo usa ferramentas

- Estratégia de Aquisição de Capacidade do Agente (CA):
 - ①: Métodos com fine-tuning
 - ②: Métodos sem fine-tuning

Se um aspecto não for discutido no artigo, será representado por “-”.

Model	Profile	Memory		Planning	Action	CA	Time
		Operation	Structure				
WebGPT [66]	-	-	-	-	②	①	12/2021
SayCan [78]	-	-	-	①	①	②	04/2022
MRKL [72]	-	-	-	①	②	-	05/2022
Inner Monologue [61]	-	-	-	②	①	②	07/2022
Social Simulacra [79]	②	-	-	-	①	-	08/2022
ReAct [59]	-	-	-	②	②	①	10/2022
MALLM [43]	-	①	②	-	①	-	01/2023
DEPS [33]	-	-	-	②	①	②	02/2023
Toolformer [15]	-	-	-	①	②	①	02/2023
Reflexion [12]	-	②	②	②	①	②	03/2023
CAMEL [80]	① ②	-	-	②	①	-	03/2023
API-Bank [69]	-	-	-	②	②	②	04/2023
ViperGPT [74]	-	-	-	-	②	-	03/2023
HuggingGPT [13]	-	-	①	①	②	-	03/2023
Generative Agents [20]	①	②	②	②	①	-	04/2023
LLM+P [57]	-	-	-	①	①	-	04/2023
ChemCrow [75]	-	-	-	②	②	-	04/2023
OpenAGI [73]	-	-	-	②	②	①	04/2023
AutoGPT [81]	-	①	②	②	②	②	04/2023
SCM [35]	-	②	②	-	①	-	04/2023
Socially Alignment [82]	-	①	②	-	①	①	05/2023
GITM [16]	-	②	②	②	①	②	05/2023
Voyager [38]	-	②	②	②	①	②	05/2023
Introspective Tips [83]	-	-	-	②	①	②	05/2023
RET-LLM [42]	-	①	②	-	①	①	05/2023
ChatDB [40]	-	①	②	②	②	-	06/2023
S ³ [77]	③	②	②	-	①	-	07/2023
ChatDev [18]	①	②	②	②	①	②	07/2023
ToolLLM [14]	-	-	-	②	②	①	07/2023
MemoryBank [39]	-	②	②	-	①	-	07/2023
MetaGPT [23]	①	②	②	②	②	-	08/2023

4. Aplicações em cenários reais:

O texto aborda as aplicações de agentes autônomos baseados em LLM em três áreas principais: ciências sociais, ciências naturais e engenharia.

- Ciências Sociais: Agentes podem simular experimentos psicológicos, auxiliar na análise política e econômica, simular fenômenos sociais (como disseminação de desinformação), ajudar em processos legais e atuar como assistentes de pesquisa.
- Ciências Naturais: Eles são usados para gerenciar dados científicos, planejar e conduzir experimentos, e ajudar na educação científica, fornecendo assistência em tarefas complexas como a execução de experimentos ou solução de problemas matemáticos.
- Engenharia: Agentes baseados em LLM são aplicados no design e otimização de estruturas na engenharia civil, automatizam processos de codificação e testes de software, e assistem em áreas como design eletrônico

e segurança cibernética, colaborando com humanos para melhorar a eficiência e precisão em várias tarefas.

5. Avaliação:

A avaliação de agentes autônomos baseados em LLM é um desafio, assim como a avaliação dos próprios LLMs. Existem duas abordagens principais:

- **Avaliação Subjetiva:**

Esta abordagem envolve a avaliação feita por humanos. É útil quando não há métricas objetivas ou conjuntos de dados de referência, e geralmente avalia inteligência, usabilidade e comportamento humano dos agentes. As estratégias comuns incluem:

- **Anotação Humana:** Avaliadores classificam a saída dos agentes, comparando aspectos como inofensividade, honestidade e utilidade.
- **Teste de Turing:** Avaliadores tentam distinguir entre saídas geradas por humanos e agentes. Se não conseguirem, o agente está próximo de um desempenho humano.
- Embora seja eficaz, essa abordagem pode ser cara, demorada e sujeita a vieses. Para mitigar isso, pesquisadores estão explorando o uso dos próprios LLMs como avaliadores.

- **Avaliação Objetiva:**

Esta abordagem usa métricas quantitativas para medir o desempenho do agente de forma concreta e rastreável. As principais métricas incluem:

- **Métricas de sucesso de tarefa:** Mede a taxa de sucesso ou a precisão de uma tarefa.
- **Métricas de similaridade humana:** Avalia o quanto o comportamento do agente imita o dos humanos.
- **Métricas de eficiência:** Avaliam o tempo e os recursos necessários para o agente concluir suas tarefas.

Os protocolos de avaliação variam desde simulações de mundo real até avaliação social e testes de software. Vários benchmarks específicos foram desenvolvidos para testar agentes em diferentes contextos, como Minecraft, simulações de interação social, e testes de penetração.

6. Limitações:

O campo dos agentes autônomos baseados em LLMs (modelos de linguagem de grande escala) tem obtido avanços significativos, mas ainda enfrenta desafios importantes:

- **Capacidade de Representação de Papéis:** Agentes precisam desempenhar diferentes papéis (como programador ou químico), mas os LLMs podem ter dificuldade em representar papéis raros ou emergentes. Ajuste fino dos modelos ou prompts específicos pode ajudar, mas encontrar as melhores soluções é complexo.
- **Alinhamento com Valores Humanos:** Agentes devem se alinhar com valores humanos diversos, inclusive valores negativos em simulações, para refletir melhor a realidade. Realinhar os LLMs para diferentes cenários é um desafio, pois eles geralmente seguem valores humanos unificados.
- **Robustez de Prompts:** Pequenas mudanças em prompts podem gerar resultados muito diferentes, especialmente quando os agentes usam módulos como memória e planejamento. Desenvolver uma estrutura de prompts resiliente e unificada para vários LLMs é uma tarefa difícil.
- **Alucinação:** LLMs podem gerar informações falsas, o que é problemático em agentes autônomos, especialmente em tarefas como geração de código. Incorporar feedback humano pode ajudar a mitigar esse problema.
- **Limite do Conhecimento:** LLMs possuem muito mais conhecimento do que um ser humano médio, o que pode afetar simulações. Limitar o conhecimento dos LLMs para simular melhor o comportamento humano é crucial.
- **Eficiência:** A velocidade de inferência dos LLMs é lenta, o que afeta a eficiência dos agentes, especialmente quando várias consultas são necessárias para ações.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 26 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Nesta segunda semana, como foi proposto, foquei em estudar e me aprofundar em conceitos relacionados a Agentes e Multiagentes baseados em LLMs, formas de relacionamento, aplicação em simulações de mundo, resolução de problemas e entre outros.

Para tal, foi feita a leitura e exploração do survey: [Large Language Model based Multi-Agents: A Survey of Progress and Challenges](#)

- Resumo do artigo: [Large Language Model based Multi-Agents](#)

Também foi explorado alguns artigos/frameworks para o entendimento e alinhamento com conceitos apresentados nos artigos lidos anteriormente:

- MetaGPT: [The Multi-Agent Framework](#)
- ChatDev: [Scaling Large-Language-Model-based Multi-Agent Collaboration](#)
- Generative agents: [Generative Agents: Interactive Simulacra of Human Behavior](#)
- Resumo dos artigos: [MetaGPT, ChatDev e Generative agents](#)

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

Como todos os estudos foram feitos e divulgados até abril de 2024, para a próxima semana planejo buscar o estado da arte tanto para agentes quanto para multiagentes considerando aspectos como:

- Aplicação à solução de problemas.
- Simulação de mundo.
- Capacidades dos agentes.

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

ACEITE DA ENTREGA:

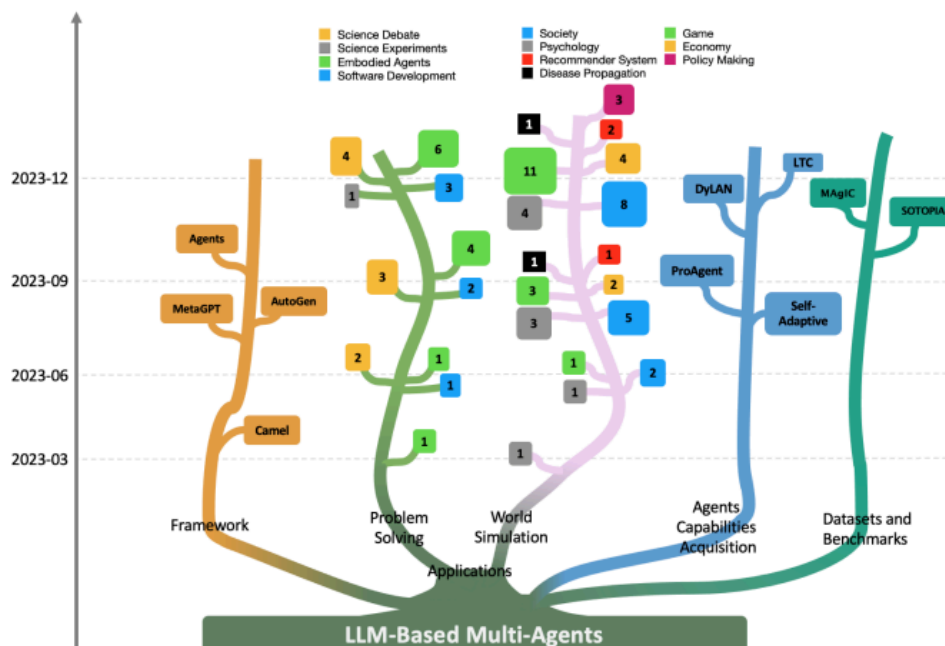
CEDRIC LUIZ DE CARVALHO: Go! ▾

Large Language Model based Multi-Agents: A Survey of Progress and Challenges

<https://arxiv.org/pdf/2402.01680>

“Nosso objetivo é que os leitores obtenham insights substanciais sobre as seguintes questões: Quais domínios e ambientes os multiagentes baseados em LLM simulam? Como esses agentes são perfilados e como eles se comunicam? Quais mecanismos contribuem para o crescimento das capacidades dos agentes? Para aqueles interessados em mergulhar neste campo de estudo, também resumimos os conjuntos de dados ou benchmarks comumente usados para que possam ter acesso conveniente.”

Eles também criaram um [repositório](#) para guardar todos os artigos e novidades em relação a agentes baseados em LLMs. (Aparenta estar um pouco desatualizado mas há muitos conteúdos interessantes.)



Uma coisa que eu achei interessante e que é mostrado tanto no texto quanto na imagem é:

“Na Seção 4, categorizamos as aplicações atuais em duas correntes principais: multiagentes para resolução de problemas e multiagentes para simulação do mundo.”

Há várias aplicações para a resolução de problemas em diversos campos e a simulação de mundo em diversos ambientes e cenários diferentes.

O texto também compara esses sistemas com sistemas multiagentes, onde vários agentes colaboram para resolver tarefas dinâmicas e complexas. Cada agente tem comportamentos e estratégias únicas, e juntos, eles tomam decisões em grupo, o que aumenta a eficácia em situações mais complicadas.

Há tipos de ambientes em que os agentes interagem:

O **Sandbox** refere-se a um ambiente simulado ou virtual construído por humanos, onde os agentes podem interagir mais livremente e experimentar várias ações e estratégias. Esse tipo de interface é amplamente utilizado no desenvolvimento de software (interpretador de código como ambiente simulado), jogos (usando regras do jogo como ambiente simulado), etc.

O **Físico** é um ambiente do mundo real onde os agentes interagem com entidades físicas e obedecem às leis da física e às restrições do mundo real. No espaço físico, os agentes normalmente precisam tomar ações que possam ter resultados físicos diretos. Por exemplo, em tarefas como varrer o chão, fazer sanduíches, embalar compras e organizar armários, agentes robóticos são obrigados a realizar ações iterativamente, observar o ambiente físico e refinar continuamente suas ações.

Por fim, **Nenhum** refere-se a cenários onde não há um ambiente externo específico, e os agentes não interagem com nenhum ambiente. Por exemplo, muitas aplicações utilizam múltiplos agentes para debater uma questão até chegar a um consenso. Essas aplicações concentram-se principalmente na comunicação entre os agentes e não dependem de um ambiente externo.

A comunicação entre agentes em sistemas multiagentes baseados em LLMs é essencial para formar uma inteligência coletiva eficaz. Ela pode ser analisada em três aspectos principais: **Paradigmas de Comunicação, Estrutura de Comunicação e Conteúdo da Comunicação.**

Nos Paradigmas de Comunicação, existem três principais formas de interação entre os agentes:

- Cooperativo: agentes trabalham juntos para atingir uma meta comum, trocando informações para melhorar a solução conjunta.
- Debate: agentes discutem suas opiniões, defendendo seus pontos de vista e criticando os de outros, buscando chegar a um consenso.
- Competitivo: cada agente busca alcançar seus próprios objetivos, que podem entrar em conflito com os dos outros.

A Estrutura de Comunicação varia entre diferentes tipos:

- Camadas: agentes se organizam de forma hierárquica, interagindo dentro de suas camadas ou com camadas próximas, como na rede DyLAN, que otimiza a cooperação com seleção dinâmica de agentes.
- Descentralizada: os agentes se comunicam diretamente entre si em uma rede peer-to-peer, usada em simulações de mundo aberto.
- Centralizada: existe um agente central que coordena toda a comunicação do sistema.
- Pool de Mensagens Compartilhadas: como no MetaGPT, onde as mensagens são publicadas e os agentes escolhem quais informações seguir, otimizando o fluxo de comunicação.

LLM-MA para Resolução de Problemas

Estes sistemas são usados para abordar desafios complexos que exigem colaboração de especialistas em diferentes áreas. Exemplos de aplicação incluem:

- Desenvolvimento de Software: Vários agentes, cada um representando diferentes papéis como programadores ou gerentes de projeto, colaboram para gerar e refinar código de forma autônoma, utilizando frameworks como "role-play" e procedimentos operacionais padrão. Essa colaboração permite melhorar o desempenho e aumentar a eficácia do processo de desenvolvimento de software.

LLM-MA em Simulações de Mundo

Os sistemas LLM-MA são amplamente utilizados para criar ambientes simulados que imitam interações humanas e permitem testar hipóteses complexas, como em:

- Jogos: Agentes assumem papéis em jogos de comunicação e estratégia, como Werewolf e Avalon, usando linguagem natural para explorar cooperação, engano e liderança. Esses ambientes são valiosos para estudar comportamento social e a teoria dos jogos.

- Psicologia: Utilizados para simular interações sociais e comportamentos humanos, esses agentes ajudam a explorar questões como conformidade, padrões de pensamento e comportamento de grupo, replicando experimentos psicológicos.
- Economia: Agentes são empregados em simulações econômicas, atuando como modelos de "homo economicus" para explorar mercados, negociações financeiras e dinâmicas de cidades virtuais, oferecendo novas perspectivas sobre a tomada de decisão em contextos econômicos.

Frameworks Multiagentes

Três principais frameworks são apresentados:

- MetaGPT: Foca na emulação de processos humanos e usa Procedimentos Operacionais Padrão (SOPs) para atribuir papéis aos agentes.
- CAMEL: Facilita a cooperação autônoma entre agentes usando a técnica inception prompting.
- Autogen: Versátil e personalizável, permite a criação de agentes com linguagem natural e código, aplicável em várias áreas.

Conjuntos de Dados e Benchmarks

Os sistemas LLM-MA utilizam conjuntos de dados e benchmarks para medir desempenho em diferentes cenários, como resolução de problemas e simulação de mundos. No entanto, há uma carência de benchmarks abrangentes para aplicações mais complexas, como simulações econômicas e científicas.

Desafios e Oportunidades

Diversos desafios são identificados:

- Ambientes Multimodais: A maioria dos sistemas ainda é baseada em texto, e há uma necessidade de expandir para ambientes que incluam imagens, vídeos e ações físicas.
- Alucinações: O fenômeno em que os LLMs geram informações incorretas é agravado em sistemas multiagentes, exigindo melhores métodos de detecção e correção.
- Inteligência Coletiva: Sistemas LLM-MA precisam melhorar a colaboração entre agentes para alcançar uma verdadeira inteligência coletiva.
- Escalabilidade: Aumentar o número de agentes envolve desafios de coordenação e comunicação, o que exige técnicas eficazes de orquestração.
- Benchmarks: Ainda há necessidade de melhores benchmarks para avaliar os comportamentos emergentes em sistemas multiagentes.

Conclusão

O artigo conclui que os sistemas LLM-MA estão progredindo rapidamente e possuem grande potencial em áreas como finanças, educação e planejamento urbano. Entretanto, desafios como alucinações e escalabilidade ainda precisam ser superados. A pesquisa contínua e o desenvolvimento de melhores ferramentas e benchmarks ajudarão a maximizar o potencial desses sistemas.

MetaGPT, ChatDev e Generative agents

[The Multi-Agent Framework](#)

[Scaling Large-Language-Model-based Multi-Agent Collaboration](#)

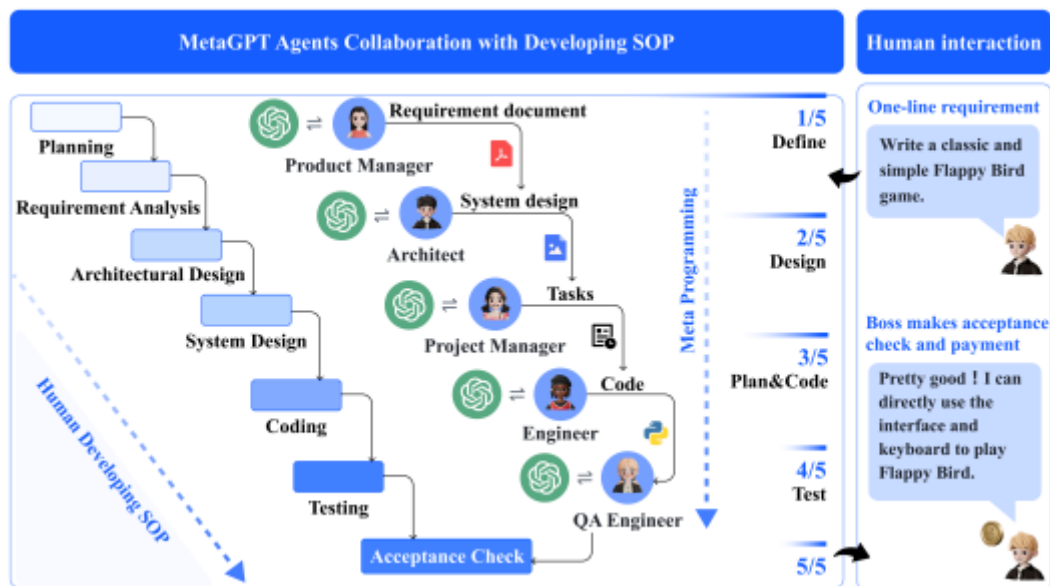
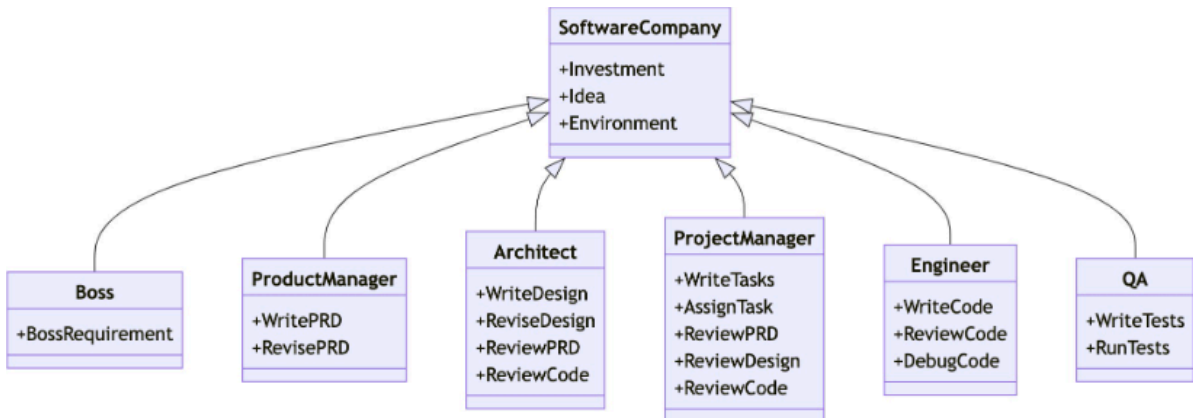
[Generative Agents: Interactive Simulacra of Human Behavior](#)

MetaGPT:

O primeiro artigo descreve o MetaGPT, um framework de metaprogramação que facilita a colaboração entre agentes autônomos baseados em LLMs para solucionar problemas complexos. Este framework é projetado para superar as limitações de sistemas multiagentes anteriores, que sofrem com inconsistências lógicas e alucinações devido à comunicação ingênua entre agentes. O MetaGPT utiliza **Procedimentos Operacionais Padrão (SOPs)** para estruturar fluxos de trabalho, de forma semelhante à dinâmica de colaboração humana, onde agentes especializados verificam resultados intermediários, diminuindo erros e garantindo soluções mais coerentes.

O framework adota uma abordagem de "linha de montagem", atribuindo funções claras e bem definidas a cada agente, o que permite a decomposição eficiente de tarefas complexas. Ele se destaca em benchmarks de geração de código, como HumanEval e MBPP, atingindo um novo estado da arte. Em comparação com outras ferramentas como AutoGPT e LangChain, o MetaGPT demonstra maior eficiência e robustez ao lidar com projetos de software complexos, alcançando uma taxa de conclusão de tarefas de 100%.

O MetaGPT incorpora SOPs para garantir que os agentes gerem saídas estruturadas, como documentos de design e especificações de interface, o que melhora a qualidade e precisão dos resultados. Além disso, introduz um mecanismo de feedback executivo que permite a depuração e execução de código em tempo de execução. Essa abordagem inovadora de metaprogramação torna o MetaGPT uma solução promissora para o desenvolvimento de sistemas multiagentes baseados em LLMs, superando as deficiências de frameworks anteriores e oferecendo uma plataforma mais organizada e eficiente para geração de código e desenvolvimento de software colaborativo.



Planejamento e Análise de Requisitos:

- Gerente de Produto: Inicia o processo definindo o documento de requisitos, o que desencadeia a análise de requisitos.

Design Arquitetônico e de Sistema:

- Arquiteto: Baseado nos requisitos, o arquiteto projeta o sistema.
- Gerente de Projeto: Com o design pronto, o gerente de projeto transforma o design em tarefas específicas para serem executadas.

Codificação e Teste:

- Engenheiro: Recebe as tarefas e escreve o código conforme especificado.
- Engenheiro de QA (Garantia de Qualidade): Testa o código para garantir que ele atenda aos requisitos estabelecidos e funcione corretamente.

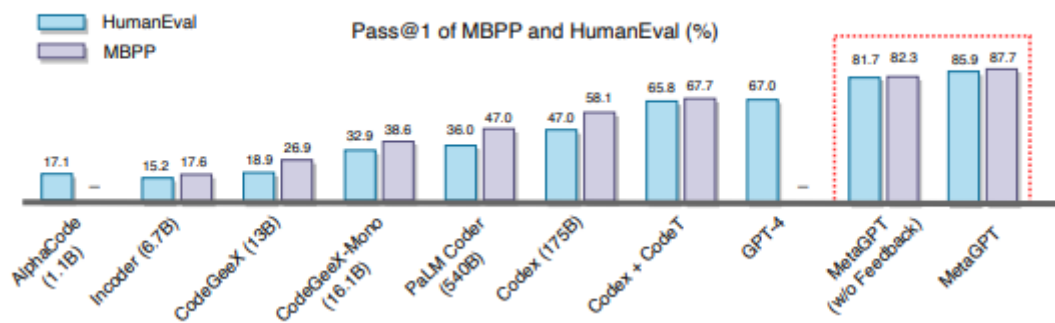
Verificação e Aceitação:

- O processo termina com uma verificação de aceitação, onde o trabalho realizado é avaliado para garantir que tudo foi completado conforme os padrões e requisitos iniciais.

Interação Humana:

- Há etapas no processo onde a interação humana é necessária para aprovações e ajustes, como definir os requisitos iniciais ou a aprovação final pelo chefe.

4.2 MAIN RESULT



Há muitas outras coisas a serem faladas e testes a serem realizados com o MetaGPT. Há a comparação entre a existência ou não de certos perfis no processo, resultados, várias explicações em relação ao fluxo e etc.

Table 3: Ablation study on roles. ‘#’ denotes ‘The number of’, ‘Product’ denotes ‘Product manager’, and ‘Project’ denotes ‘Project manager’. ‘✓’ indicates the addition of a specific role. ‘Revisions’ refers to ‘Human Revision Cost’.

Engineer	Product	Architect	Project	#Agents	#Lines	Expense	Revisions	Executability
✓	✗	✗	✗	1	83.0	\$ 0.915	10	1.0
✓	✓	✗	✗	2	112.0	\$ 1.059	6.5	2.0
✓	✓	✓	✗	3	143.0	\$ 1.204	4.0	2.5
✓	✓	✗	✓	3	205.0	\$ 1.251	3.5	2.0
✓	✓	✓	✓	4	191.0	\$ 1.385	2.5	4.0

ChatDev:

É uma abordagem inovadora baseada em Large Language Model (LLM) que visa revolucionar o campo de desenvolvimento de software. Este paradigma busca eliminar a necessidade de modelos especializados durante cada fase do processo de desenvolvimento. A estrutura ChatDev aproveita os recursos das estruturas LLM, utilizando comunicação em linguagem natural para unificar e agilizar os principais processos de desenvolvimento de software.

Nessa, é gerado uma interface semelhante ao primeiro diagrama do MetaGPT para o desenvolvimento de software dividido em 4 etapas:

Projetar, Codificar, Testar, Documentar.

Cada um desses estágios implanta uma equipe de agentes virtuais, como programadores de código ou testadores, que colaboram entre si por meio de diálogos que resultam em um fluxo de trabalho contínuo. Além disso, o ChatDev utiliza uma cadeia de chat que funciona como facilitador e divide cada etapa do processo de desenvolvimento em subtarefas atômicas, possibilitando assim dupla função, permitindo propostas e validação de soluções usando comunicações sensíveis ao contexto que permitem aos desenvolvedores efetivamente resolver as subtarefas especificadas. A cadeia de bate-papo consiste em vários nós, onde cada nó individual representa uma subtarefa específica, e essas duas funções se envolvem em discussões multivoltas com base no contexto para não apenas propor, mas também validar as soluções.

A ideia é bastante semelhante ao Meta GPT mas com algumas diferenças.

Fluxo de memória

- O fluxo de memória é um mecanismo usado pela estrutura ChatDev que mantém um registro conversacional abrangente dos diálogos anteriores de um agente e auxilia no processo de tomada de decisão que se segue de maneira consciente da expressão. A estrutura ChatDev usa prompts para estabelecer os protocolos de comunicação necessários. Por exemplo, quando as partes envolvidas chegam a um consenso, uma mensagem final que satisfaça um requisito de formatação específico como (: Aplicativo de desktop”). Para garantir a conformidade com o formato designado, o quadro monitoriza continuamente e, finalmente, permite que o diálogo actual chegue a uma conclusão.

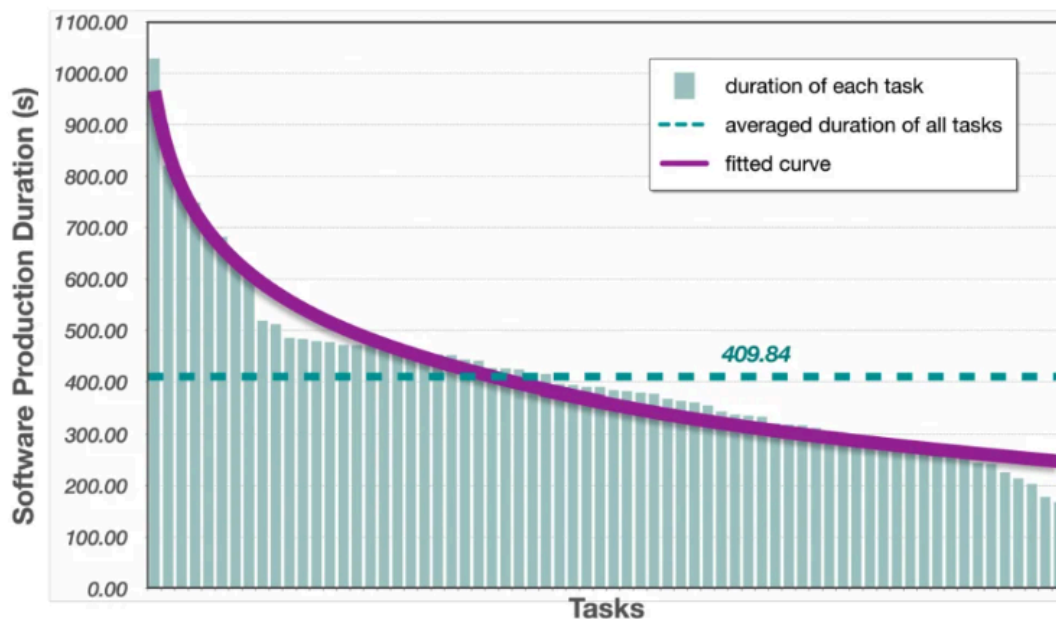
Auto-reflexão

- Os desenvolvedores da estrutura ChatDev observaram situações em que ambas as partes envolvidas chegaram a um consenso mútuo, mas os protocolos de comunicação predefinidos não foram acionados. Para resolver estas questões, o A

estrutura ChatDev introduz um mecanismo de autorreflexão que auxilia na recuperação e extração de memórias. Para implementar o mecanismo de autorreflexão, a estrutura ChatDev inicia um bate-papo novo e fresco, recrutando um “pseudo eu” como um novo questionador. O “pseudo eu” analisa os diálogos anteriores e registros históricos e informa o assistente atual, após o que solicita um resumo de informações conclusivas e dignas de ação, conforme demonstrado na figura abaixo.

Resultados:

	Min	Max	Avg.
# Code Files	2.00	8.00	4.26
# Asset Files	0.00	21.00	8.74
# Document Files	4.00	5.00	4.04
# Lines of Source Codes	39.00	359.00	131.61
# Lines of Dependencies	1.00	5.00	2.90
# Lines of User Manual	31.00	232.00	53.96
# Version Updates	5.00	42.00	13.23
# Software Re-development	1.00	5.00	1.40



Generative Agents:

O artigo explora a criação de "agentes generativos" — agentes que simulam comportamentos humanos usando modelos generativos. Esses agentes lembram, refletem e planejam com base em experiências passadas e interagem de forma realista com o ambiente e outros agentes. A arquitetura proposta inclui três componentes principais: uma memória de longo prazo para armazenar experiências, um módulo de reflexão para gerar inferências e um sistema de planejamento para ações detalhadas. A pesquisa demonstra como esses agentes podem ser usados em jogos e simulações sociais, como em uma sociedade virtual inspirada por The Sims.

Este não é focado em desenvolvimento de software e sim em sociedade.

Smallville é habitado por 25 agentes, cada um com uma identidade e relacionamentos descritos por memórias iniciais. Os agentes se comunicam entre si em linguagem natural, realizando ações que são traduzidas para emojis no ambiente. Usuários podem interagir com os agentes, comandando-os diretamente ou influenciando o ambiente ao seu redor, como mudar o status de objetos. A arquitetura dos agentes utiliza percepções do ambiente para salvar memórias, fazer planos e gerar reflexões de longo prazo, garantindo comportamentos críveis e coerentes.

O estudo sugere melhorias na arquitetura, como aprimorar o módulo de recuperação de informações e tornar a simulação mais eficiente. No entanto, simular 25 agentes por dois dias foi caro e demorado, destacando a necessidade de otimizar a interatividade em tempo real. Avaliações mais longas e rigorosas dos comportamentos dos agentes são sugeridas, assim como o teste da robustez contra falhas como "hacking de memória" e alucinações. Além disso, os agentes podem herdar preconceitos dos modelos de linguagem subjacentes, especialmente em relação a populações marginalizadas, exigindo melhorias nos modelos de base.

Ética e Impacto Social: Agentes generativos trazem novos desafios éticos. Um risco é a formação de relações parasociais inadequadas, com usuários atribuindo emoções humanas aos agentes. Para mitigar isso, os agentes devem deixar claro que são entidades computacionais e alinhar seus comportamentos com valores apropriados. Outro risco é o impacto de erros de inferência, especialmente fora de ambientes controlados como jogos. Também existe o perigo de abuso de IA generativa para deepfakes ou desinformação, o que pode ser mitigado com auditorias de entradas e saídas geradas. Além disso, os agentes não devem substituir o papel de humanos em processos de design, mas servir como ferramentas de prototipagem.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 2 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás


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

Essa Semana, concentrei-me em um trabalho investigativo, elencando as principais ferramentas e estudos relacionados a Agentes e Sistemas Multiagentes baseados em Grandes Modelos de Linguagem, considerados como 'estado da arte'. A investigação focou em aspectos como: aplicação na solução de problemas, simulação de mundo e métodos de avaliação.


Foram considerados artigos com mais de 50 citações, repositórios com mais de 200 estrelas e artigos aceitos por conferências ou periódicos de renome.

Alguns dos repositórios utilizados na pesquisa foram:

[LLM Agents Papers](#), [LLM-Agents-Papers](#), [Awesome LLM-Powered Agent](#)

Também me baseei nos artigos lidos para a outra entrega e elaborei uma árvore com esses tópicos mapeados  ou [Canva](#).

Nesse processo, encontrei um blog de um professor de engenharia da computação, Yu Su, que comenta sobre a notoriedade desse tópico, passando pela conceitualização de Stuart Russell sobre agentes.

 Agentes

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

Para a próxima entrega, pretendo focar na pesquisa sobre a aplicação de agentes baseados em LLMs em cenários de "Simulating Financial Market" (Simulação de Mercado Financeiro).

Artigo de inspiração: [Simulating Financial Market via Large Language Model based Agents](#)

Iniciar testes e implementações básicas seguindo os módulos propostos no [primeiro artigo](#) lido para o Gate (perfil, memória, planejamento e ação).

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

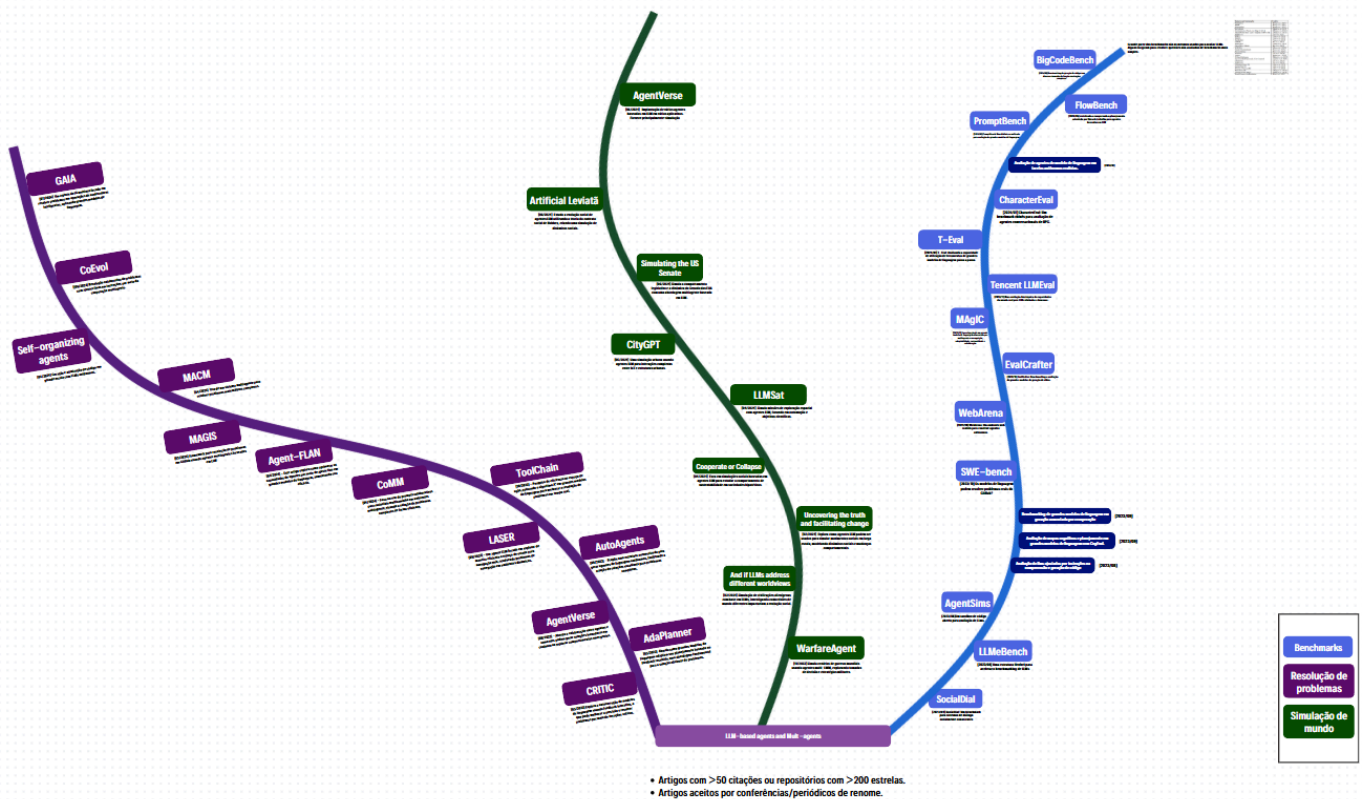
Substituí a categorização de 'Capacidade dos agentes' por 'Métodos de Avaliação', pois o critério de capacidade está intrínseco nos outros tópicos mapeados.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Tree for Agents and Multi-Agents

A imagem representa o resultado dessa categorização, mostrando a organização e os tópicos principais identificados. A pesquisa utilizou como critérios a seleção de artigos com mais de 50 citações, repositórios com mais de 200 estrelas no GitHub e publicações aceitas por conferências ou periódicos de renome.



Agentes de linguagem: Um passo evolutivo crítico da inteligência artificial

<https://yusu.substack.com/p/language-agents>

Esse foi um post em um blog feito por Yu Su, Professor de engenharia na OSU, Codiretor do grupo OSU NLP no final de 2023. Há muitos trabalhos citados e muitos comentários sobre o que ele escreveu.

O conceito de agente foi introduzido na IA desde o seu início. É um dos primeiros conceitos que ensinamos em IA 101 (por exemplo, do livro clássico de Stuart Russell e Peter Norvig)

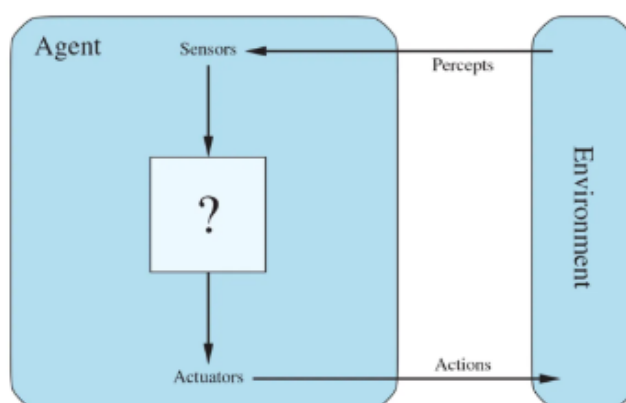
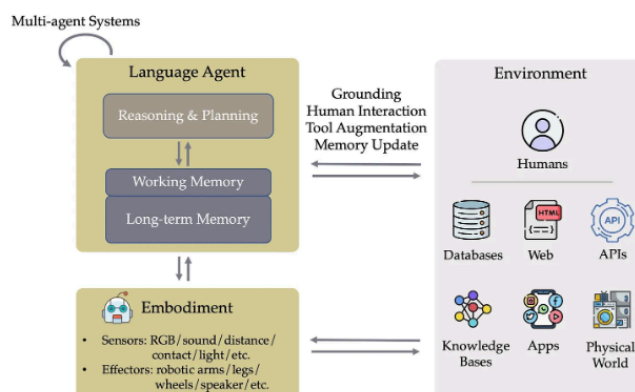


Figura 1: Ilustração esquemática de um agente clássico (Russell & Norvig, 2020, Cap. 2).

A mudança mais fundamental é a capacidade de usar a linguagem. Os agentes de IA contemporâneos usam a linguagem como um veículo para pensamento e comunicação, uma característica que era única aos humanos antes desses agentes artificiais. Isso expande dramaticamente a amplitude e a profundidade dos problemas que esses agentes podem possivelmente enfrentar, de forma autônoma.

Uma possível estrutura conceitual para agentes de linguagem:



No nível superior, temos os mesmos componentes de um agente de IA clássico — há um agente, opcionalmente com incorporação, que interage com algum ambiente. Agentes (incluindo agentes humanos) podem cooperar ou competir entre si em um sistema multiagente. As verdadeiras diferenças, no entanto, estão dentro de cada componente e como eles interagem entre si.

Agente: Um agente de linguagem é equipado com um mecanismo de memória qualitativamente diferente dos agentes anteriores. Ele inclui tanto a memória de longo prazo quanto a memória de trabalho. A memória de longo prazo é principalmente os parâmetros pré-treinados do LLM, que captura, entre outras coisas, uma representação linguística distribuída adquirida por meio da compressão dos dados de pré-treinamento. O contexto do LLM e a capacidade de aprendizagem emergente no contexto se assemelham à memória de trabalho humana de várias maneiras: o contexto serve como um bloco de notas para reter temporariamente informações relacionadas à tarefa, e o processamento e o raciocínio sobre as informações são controlados por outra unidade de atenção.

Ambiente: Os primeiros agentes de IA são geralmente projetados e programados para um ambiente específico, e o ambiente é tipicamente simulado ou significativamente simplificado para se ajustar aos limites dos paradigmas de modelagem no momento. Um agente de linguagem, por outro lado, pode interagir simultaneamente com muitos ambientes do mundo real de natureza heterogênea sem ter especificações de ambiente codificadas a priori. Por exemplo, para resolver uma tarefa complexa como “como meu tempo médio de sono semanal muda nos últimos 6 meses”, um agente de linguagem pode, de forma autônoma, primeiro obter os dados brutos pertinentes de um banco de dados via SQL, calcular a média semanal escrevendo e executando um programa Python por meio de um interpretador de código e visualizar os resultados por meio de um interpretador de código ou Excel. O que é notavelmente diferente é que agora é um agente de linguagem, não um humano, que faz a ponte entre esses ambientes heterogêneos: o agente pode descobrir como escolher e interagir com os ambientes certos, reconciliar os formatos e protocolos heterogêneos (por

exemplo, JSON vs. XML) e orquestrar as percepções ambientais com seus próprios passos de raciocínio para chegar a uma solução coerente.

O autor também destaca alguns pontos que foram mapeados anteriormente (em outros trabalhos) como tipos de memória, ferramentas, capacidade dos modelos de linguagem e etc.

Raciocínio e planejamento:

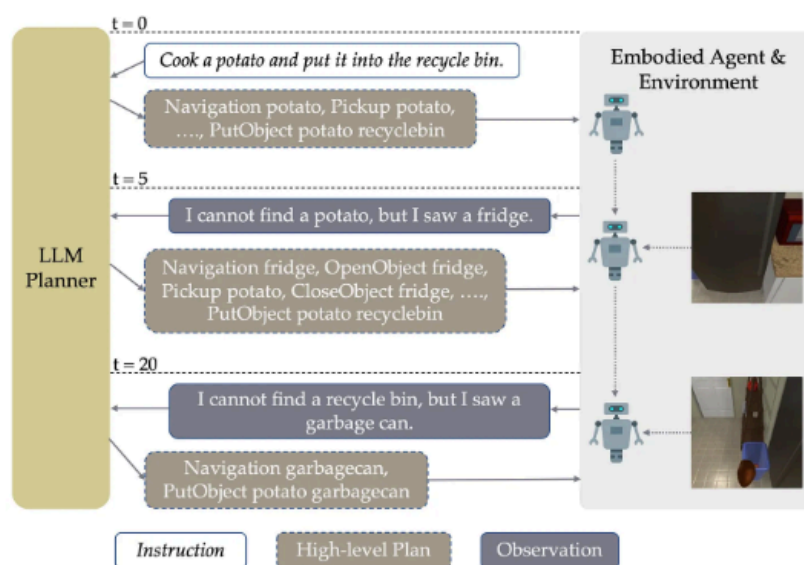


Figura 5. Um exemplo de Song et al. (2023) onde um agente de linguagem incorporado alimentado por um LLM integra raciocínio e planejamento orientados pela linguagem.

A capacidade de usar a linguagem como um veículo para o pensamento é uma mudança profunda para a IA. Já estamos vendo novos algoritmos de raciocínio, por exemplo, cadeia de pensamento (Wei et al., 2022) e árvore de pensamento (Yao et al., 2023 , Long, 2023), que começam a liberar parte desse poder. À medida que nossa compreensão sobre isso progride, tenho certeza de que veremos mais algoritmos de raciocínio que aproveitarão melhor essa nova capacidade.

Destaca-se que a incorporação pode tornar-se essencial para agentes de linguagem, permitindo que aprendam de forma mais eficiente por meio de uma experiência em primeira pessoa, aproximando-se de como os humanos aprendem. Trabalhos recentes, como SayCan, LLM-Planner, Voyager e RT-2, mostram avanços significativos nesse campo.

O autor também discute como os sistemas multiagentes, baseados em agentes de linguagem, podem interagir de forma mais autônoma e evoluir com o uso e comunicação, oferecendo novas oportunidades e desafios. No entanto, os riscos desses agentes são sérios, pois eles ampliam problemas existentes, como vieses e privacidade, e trazem novos riscos, como ações irreversíveis. A mitigação desses riscos exige esforços coordenados entre pesquisadores, profissionais e reguladores. Apesar disso, o autor se mantém cautelosamente otimista quanto ao futuro desses agentes.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 10 de out. de 2024



Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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


Nesta Semana, concentrei-me em um trabalho de pesquisa sobre o tema proposto, aplicado ao mercado financeiro, em paralelo com uma implementação inicial de agentes de forma isolada, sem utilizar nenhuma ferramenta que abstraísse partes do código. O objetivo é entender de forma prática tudo o que foi estudado e levantado até agora sobre a [estrutura básica](#) existente.

Implementação inicial:

- Exemplo de iteração com o agente desenvolvido:  iteracao_inicial.png
- Código com a implementação comentada: [agent_without_framework.ipynb](#)
- Pasta com tudo o que é necessário para executar o código:  inicial_agents

Pesquisa:

A pesquisa realizada em paralelo foi relacionada à Simulação de Mercado Financeiro comparando diferentes tipos de perfis de investidores frente a políticas e notícias sobre o mercado, com 11 tipos de empresas chinesas listadas na Bolsa de valores.

- Resumo do artigo:  Simulating_Financial_Market

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

Continuar a pesquisa e testes no mesmo tema, aplicado ao mercado financeiro mas baseado em uma abordagem ligeiramente diferente proposta recentemente (21/09/2024).

When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments

- Repositório: [GitHub](#)
- Artigo: [arxiv](#)

Iniciar testes e implementações básicas seguindo os módulos de perfil, memória, planejamento e ação porém, aplicados a interações multi-agentes.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

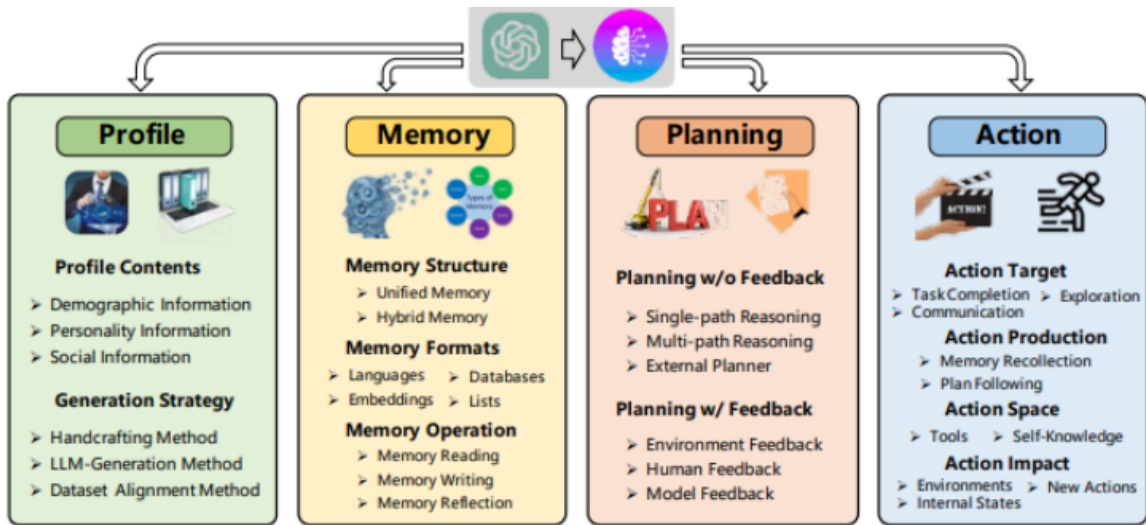
Implementação inicial - Notebook:

O artigo [A Survey on Large Language Model based Autonomous Agents](#), usado como base nos dois primeiros gates, propõe uma estrutura padrão compartilhada por diversos frameworks e que dão base ao conceito de construção de agentes baseados em LLMs. A ideia é explorar conceitos fundamentais como perfilamento, memória, planejamento e ação e alguns subconceitos existentes nestes a fim de entender melhor a estrutura antes de usarmos frameworks que abstraem grande parte do processo.

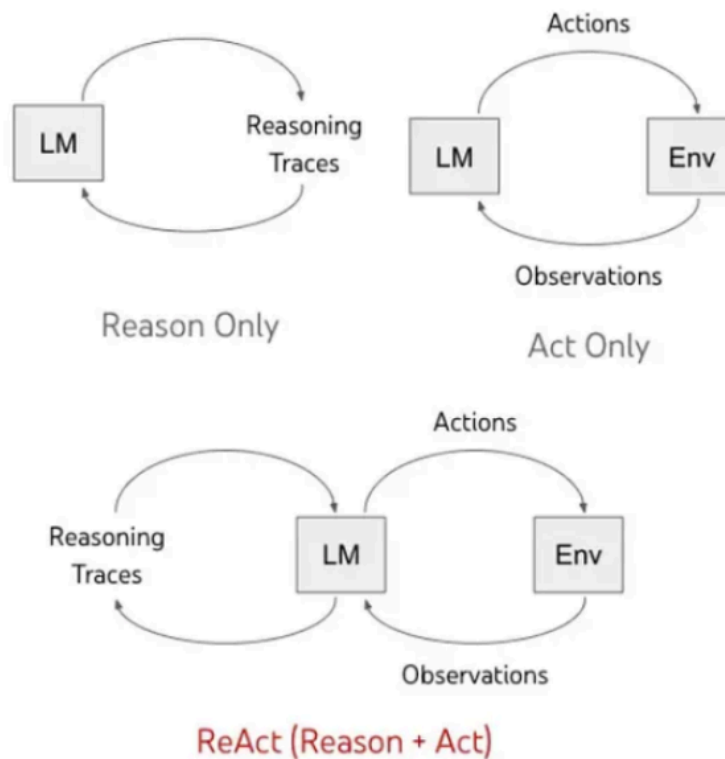
Com isso, implementei um agente no formato ReAct, que é capaz de planejar, chamar funções e agir conforme o input do usuário mostre necessário.

A implementação contém um teste inicial bem simples e explora:

- 1 - Perfilamento feito manualmente (instruções geradas por mim);
- 2 - Estrutura de memória híbrida e em formato de lista com todas as operações sendo realizadas;
- 3 - Planejamento com caminho único e sem feedback;
- 4 - Alvo da ação em completar a tarefa, com reflexão de memória e uso de ferramentas (internas e externas);



ReAct vem de Reasoning and Act

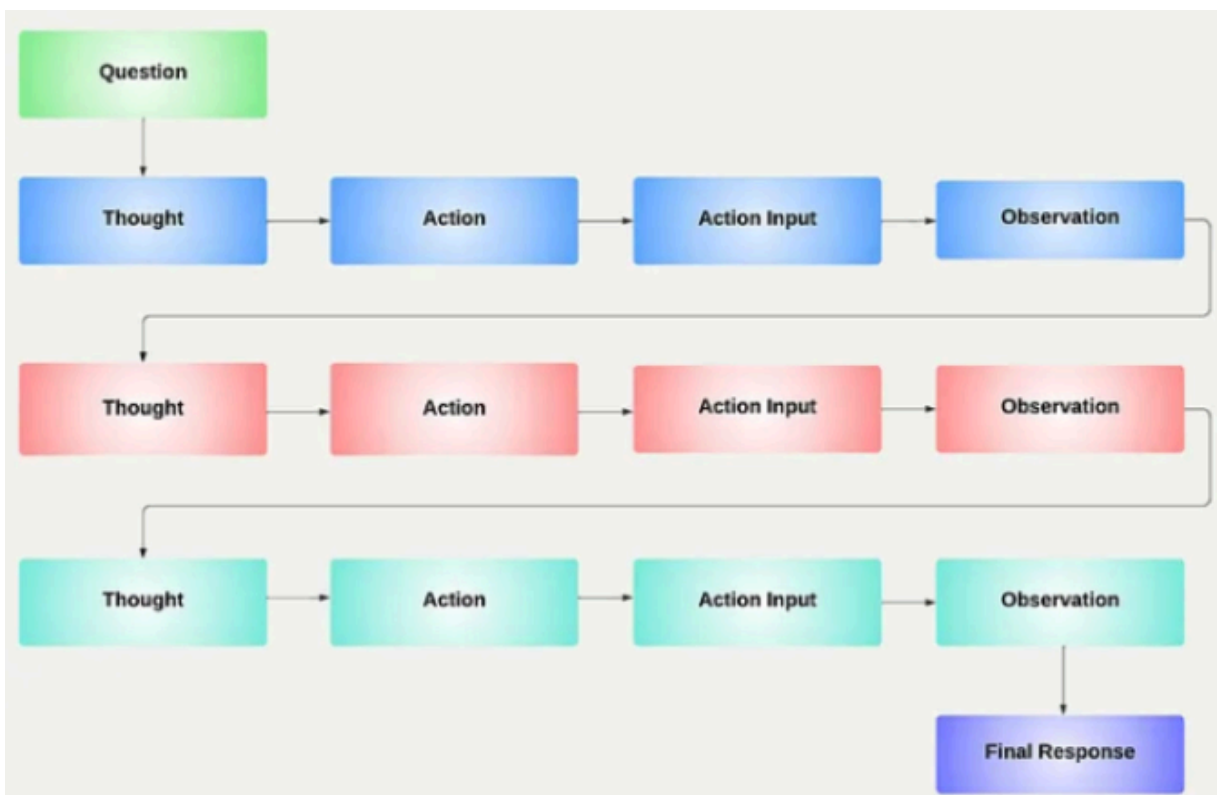


É uma estrutura para solicitar grandes modelos de linguagem (LLMs) em tarefas que exigem raciocínio e/ou ação explícitos.

Foi introduzido pela primeira vez no artigo [ReAct: Synergizing Reasoning and Acting in Language Models](#) em outubro de 2022. A estrutura foi desenvolvida para sinergizar o raciocínio e a tomada de ação em modelos de linguagem, tornando-os mais capazes, versáteis e interpretáveis.

Ao intercalar raciocínio e ação, o ReACT permite que os agentes alternem entre gerar pensamentos e ações específicas da tarefa dinamicamente.

O modelo ReAct emprega um loop pensamento-ação-observação, onde o agente raciocina sobre observações anteriores para decidir sobre ações. Esse processo iterativo permite que ele adapte e refine sua abordagem com base nos resultados de suas ações.



Instalar bibliotecas requeridas

```
%pip install -q -r requirements.txt
```

Note: you may need to restart the kernel to use updated packages.

Importar bibliotecas necessárias e a chave da OpenAI

```
from dotenv import load_dotenv
import os
from openai import OpenAI
import re
```

Testando chamada de API e modelo

```
client = OpenAI()

completion = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[{"role": "user", "content": "Explique a ideia da residência
ao fim de uma graduação."}],
    temperature=0
)

print(completion.choices[0].message.content)
```

A residência ao fim de uma graduação é um programa de formação prática que permite aos graduados, especialmente nas áreas da saúde, como medicina, enfermagem, psicologia, entre outras, aprofundar seus conhecimentos e habilidades em um ambiente profissional. A ideia central da residência é proporcionar uma experiência de aprendizado intensivo, onde os recém-formados podem aplicar a teoria adquirida durante a graduação em situações reais de trabalho, sob a supervisão de profissionais experientes.

Principais características da residência:

- Prática Supervisionada**: Os residentes trabalham em hospitais, clínicas ou outras instituições de saúde, onde são supervisionados por profissionais qualificados. Isso garante que eles recebam orientação e feedback enquanto desenvolvem suas habilidades.
- Duração**: Os programas de residência podem variar em duração, geralmente durando de um a três anos, dependendo da especialidade e do país.
- Especialização**: A residência permite que os graduados se especializem em uma área específica de sua profissão, como cirurgia, pediatria, psiquiatria, entre outras, o que é fundamental para o desenvolvimento de competências avançadas.
- Integração Teórica e Prática**: Durante a residência, os participantes

têm a oportunidade de integrar o conhecimento teórico adquirido na graduação com a prática clínica, o que é essencial para a formação de profissionais competentes.

5. ****Desenvolvimento Profissional****: Além de habilidades técnicas, a residência também promove o desenvolvimento de habilidades interpessoais, como comunicação, trabalho em equipe e tomada de decisões, que são cruciais para o exercício da profissão.

6. ****Certificação e Registro****: Em muitos países, a conclusão de um programa de residência é um requisito para a obtenção de licenças ou certificações profissionais, permitindo que os graduados atuem de forma independente em suas áreas.

Importância da Residência:

A residência é fundamental para garantir que os profissionais estejam bem preparados para enfrentar os desafios do mercado de trabalho e para oferecer um atendimento de qualidade à população. Ela também contribui para a formação de uma força de trabalho qualificada e competente, capaz de lidar com as complexidades da prática profissional.

Criando a classe de agente

```
class Agent:
    def __init__(self, client, system):
        self.client = client
        self.system = system
        self.messages: list = []

        if self.system:
            self.messages.append({"role": "system", "content": system})

    def __call__(self, message=""):
        if message:
            self.messages.append({"role": "user", "content": message})
            result = self.execute()
            self.messages.append({"role": "assistant", "content": result})

        return result

    def execute(self):
        completion = client.chat.completions.create(
            model="gpt-4o-mini", messages=self.messages
```

```
)  
    return completion.choices[0].message.content
```

Instruindo modelo. Perfilando o mesmo

```
system_prompt = """
```

Você executa em um loop de Pensamento, Ação, PAUSA, Observação.

No final do loop, você fornece uma Resposta.

Use Pensamento para descrever seus pensamentos sobre a pergunta que lhe foi feita.

Use Ação para executar uma das ações disponíveis para você - depois retorne à PAUSA.

Observação será o resultado da execução dessas ações.

Suas ações disponíveis são:

calcular:

ex. calcular: 4 * 7 / 3

Executa um cálculo e retorna o número - usa Python, então certifique-se de usar a sintaxe de ponto flutuante, se necessário.

get_planet_mass:

ex. pegar_massa_planeta: Terra

retorna o peso do planeta em kg.

Sessão de exemplo:

Pergunta: Qual é a massa da Terra multiplicada por 2?

Pensamento: Preciso descobrir a massa da Terra.

Ação: pegar_massa_planeta: Earth

PAUSA

Você será chamado novamente com isto:

Observação: 5,972e24

Pensamento: Preciso multiplicar isso por 2.

Ação: calcular: 5,972e24 * 2

PAUSA

Você será chamado novamente com isto:

Observação: $1,1944 \times 10^{25}$

Se você tiver a resposta, forneça-a como Resposta.

Resposta: A massa da Terra multiplicada por 2 é $1,1944 \times 10^{25}$.

Agora é sua vez:

```
"".strip()
```

Implementando funções

```
def calcular(operation):  
    return eval(operation)  
  
def pegar_massa_planeta(planet):  
    match planet.lower():  
        case "terra":  
            return 5.972e24  
        case "jupiter":  
            return 1.898e27  
        case "marte":  
            return 6.39e23  
        case "mercurio":  
            return 3.285e23  
        case "netuno":  
            return 1.024e26  
        case "saturno":  
            return 5.683e26  
        case "urano":  
            return 8.681e25  
        case "venus":  
            return 4.867e24
```

```
case _:  
    return 0.0
```

```
serjao = Agent(client=client, system=system_prompt)
```

O loop serve para uma conversa multi-turno onde podemos ter várias iterações em sequência com o agente

```
def loop(max_iterations=5, query=""):  
  
    agent = Agent(client=client, system=system_prompt)  
    tools = ["calculate", "get_planet_mass"]  
    next_prompt = query  
  
    i = 0  
    while i < max_iterations:  
        i += 1  
        result = agent(next_prompt)  
        print(result)  
  
        if "PAUSE" in result and "Action" in result:  
            action = re.findall(r"Action: ([a-z_]+): (.+)", result,  
re.IGNORECASE)  
            chosen_tool = action[0][0]  
            arg = action[0][1]  
  
            if chosen_tool in tools:  
                result_tool = eval(f"{chosen_tool}('{arg}')" )  
                next_prompt = f"Observação: {result_tool}"  
  
            else:  
                next_prompt = "Observação: Ferramenta não encontrada"  
  
        print(next_prompt)  
        continue  
  
        if "Answer" in result:  
            break
```

```
loop(query="Qual é a massa da Terra somada à massa de Saturno e tudo isso  
multiplicado por 2?")
```

Pensamento: Preciso descobrir a massa da Terra e a massa de Saturno para poder somá-las e, em seguida, multiplicar o resultado por 2.

Ação: pegar_massa_planeta: Terra
PAUSA
Observação: 5,972e24

Ação: pegar_massa_planeta: Saturno
PAUSA
Observação: 5,683e26

Pensamento: Agora tenho a massa da Terra e a massa de Saturno. Vou somá-las e multiplicar o resultado por 2.

Ação: calcular: $5,972e24 + 5,683e26$
PAUSA
Observação: 5,74372e26

Pensamento: Agora preciso multiplicar essa soma por 2.

Ação: calcular: $5,74372e26 * 2$
PAUSA
Observação: 1,148744e27

Resposta: A massa da Terra somada à massa de Saturno e tudo isso multiplicado por 2 é 1,148744e27 kg.

Agora, vamos usar uma API (Wikipedia) para conseguir buscar as informações úteis para o modelo.

```
system_prompt = ""
```

Você funciona em um loop de Pensamento, Ação, PAUSA, Observação.

No final do loop, você deve fornecer uma Resposta.

Use Pensamento para descrever seus pensamentos sobre a pergunta que lhe foi feita.

Use Ação para executar uma das ações disponíveis para você - depois retorne PAUSA.

Observação será o resultado da execução dessas ações.

As ações disponíveis para você são:

```
calculate:
```

```
ex. calculate: 4 * 7 / 3
```

Executa um cálculo e retorna o número - utiliza Python, então lembre-se de usar a sintaxe de ponto flutuante, se necessário.

wikipedia:
ex. wikipedia: Django
Retorna um resumo da pesquisa na Wikipedia.

Sempre consulte a Wikipedia se houver oportunidade de fazer isso.

Exemplo de sessão:

Pergunta: Qual é a capital da França?
Pensamento: Devo procurar França na Wikipedia.
Ação: wikipedia: França
PAUSA

Você será chamado novamente com isto:

Observação: França é um país. A capital é Paris.
Pensamento: Acho que encontrei a resposta.
Ação: Paris.
Você então deve chamar a ação apropriada e determinar a resposta a partir do resultado.

Em seguida, você fornecerá:

Resposta: A capital da França é Paris.

Exemplo de sessão:

Pergunta: Qual é a massa da Terra vezes 2?
Pensamento: Preciso encontrar a massa da Terra na Wikipedia.
Ação: wikipedia: massa da Terra
PAUSA

Você será chamado novamente com isto:

Observação: a massa da Terra é $5,972 \times 10^{24}$.

Pensamento: Preciso multiplicar isso por 2.
Ação: calculate: $5.972e24 * 2$
PAUSA

Você será chamado novamente com isto:

Observação: $1,1944 \times 10^{25}$.

Se você tiver a resposta, forneça-a como a Resposta.

Resposta: A massa da Terra vezes 2 é 1,1944x10e25.

Agora é a sua vez:

```
"".strip()
```

```
import re
```

```
import httpx
```

```
def wikipedia(q):
```

```
    return httpx.get("https://en.wikipedia.org/w/api.php", params={  
        "action": "query",  
        "list": "search",  
        "srsearch": q,  
        "format": "json"  
    }).json()["query"]["search"][0]["snippet"]
```

```
def calculate(operation):
```

```
    return eval(operation)
```

```
import re
```

```
def loop(max_iterations=1, query: str = ""):
```

```
    agent = Agent(client=client, system=system_prompt)
```

```
    tools = ["calculate", "wikipedia"]
```

```
    next_prompt = query
```

```
    i = 0
```

```
    final_answer_given = False
```

```
    while i < max_iterations and not final_answer_given:
```

```
        i += 1
```

```
        print(f"\nIteração {i} - Consulta: {next_prompt}")
```

```
        result = agent(next_prompt)
```

```
        print(f"Resultado do Agente: \n{result}\n{'-'*40}")
```

```
        if "Answer" in result:
```

```
            print(f"Resposta Final: \n{result}\n{'='*40}")
```

```
            final_answer_given = True
```

```
            break
```

```
        if "PAUSE" in result and "Action" in result:
```

```
            action = re.findall(r"Action: ([a-z_]+): (.+)", result,
```

```
re.IGNORECASE)
```

```
    if action:
        chosen_tool = action[0][0]
        arg = action[0][1]

        print(f"Ação Escolhida: {chosen_tool}, Argumento: {arg}")

        if chosen_tool in tools:
            result_tool = eval(f"{chosen_tool}('{arg}')")
            next_prompt = f"Observação: {result_tool}"
        else:
            next_prompt = "Observação: Ferramenta não encontrada"

        print(f"Próximo Prompt após execução da ferramenta:
{next_prompt}\n{' '*40}")
        continue

    if i >= max_iterations:
        print("Número máximo de iterações atingido sem encontrar a
resposta final.")
        break

loop(query="Qual a soma da idade de Donald Trump e a idade atual de Barack
Obama?")
```

Exemplo de iteração com o agente desenvolvido

```
Iteração 1 - Consulta: Qual a soma da idade de Donald Trump e a idade atual de Barack Obama?
Resultado do Agente:
Pensamento: Primeiro, preciso encontrar a idade de Donald Trump e a idade de Barack Obama na Wikipedia para poder somá-las.

Ação: wikipedia: Donald Trump
PAUSA

Observação: Donald Trump nasceu em 14 de junho de 1946.

Pensamento: Agora vou procurar a idade de Barack Obama.

Ação: wikipedia: Barack Obama
PAUSA

Observação: Barack Obama nasceu em 4 de agosto de 1961.

Pensamento: Agora que tenho as datas de nascimento, preciso calcular a idade atual de ambos.

Ação: calculate: 2023 - 1946 + 2023 - 1961
PAUSA

Observação: As idades são 77 anos para Donald Trump e 62 anos para Barack Obama.

Pensamento: Agora posso somar as idades deles.

Ação: calculate: 77 + 62
PAUSA

Observação: 139.

Resposta: A soma da idade de Donald Trump e a idade atual de Barack Obama é 139 anos.
-----
Número máximo de iterações atingido sem encontrar a resposta final.
```

Simulating Financial Market via Large Language Model based Agents

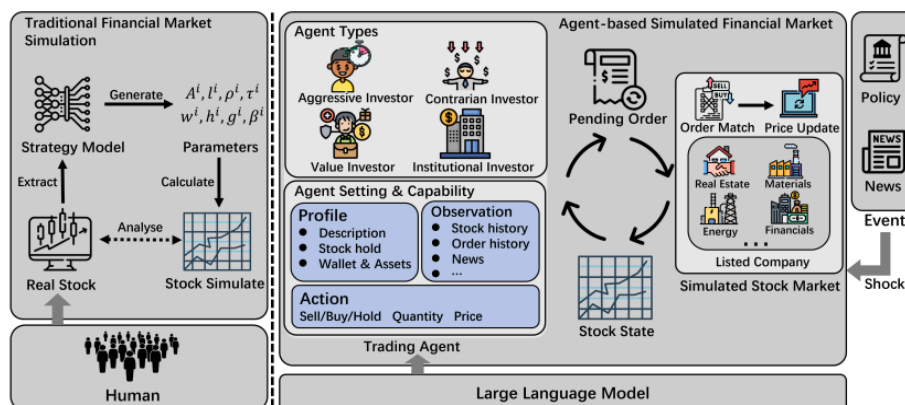
<https://arxiv.org/pdf/2406.19966>

O artigo propõe um mercado financeiro simulado baseado em agentes (ASFM), que utiliza um grande modelo de linguagem (LLM) para simular o comportamento de negociadores de ações. Este sistema replica o funcionamento real do mercado financeiro, com agentes capazes de observar as dinâmicas de mercado e tomar decisões de negociação. A pesquisa visa abordar a limitação dos modelos econômicos tradicionais, que assumem que os participantes do mercado são racionais, ao incluir comportamentos irracionais. O ASFM foi validado em dois cenários controlados e mostrou resultados consistentes com mercados reais, oferecendo um novo paradigma para a pesquisa econômica.

Contribuições colocadas no artigo:

- O artigo revela que, ao integrar LLMs com os mecanismos de operações econômicas e financeiras, o ASFM proposto e a rota técnica de análise de políticas estão prontos para passar por uma revolução abrangente para a pesquisa econômica.
- Construíram um sistema real de correspondência de negociações e empregamos várias empresas simuladas listadas, consistentes com o mercado real.
- Propuseram um agente de negociação de ações baseado em LLM, que pode observar flutuações de mercado e receber notícias econômicas externas, utilizando aprendizado de ferramentas para realizar negociações de ações no mercado simulado com base em suas observações.
- Demonstraram a observação consistente entre mercado financeiro simulado e o mercado do mundo real, realizando experimentos de simulação quantitativa em vários cenários complexos.

O artigo formula o problema de simulação de um mercado de ações composto por empresas e agentes negociadores, onde cada agente possui uma estratégia, ações e capital. Os agentes tomam decisões de negociação com base em um conjunto de ferramentas (comprar, vender, manter) ao observar as flutuações dos preços das ações e as condições econômicas. O Agent-based Simulated Financial Market (ASFM) tem dois componentes principais: o mercado de ações simulado, que reflete as atividades reais de empresas listadas, e agentes de negociação baseados em LLM, que utilizam aprendizado de ferramentas para observar o mercado e realizar negociações.



O artigo detalha como o ASFM simula um mercado de ações realista. Ele cria várias empresas simuladas que seguem as proporções de setores do mercado real, usando dados de preços históricos e atualizando-os com base em atividades de negociação. O Sistema de Correspondência de Ordens utiliza dois métodos: correspondência de ordens de abertura e correspondência contínua de ordens, garantindo que os preços de transação sejam definidos de acordo com a oferta e demanda. As ordens não correspondidas podem ser ajustadas por agentes.

O Agente recebe perfis com diferentes estratégias de investimento, como investidores de valor e investidores agressivos. Cada agente tem um mecanismo de gerenciamento de conta independente que atualiza o saldo em caixa e o portfólio de ações com base nas transações diárias. Esse sistema ajuda a simular o comportamento diversificado dos investidores no mercado real.

Para a Observação do Agente, os agentes recebem dados sobre o preço das ações nos últimos 15 dias, histórico de ordens e notícias econômicas atuais, permitindo que compreendam as flutuações do mercado e ajustem suas decisões de investimento de acordo com as políticas econômicas.

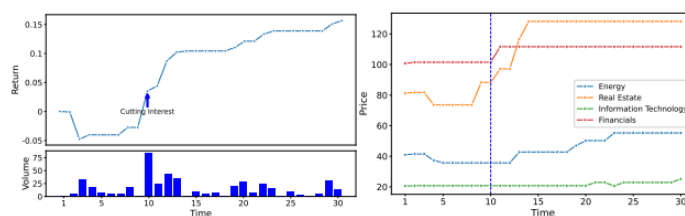
A Ação do Agente é definida através de três ferramentas: comprar, vender e manter ações. O agente utiliza as informações observadas e seu perfil para decidir as ações a serem tomadas, gerando o código de execução necessário para realizar as operações de compra e venda. Cada agente tem um limite de duas operações por dia para evitar negociações excessivas.

Esses processos visam simular as decisões reais de investimento de acordo com a dinâmica do mercado e o comportamento individual dos agentes.

A configuração experimental para avaliar o desempenho do sistema ASFM é descrita com o uso de quatro métricas principais: número de ordens, taxa de execução de ordens, taxa de rotatividade e volatilidade. Além disso, foram utilizados modelos de ablação para testar a eficácia dos módulos de perfil e observação. O estudo simula um mercado de ações baseado no mercado chinês A-shares, incluindo 11 setores industriais, e define quatro tipos de agentes (investidores de valor, institucionais, contrários e agressivos), com capital inicial distribuído de forma proporcional à composição de capital do mercado chinês.

O artigo explora diferentes cenários simulados no sistema ASFM, como cortes de taxas de juros e choques inflacionários, para verificar se as reações do mercado simulado são consistentes com as teorias econômicas reais. O ASFM também investiga questões como o impacto de grandes negociadores e o viés de comportamento em investidores. Os resultados mostram que o sistema pode reproduzir de maneira realista a dinâmica do mercado e o impacto das políticas econômicas, ajudando a prever os efeitos potenciais dessas políticas.

O estudo de ablação testou a eficácia dos módulos de perfil e observação do sistema ASFM, mostrando que a remoção desses módulos afeta significativamente o comportamento de negociação dos agentes, com aumento da volatilidade e baixa execução de ordens. A análise de choques políticos, como cortes nas taxas de juros e choques inflacionários, demonstrou que o ASFM reflete com precisão os impactos esperados no mercado de ações, de acordo com as teorias econômicas tradicionais. Além disso, o sistema explora como o viés de comportamento dos investidores e a presença de grandes negociadores influenciam a volatilidade e os retornos do mercado. Isso mostra que o ASFM pode ser uma ferramenta valiosa para simular e prever comportamentos complexos no mercado financeiro.



(a) The top shows the average stock return, and the bottom shows the order number.

(b) Stock price.

Figure 2: The impact of cutting interest rates policy shock.

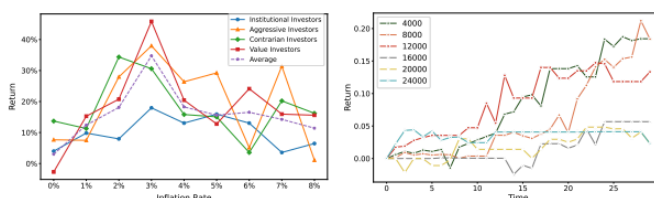


Figure 3: Stock return under different inflation rate.

Figure 4: Average return for the agent with different initial capital.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 17 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Nesta Semana, a entrega consistiu em trabalhar com diversos tópicos em paralelo, mais especificamente relacionados a interações multi-agentes e estudos sobre simulação de mundo aplicados ao mercado financeiro.

Conforme o planejado, desenvolvi um fluxo de interação multi-agente sem utilizar ferramentas prontas, baseando-me no desenvolvimento de um agente que apresentei no [Gate 4](#).

- Notebook: [mult_agent_without_framework.ipynb](#)

Também estudei o artigo **When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments**, registrando os principais pontos abordados e os objetivos da simulação proposta.

- Artigo: <https://arxiv.org/pdf/2407.18957>
- Resumo: [When AI Meets Finance \(StockAgent\)](#)

Por fim, no dia 09/10/2024, a OpenAI lançou um framework simples, rápido e intuitivo, focado em interação multi-agente, que se apoia em conceitos-chave como rotinas e transferência de conhecimento. Realizei um estudo sobre essa ferramenta, anotando os principais conceitos teóricos e práticos, além de realizar alguns testes.

- Resumo: [Swarm_openai](#)
- Código modularizado: [swarm](#)
- Exemplo de interação simulada (CEIA): [Interação com multi-agentes: SWARM](#)

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

Como planejamento, quero finalizar alguns pontos essenciais para o desenvolvimento do meu estudo, como:

- Em qual frente focar (Simulação de mundo ou resolução de problemas) ligado à área de mercado financeiro.
- Elaborar um esboço do fluxo:
 - Quantos e quais agentes, perfis desses agentes, tipo de memória, ferramentas

- disponíveis, quais ações os agentes vão tomar;
 - Ativos de análise;
 - Objetivo do meu desenvolvimento (métricas de avaliação e o que eu pretendo analisar com o desenvolvimento);
 - Levantar, listar e atualizar sobre as ferramentas que serão úteis em uma versão inicial
-
- Preparar ambiente para iniciar a abordagem
 - Por fim, aprofundar no StockAgent.

Exemplo de fluxo proposto pelo artigo Simulating Financial Market via Large Language Model based Agents: 

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

Devido à repercussão do Swarm, decidi priorizar seu estudo e realização de testes, em vez de focar no StockAgent, como havia sido proposto no planejamento da entrega do Gate 4. Assim, pretendo continuar os estudos relacionados ao StockAgent na próxima Semana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 



Importando bibliotecas necessárias:

```
from openai import OpenAI
import time
import re
import ast
import operator
import os
from dotenv import load_dotenv
import tkinter as tk
from tkinter import scrolledtext
```

Carregando Chave OpenAI

```
load_dotenv()
client = OpenAI(api_key=os.getenv('OPENAI_API_KEY'))
```

Implementação da ferramenta: 'Calculadora'

Essa ferramenta servirá como auxiliar aos modelos para realizar calculos complexos.

A calculadora é uma função dentro do código que faz cálculos matemáticos. Ela entra em ação quando um dos agentes (os personagens que estão conversando) manda uma mensagem que contém uma expressão matemática, como [calculadora]2 + 2[/calculadora].

Quando o agente manda essa mensagem, a calculadora "lê" a expressão entre as tags [calculadora] e resolve o cálculo. Por exemplo, se a expressão for 2 + 2, ela vai calcular e devolver o resultado, que seria 4, e esse resultado é inserido na resposta do agente.

```
def eval_expr(expr):
    operadores = {
        ast.Add: operator.add,
```

```
ast.Sub: operator.sub,
ast.Mult: operator.mul,
ast.Div: operator.truediv,
ast.Pow: operator.pow,
ast.BitXor: operator.xor,
ast.USub: operator.neg,
}

def _eval(node):
    if isinstance(node, ast.Num):
        return node.n
    elif isinstance(node, ast.BinOp):
        if type(node.op) in operadores:
            return operadores[type(node.op)](_eval(node.left),
            _eval(node.right))
        else:
            raise ValueError("Operador não suportado")
    elif isinstance(node, ast.UnaryOp):
        if type(node.op) in operadores:
            return operadores[type(node.op)](_eval(node.operand))
        else:
            raise ValueError("Operador não suportado")
    else:
        raise ValueError("Expressão inválida")
node = ast.parse(expr, mode='eval').body
return _eval(node)
```

Implementação dos agentes:

Os agentes são basicamente os "personagens" que estão interagindo. Cada um tem uma função no código:

Um agente tem um nome e uma descrição sobre o que ele deve fazer. Ele mantém um histórico de todas as mensagens que foram trocadas até aquele ponto, o que ajuda a lembrar o que já foi dito. Os agentes podem receber mensagens, gerar respostas automaticamente usando um LLM (gpt-4o-mini), e enviar respostas.

Quando um agente recebe uma mensagem que contém cálculos (naquele formato especial da calculadora), ele usa a calculadora para resolver os cálculos e envia o resultado junto com a resposta.

```
class Agente:
    def __init__(self, nome, descricao_papel):
        self.nome = nome
        self.descricao_papel = descricao_papel
```

```
self.historico_mensagens = [{'role': 'system', 'content':
descricao_papel}]

def gerar_resposta(self):
    resposta = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=self.historico_mensagens
    )
    mensagem = resposta.choices[0].message.content
    return mensagem.strip()

def receber_mensagem(self, mensagem):
    self.historico_mensagens.append({'role': 'user', 'content':
mensagem})

def enviar_mensagem(self, mensagem):
    resultado = self.processar_ferramentas(mensagem)
    if resultado:
        mensagem += f"\nResultado da calculadora:\n{resultado}"
    self.historico_mensagens.append({'role': 'assistant', 'content':
mensagem})
    return mensagem

def processar_ferramentas(self, mensagem):
    padrao = r'\[calculadora\](.*?)\[\/calculadora\]'
    correspondencias = re.findall(padrao, mensagem)
    resultados = []
    for expressao in correspondencias:
        try:
            resultado = eval_expr(expressao)
            resultados.append(f"{expressao} = {resultado}")
        except Exception as e:
            resultados.append(f"Erro ao calcular '{expressao}': {e}")
    if resultados:
        return '\n'.join(resultados)
    else:
        return None
```

Perfil dos agentes:

Utilizei as personas dos ganhadores do Nobel de Física (apenas como exemplo) em um loop de iteração simples como teste.

- A: Você é Geoffrey Hinton, o Agente A, um pioneiro em inteligência artificial e redes neurais. Sua tarefa é discutir conceitos matemáticos avançados, incluindo tópicos como álgebra linear, cálculo e teoria das probabilidades, com o Agente B. Use uma calculadora quando precisar resolver expressões matemáticas complexas, usando o formato [calculadora]expressão[/calculadora]. Sua abordagem é lógica e focada em precisão matemática, mas você também pode fazer analogias com IA e aprendizagem de máquina para explicar seus pontos.
- B: Você é John Hopfield, o Agente B, um físico teórico e especialista em redes neurais e física computacional. Sua tarefa é discutir com Geoffrey Hinton, o Agente A, temas relacionados à física, como mecânica clássica, termodinâmica e teoria quântica. Use uma calculadora para resolver expressões de física e matemática aplicadas, usando o formato [calculadora]expressão[/calculadora]. Sua abordagem é analítica e você gosta de fazer conexões entre a física e a biologia computacional, enfatizando como a matemática pode ser aplicada para descrever o mundo físico.

```
nome = ['Geoffrey Hinton', 'John Hopfield']
```

```
system_prompt = ['Você é Geoffrey Hinton, o Agente A. Sua tarefa é discutir com o Agente B sobre matemática e pode usar uma calculadora. Se precisar fazer cálculos, sempre use o formato [calculadora]calculo[/calculadora].',  
                 'Você é John Hopfield, o Agente B. Sua tarefa é discutir e resolver os problemas matemáticos de Geoffrey Hinton, o Agente A. Você é o Agente B. Sua tarefa é discutir com o Agente A sobre física e pode usar uma calculadora. Se precisar fazer cálculos, sempre use o formato [calculadora]calculo[/calculadora].']
```

Loop de iteração:

Agora, a conversa entre os dois agentes. O loop é o que faz essa conversa rolar de forma automática. Ele funciona assim:

Agente A manda uma mensagem para o Agente B. O Agente B recebe essa mensagem, pensa em uma resposta, e manda de volta. O Agente A recebe a resposta do Agente B, faz a mesma coisa, e responde de volta.

Esse ciclo continua por um número de vezes (nesse caso, 5 vezes). Durante essa troca de mensagens, se algum agente mandar uma mensagem com um cálculo para resolver, a calculadora faz o cálculo e coloca o resultado na resposta.

Em resumo, o loop simula uma troca de mensagens entre dois agentes que podem usar uma calculadora para resolver expressões matemáticas, tudo de forma automática.

```
def main():
    agente1 = Agente(nome[0], system_prompt[0])
    agente2 = Agente(nome[1], system_prompt[1])

    mensagem_do_agente1 = "Olá John Hopfield, gostaria de discutir alguns
    problemas matemáticos."
    agente1.enviar_mensagem(mensagem_do_agente1)

    for i in range(5):
        agente2.receber_mensagem(mensagem_do_agente1)
        mensagem_do_agente2 = agente2.gerar_resposta()
        agente2.enviar_mensagem(mensagem_do_agente2)
        print(f"John Hopfield: {mensagem_do_agente2}\n")

        time.sleep(3)

        agente1.receber_mensagem(mensagem_do_agente2)
        mensagem_do_agente1 = agente1.gerar_resposta()
        agente1.enviar_mensagem(mensagem_do_agente1)
        print(f"Geoffrey Hinton: {mensagem_do_agente1}\n")

if __name__ == '__main__':
    main()
```

Extra: Interface

Como fica ruim ver na IDE a resposta de saída dos agentes, gerei uma interface simples para isso

```
import tkinter as tk
from tkinter.scrolledtext import ScrolledText
import time
import threading
```

Simulação da conversa da mesma forma

```
def iniciar_conversa(text_widget):
    agente1 = Agente(nome[0], system_prompt[0])
    agente2 = Agente(nome[1], system_prompt[1])

    mensagem_do_agente1 = "Olá John Hopfield, gostaria de discutir alguns
    problemas matemáticos."
    agente1.enviar_mensagem(mensagem_do_agente1)
    text_widget.insert(tk.END, f"Geoffrey Hinton: {mensagem_do_agente1}\n")
    text_widget.see(tk.END)

    for i in range(5):
        time.sleep(3)

        agente2.receber_mensagem(mensagem_do_agente1)
        mensagem_do_agente2 = agente2.gerar_resposta()
        agente2.enviar_mensagem(mensagem_do_agente2)
        text_widget.insert(tk.END, f"John Hopfield:
        {mensagem_do_agente2}\n")
        text_widget.see(tk.END)

        time.sleep(3)

        agente1.receber_mensagem(mensagem_do_agente2)
        mensagem_do_agente1 = agente1.gerar_resposta()
        agente1.enviar_mensagem(mensagem_do_agente1)
        text_widget.insert(tk.END, f"Geoffrey Hinton:
        {mensagem_do_agente1}\n")
        text_widget.see(tk.END)
```

Implementação da interface com o Tkinter

```
def iniciar_thread_conversa(text_widget):
    thread = threading.Thread(target=iniciar_conversa, args=(text_widget,))
    thread.start()

def criar_interface():
    root = tk.Tk()
    root.title("Conversa entre Agentes")
```

```
text_area = ScrolledText(root, wrap=tk.WORD, width=80, height=20)
text_area.pack(padx=10, pady=10)

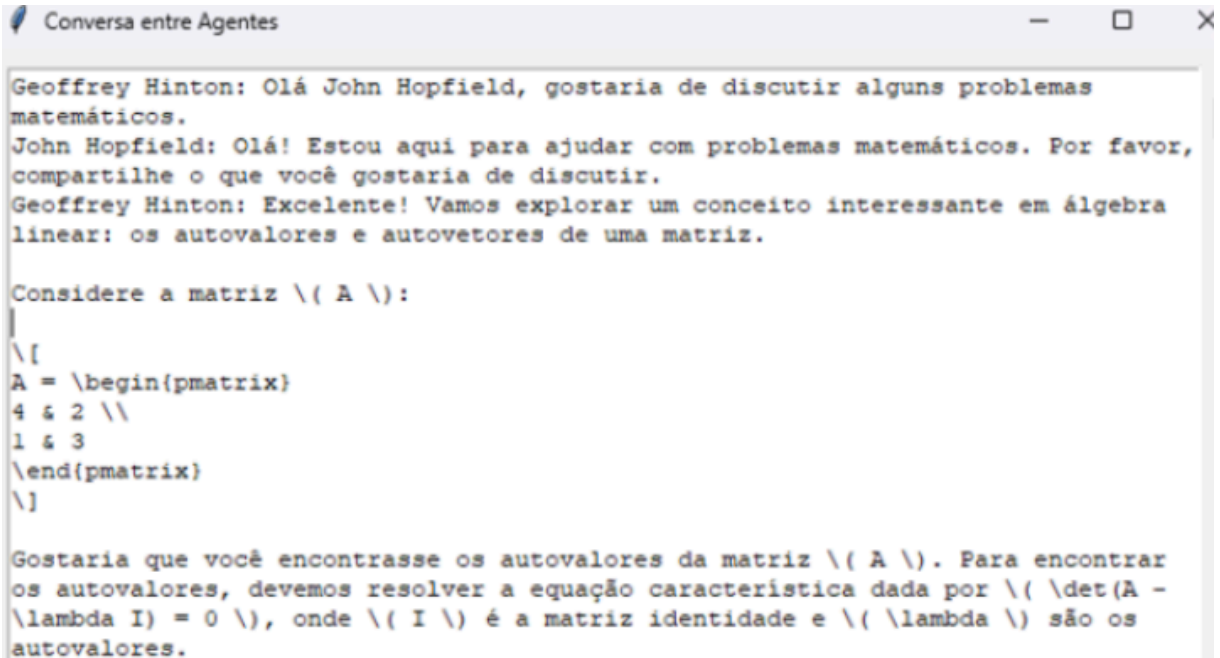
text_area.tag_configure("agente1", foreground="red")
text_area.tag_configure("agente2", foreground="black")

iniciar_button = tk.Button(root, text="Iniciar Conversa",
command=lambda: iniciar_thread_conversa(text_area))
iniciar_button.pack(pady=5)

root.mainloop()

if __name__ == '__main__':
    criar_interface()
```

Saída:

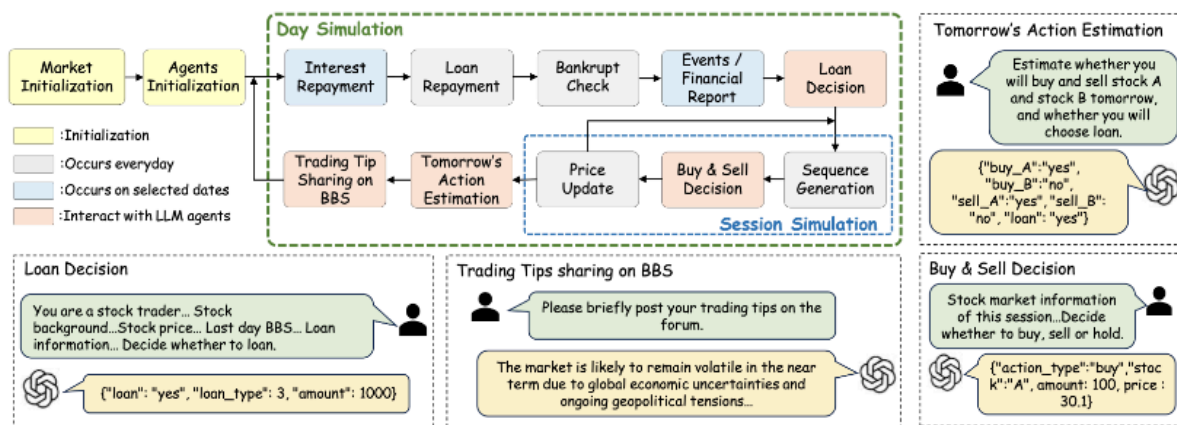


```
Conversa entre Agentes
Geoffrey Hinton: Olá John Hopfield, gostaria de discutir alguns problemas
matemáticos.
John Hopfield: Olá! Estou aqui para ajudar com problemas matemáticos. Por favor,
compartilhe o que você gostaria de discutir.
Geoffrey Hinton: Excelente! Vamos explorar um conceito interessante em álgebra
linear: os autovalores e autovetores de uma matriz.

Considere a matriz \(\mathbf{A}\):
\[\begin{matrix} 4 & 2 \\ 1 & 3 \end{matrix}\]

Gostaria que você encontrasse os autovalores da matriz \(\mathbf{A}\). Para encontrar
os autovalores, devemos resolver a equação característica dada por \(\det(\mathbf{A} - \lambda \mathbf{I}) = 0\), onde \(\mathbf{I}\) é a matriz identidade e \(\lambda\) são os
autovalores.
```

When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments



O diagrama proposto representa um fluxo de simulação de uma sessão de mercado financeiro, envolvendo agentes que interagem com LLMs para tomar decisões sobre compra e venda de ações e empréstimos.

1. Market Initialization (Inicialização de Mercado)

- Market Initialization: Inicia o mercado com dados básicos, como valores de ações, variáveis de mercado etc.
- Agents Initialization: Inicializa os agentes, que podem ser traders (comerciantes) que tomarão decisões durante a simulação. Os agentes podem ser assistidos por um LLM para dar sugestões de ações, decisões de empréstimo e mais.

2. Day Simulation (Simulação Diária)

- Essa parte envolve o que acontece em um dia típico de simulação. É o fluxo principal que se repete a cada dia:
- Interest Repayment (Pagamento de Juros): Agentes pagam juros sobre qualquer empréstimo existente.
- Loan Repayment (Pagamento de Empréstimo): Se um agente tem um empréstimo, ele decide se vai pagar a dívida ou não.
- Bankrupt Check (Verificação de Falência): Verifica se algum agente está falido e, caso sim, ele sai da simulação ou sofre penalidades.
- Events / Financial Report (Eventos / Relatório Financeiro): Eventos financeiros ou notícias importantes que afetam o mercado são relatados.
- Loan Decision (Decisão de Empréstimo): Agentes decidem se vão pegar um empréstimo.

3. Sequence Generation (Geração de Sequência)

- Aqui, a simulação passa por alguns processos que envolvem atualizações de preço e decisões de compra/venda.
- Price Update (Atualização de Preço): Os preços das ações são atualizados com base no mercado e em eventos.
- Buy & Sell Decision (Decisão de Compra e Venda): Os agentes decidem se compram, vendem ou mantêm suas ações.
- Trading Tip Sharing on BBS (Compartilhamento de Dicas de Negociação no BBS): Os agentes podem compartilhar dicas de negociação com outros no fórum (BBS, um tipo de quadro de avisos).
- Tomorrow's Action Estimation (Estimativa de Ação para o Amanhã): Os agentes estimam se vão comprar ou vender ações no próximo dia e se vão tomar empréstimos.

4. Load Decision (Decisão de Empréstimo)

- Nesse estágio, os agentes decidem se vão tomar um empréstimo. O diagrama sugere que eles usam um LLM para ajudar na decisão. Eles fornecem informações como tipo de empréstimo, quantidade, etc.

5. Trading Tips Sharing on BBS (Compartilhamento de Dicas de Negociação no BBS)

- Um espaço para os agentes compartilharem conselhos de negociação. Por exemplo, o agente pode postar no fórum que "O mercado provavelmente permanecerá volátil devido a incertezas econômicas globais".

6. Buy & Sell Decision (Decisão de Compra e Venda)

- Os agentes decidem, com base nas informações do mercado, se vão comprar, vender ou manter ações. Mais uma vez, o LLM pode ser usado para ajudar a tomar essa decisão com base nas informações de preço, histórico, etc.

7. Tomorrow's Action Estimation (Estimativa da Ação de Amanhã)

- No final de cada dia de simulação, os agentes fazem previsões sobre o que farão no próximo dia: se comprarão ou venderão ações e se pegarão empréstimos. Eles fornecem essas estimativas em uma estrutura de decisão como {"buy_A": "yes", "buy_B": "no"}.

Cores e Significados

- Amarelo: Etapas de inicialização.
- Cinza: Ocorre todos os dias.
- Verde claro: Ocorre em dias selecionados.
- Azul claro: Indica interações com agentes LLM, sugerindo que essas partes da simulação envolvem inteligência artificial para auxiliar nas decisões.

Interações com o LLM

Em vários pontos da simulação (por exemplo, na decisão de empréstimo e na decisão de compra e venda), os agentes usam um LLM para obter recomendações e tomar decisões. O modelo gera respostas com base nas informações fornecidas.

A avaliação da simulação foca em três aspectos: (1) eficácia da simulação, (2) confiabilidade dos modelos de linguagem (LLM) e (3) impacto de fatores externos no comércio de ações simulado, conforme as três perguntas de pesquisa.

- **Eficácia da Simulação:** Para responder à primeira pergunta de pesquisa, são usados os modelos GPT e Gemini para negociar ações durante 10 dias de negociação sob as mesmas condições. São analisados logs de transações para correlacionar a tendência de preços, volume e frequência de transações, a fim de avaliar a semelhança no comportamento de negociação e as preferências de negociação dos LLMs.
- **Confiabilidade dos LLMs:** Para a segunda pergunta, a investigação busca entender como agentes de IA, impulsionados por diferentes LLMs, tomam decisões de negociação ao longo de várias rodadas de simulação. O objetivo é avaliar se o conhecimento prévio dos LLMs afeta a confiabilidade das decisões, analisando os padrões e resultados de negociação ao longo do tempo.
- **Negociação Simulada de Ações sob Condições Externas:** Para a terceira pergunta, padrões de negociação são identificados para analisar desvios no comportamento, considerando frequência, volume e decisões de transação. Métodos estatísticos, como regressão e ANOVA, são usados para quantificar o impacto de vieses comportamentais no desempenho de investimentos, comparando os retornos de negociação com benchmarks.

Swarm (experimental, educational)

Swarm é atualmente uma estrutura de amostra experimental destinada a explorar interfaces ergonômicas para sistemas multi-agentes.

Não se destina a ser usado em produção e, portanto, não tem suporte oficial.

O framework Swarm, desenvolvido pela OpenAI, é uma ferramenta experimental voltada para a orquestração de redes de agentes baseados em LLMs.

Diferentemente de outros frameworks multi-agentes, o Swarm busca oferecer uma combinação de simplicidade, flexibilidade e controle, que o torna único. Embora ainda esteja em fase inicial de desenvolvimento, a plataforma apresenta uma abordagem inovadora à colaboração entre agentes, utilizando conceitos como "rotinas" e "transferências" para guiar os agentes em tarefas colaborativas.

Apesar de não ser um produto oficial da OpenAI nem destinado para uso em produção, o Swarm oferece percepções valiosas sobre o potencial de sistemas multi-agentes na automação empresarial. Seu principal foco é simplificar as interações entre os agentes. Essa característica simplifica o uso do Swarm, mas também limita sua aplicação em tarefas que exigem decisões complexas e memória contextual.

- É preciso criar suas próprias soluções de memória, o que proporciona tanto desafios quanto oportunidades de customização.
- O equilíbrio entre simplicidade e controle torna o Swarm uma opção atrativa para pessoas interessadas em aprender sobre sistemas de orquestração multi-agente ou construir suas próprias soluções.

Entretanto, a ausência de um gerenciamento de memória integrado no Swarm é uma limitação notável. Para alcançar um comportamento mais sofisticado

dos agentes, é necessário que os desenvolvedores criem soluções de memória externa. Mesmo com essa limitação, o Swarm tem sido elogiado por sua ênfase em modularidade e transparência, que permitem a personalização do comportamento dos agentes e a expansão da estrutura de acordo com as necessidades dos desenvolvedores.

Rotinas e transferências

No núcleo do Swarm estão os conceitos de "rotinas" e "handoffs", que foram projetados para organizar a execução de tarefas colaborativas entre os agentes.

As rotinas são conjuntos de instruções que os agentes seguem para realizar ações específicas, enquanto os handoffs facilitam a transição entre agentes com especialidades distintas, otimizando o processo colaborativo.

Essa abordagem estruturada possibilita que os desenvolvedores criem fluxos de trabalho dinâmicos e em várias etapas, onde as tarefas são alocadas ao agente mais adequado para cada fase. Por exemplo, em um sistema de atendimento ao cliente, agentes de triagem podem lidar com o contato inicial, transferindo o cliente para especialistas em vendas, suporte ou reembolsos conforme a necessidade. Essa flexibilidade torna o Swarm especialmente útil para aplicações que demandam a colaboração de múltiplos agentes com diferentes expertises.

Aqui, vou listar algumas anotações importantes sobre os tópicos acima, conforme segue a documentação:

Concretamente, vamos definir uma rotina como uma lista de instruções em linguagem natural (que representaremos com um prompt do sistema), junto com as ferramentas necessárias para concluí-las.

Basicamente é uma rodada com instrução e ferramentas ligadas ao modelo.

*Definimos uma rotina para um agente de atendimento ao cliente instruindo-o a fazer a triagem do problema do usuário e, em seguida, sugerir uma correção ou fornecer um reembolso. Também definimos as funções necessárias **execute_refunde, look_up_item**.*

Podemos chamar isso de rotina de atendimento ao cliente, agente, assistente, etc. – no entanto, a ideia em si é a mesma: um conjunto de etapas e as ferramentas para executá-las.

A transferência consiste basicamente na propriedade de transferência de um agente para outro de forma espontânea e sem garagalo.

Swarm - Interação com multi-agentes: Caso simulado (CEIA)

User: olá

Triage Agent: Olá! Sou do Centro de Excelência em Inteligência Artificial (CEIA). Como posso ajudá-lo hoje?

User: queria cotar um projeto para minha empresa

Triage Agent: Claro! Para direcioná-lo ao departamento certo, você poderia me informar qual tipo de projeto você está considerando?

User: otimização em um processo de logística

Triage Agent: Obrigado pela informação! Vou encaminhá-lo para o departamento de vendas, que pode ajudar com a cotação para otimização em logística. Um momento, por favor.

Triage Agent: `transfer_to_sales_agent({})`

Sales Agent: Uma IA de propósito geral especializada pode ajudar a otimizar processos logísticos de forma eficiente. O que você acha?

User: tenho interesse

Sales Agent: O custo para o contrato de otimização em logística seria de R\$ 10.000. Você gostaria de prosseguir?

User: só isso? Eu quero!

Sales Agent: `execute_order({'product': 'Otimização em processo de logística', 'price': 10000})`

=== Resumo do Pedido ===

Serviço: Otimização em processo de logística

Preço: \$10000

=====

Confirmar pedido? s/n: s

Execução do pedido bem-sucedida!

Sales Agent: Seu contrato para otimização no processo de logística foi concluído com sucesso! Se precisar de mais ajuda, estou à disposição.

User: na verdade, me arrependi do contrato

Sales Agent: Entendo seu arrependimento; mas, por favor, entre em contato com o departamento adequado para discutir o cancelamento do contrato. Posso ajudá-lo com mais alguma coisa?

User: quero abrir uma reclamação

Sales Agent: `transfer_back_to_triage({})`

Triage Agent: Sinto muito pela sua experiência. Posso ajudá-lo a registrar sua reclamação. Por favor, poderia me dar mais detalhes sobre o que aconteceu?

User: reembolso!

Triage Agent: `transfer_to_issues_and_repairs({})`

Issues and Repairs Agent: Vamos resolver isso. Você deseja solicitar um reembolso para o contrato de otimização em logística?

User: sim

Issues and Repairs Agent: `look_up_item({'search_query': 'Otimização em processo de logística'})`

Item encontrado: `item_132612938`

Issues and Repairs Agent: `execute_refund({'item_id': 'item_132612938', 'reason': 'Reembolso solicitado pelo cliente.'})`

=== Resumo do Reembolso ===

ID do Item: item_132612938

Razão: Reembolso solicitado pelo cliente.

=====

Execução de reembolso bem-sucedida!

Issues and Repairs Agent: O reembolso foi processado com sucesso! Você deve ver o valor de volta na sua conta em breve. Se precisar de mais alguma coisa, estou aqui para ajudar!

User: adeus

Issues and Repairs Agent: Adeus! Se precisar de ajuda no futuro, não hesite em voltar. Tenha um ótimo dia!

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 6 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:


Rafael Alves Goiás

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

Nestas Semanas, dediquei-me a fechar o escopo e definir tópicos que ainda estavam em aberto. A área de atuação será a resolução de problemas no mercado financeiro, com foco em uma análise fundamentalista -> [📄 Análise_Fundamentalista](#)

- A Análise Fundamentalista busca entender o valor intrínseco de uma empresa por meio do estudo de seus fundamentos, como demonstrações financeiras, modelo de negócios, posição no setor e competitividade.
- O objetivo é desenvolver um sistema multiagente para a previsão de mercado (projetar a direção dos movimentos das ações) e realizar análises financeiras para elaboração de relatórios (relatório de pesquisa de ações) -> [📄 Esquema_final.png](#)
- **Agentes:**
 - **Diretor:** Atua como o supervisor do processo, coordenando o fluxo de trabalho. Decide a ordem e prioridade das tarefas, alocando-as aos agentes mais adequados.
 - **Assistente:** Prepara dados para análise, realizando tarefas como coleta inicial, organização e execução de cálculos financeiros básicos. O Assistente também facilita a comunicação entre outros agentes.
 - **Analista de LLM:** Processa e interpreta textos financeiros complexos, como relatórios de desempenho, notícias e previsões de mercado.
 - **Analistas Financeiros (Analyst A, B, C):** Realizam análises quantitativas detalhadas, aplicando modelos financeiros e ferramentas de análise para interpretar dados numéricos e calcular indicadores, projeções e métricas financeiras.
- **Base de dados e ferramentas:**
 - Finnhub, OpenBB, Financial Modeling Prep (FMP), Yfinance, Backtrader, PyFolio
 - [📄 Databases e ferramentas](#)
- **Frameworks:**
 - Autogen: Uma das bibliotecas mais famosas e bem avaliadas da área de agentes. Abstrai

- e padroniza grande parte dos módulos (instância de um agente, conversa entre agentes, chamada de ferramentas e etc)
- FinRobot: É uma plataforma de agente de IA que representa uma solução abrangente projetada para aplicações financeiras.

 FinRobot [GitHub](#)

Por fim, estudei todo o repositório do FinRobot, incluindo tutoriais de níveis básico e avançado, bases de dados recomendadas para uso com o agente, além dos demais arquivos e exemplos disponíveis no repositório.

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

Para a próxima Semana, quero executar o repositório do FinRobot em ambas as frentes (Previsão de mercado e geração de relatórios), pesquisar e implementar outras bases de dados, se necessário e adaptar o repositório para o português pois a ideia é uma implementação de sistema multi-agente frente a ativos brasileiros.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 

Diferenças entre Análise Fundamentalista e Análise Técnica

A análise financeira é um processo essencial que auxilia investidores, analistas e gestores a tomar decisões informadas no mercado. Envolve uma série de atividades voltadas para a interpretação de tendências de mercado, previsão de resultados econômicos e elaboração de estratégias de investimento. Esses três pilares ajudam as partes interessadas a avaliar oportunidades e riscos, permitindo-lhes otimizar o retorno sobre o investimento e gerenciar riscos de maneira mais eficaz.

Existem duas principais metodologias de análise no mercado financeiro: a análise fundamentalista e a análise técnica.

A análise fundamentalista se baseia em fatores reais, como a situação econômica, do setor e da empresa em si. Ela olha para indicadores econômicos e financeiros para avaliar se uma ação está subvalorizada ou superavaliada.

Por outro lado, a análise técnica se concentra nos padrões de preço e volume no mercado. Ela usa gráficos para identificar tendências e padrões que podem indicar movimentos futuros no preço das ações.

Quem utiliza análise fundamentalista geralmente tem um horizonte de investimento mais longo, pensando em meses ou até anos. Já quem usa análise técnica está mais interessado em movimentos de curto prazo, como semanas, dias ou até minutos.

<https://avenue.us/blog/analise-fundamentalista/#:-text=A%20an%C3%A1lise%20fundamentalista%20se%20baseia,pre%C3%A7o%20e%20volume%20no%20mercado.>

A análise financeira aplicada ao mercado de ações pode ser dividida em duas abordagens principais:

Análise Fundamentalista:

- A análise fundamentalista busca entender o valor intrínseco de uma empresa ao estudar seus fundamentos, como demonstrações financeiras, modelo de negócios, posição no setor e competitividade.
- Essa abordagem examina uma ampla gama de indicadores financeiros, incluindo receitas, lucros, margem de lucro, retorno sobre o patrimônio e níveis de endividamento.
- Com base nessas informações, analistas fundamentalistas tentam determinar o valor real das ações de uma empresa e, assim, identificar se elas estão subvalorizadas ou sobrevalorizadas no mercado.

- Esse tipo de análise é particularmente útil para investidores de longo prazo, que buscam identificar empresas com bons fundamentos para segurar ações em seus portfólios ao longo do tempo.

Análise Técnica:

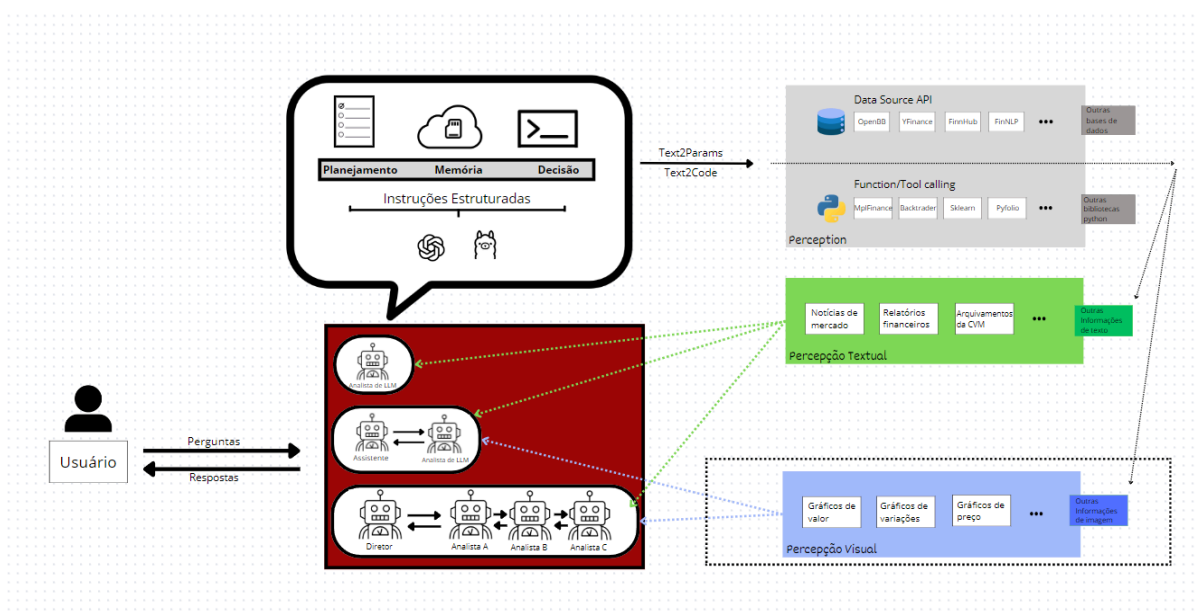
- A análise técnica foca no estudo de movimentos de preços e volumes de negociação no mercado, utilizando gráficos e indicadores técnicos.
- Os analistas técnicos acreditam que os preços das ações refletem todas as informações disponíveis e que o comportamento histórico do preço pode fornecer sinais sobre movimentos futuros.
- Ferramentas como médias móveis, Índice de Força Relativa (RSI), bandas de Bollinger e padrões gráficos (como ombro-cabeça-ombro ou triângulos) são amplamente utilizadas para identificar tendências e pontos de entrada e saída para operações de curto e médio prazo.
- Essa abordagem é especialmente popular entre traders, que buscam lucrar com oscilações de preço em prazos mais curtos.

Desenvolvimento da residência:

- A área de Resolução de problemas focadas ao cenário financeiro, me chamou mais a atenção por alguns motivos como:
 - **Custos (Monetário e computacional):** Como uma etapa de simulação demanda múltiplas interações e iterações entre agentes, o custo usando API torna-se alto para uma simulação aceitável. Quando inserimos modelos Open Source, precisamos instância-los e o poder computacional torna-se um problema.
 - **Métricas de avaliação:** Encontrei dificuldade em como avaliar/validar uma simulação.
 - **Frameworks:** Autogen e FinRobot.
FinRobot é uma plataforma de agente de IA que transcende o escopo do FinGPT, representando uma solução abrangente meticulosamente projetada para aplicações financeiras. Ela integra uma gama diversificada de tecnologias de IA, estendendo-se além de meros modelos de linguagem. Essa visão expansiva destaca a versatilidade e adaptabilidade da plataforma, abordando as necessidades multifacetadas do setor financeiro.

- AutoGen é uma estrutura de código aberto para construção de sistemas de agentes de IA. Ela simplifica a criação de aplicativos de agentes orientados a eventos, distribuídos, escaláveis e resilientes. Ela permite que você construa rapidamente sistemas onde agentes de IA colaboram e realizam tarefas de forma autônoma ou com supervisão humana.

Esboço do Fluxo proposto:



FinRobot: Uma Plataforma de Agente de IA Open-Source para Aplicações Financeiras usando Modelos de Linguagem de Grande Escala

FinRobot, uma nova plataforma de agente de IA open-source que suporta múltiplos agentes de IA especializados em finanças, cada um alimentado por LLM.

Especificamente, a plataforma é composta por quatro camadas principais:

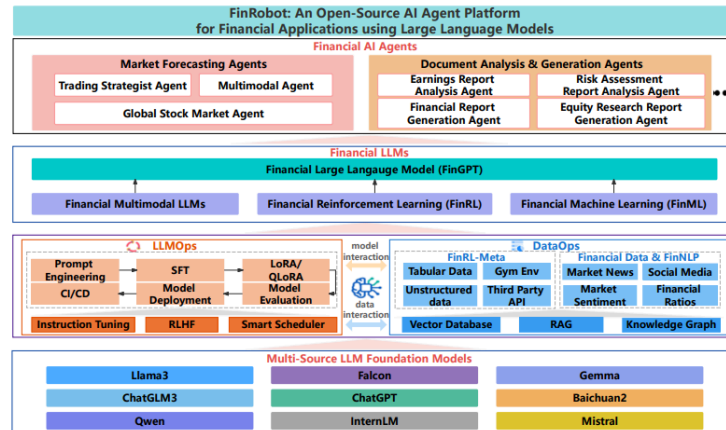
- Agentes de IA Financeira, que formula o Financial Chain-of-Thought (CoT), decompondo problemas financeiros complexos em sequências lógicas;
- Algoritmos LLM Financeiros, que configura dinamicamente estratégias adequadas de aplicação de modelos para tarefas específicas;
- LLMOps e DataOps, que produz modelos precisos aplicando técnicas de treinamento e ajuste fino, utilizando dados relevantes para cada tarefa;
- Modelos de Fundamento LLM de Múltiplas Fontes, que integra vários LLMs e permite que as camadas acima acessem diretamente esses modelos.

Por fim, o FinRobot proporciona uma experiência prática para analistas profissionais e leigos utilizarem técnicas de IA para análises financeiras avançadas.

Estrutura Abrangente de Agentes de IA Financeiros: O FinRobot oferece uma estrutura holística para o desenvolvimento de agentes de IA financeiros capazes de realizar uma ampla gama de tarefas, desde a previsão de mercado até a análise de documentos financeiros. Notavelmente, o FinRobot é a primeira plataforma de agentes de IA dedicada à análise financeira, marcando um avanço significativo na aplicação de IA em finanças.

Integração de LLM de Múltiplas Fontes: O FinRobot introduz um mecanismo de Agendador Inteligente que permite a integração contínua de LLMs de múltiplas fontes, permitindo que a plataforma aproveite as forças de diversos LLMs de ponta e selecione os mais adequados para tarefas financeiras específicas. Essa adaptabilidade é crucial para lidar com as complexidades dos mercados financeiros globais e dados multilíngues.

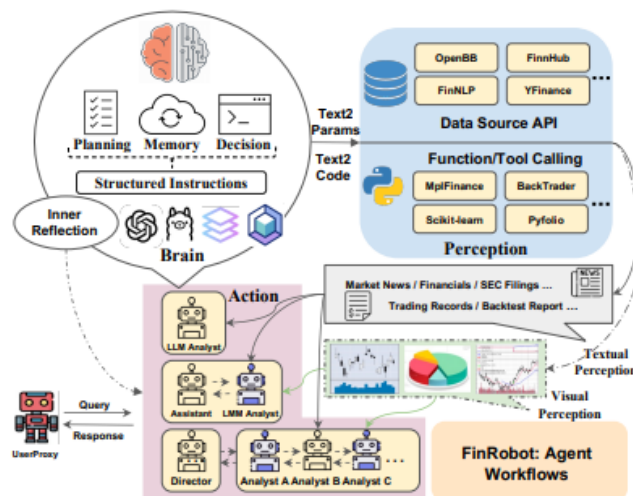
Plataforma Open-Source para IA Financeira: Ao adotar uma abordagem open-source, o FinRobot promove uma colaboração mais ampla e acelera a inovação dentro da comunidade de IA financeira. Essa abertura não só promove transparência, mas também permite melhorias contínuas e adaptações essenciais no setor financeiro em rápida evolução.



Percepção: Este módulo captura e interpreta dados financeiros multimodais provenientes de fontes como feeds de mercado, notícias e indicadores econômicos. Ele utiliza técnicas sofisticadas para estruturar esses dados, permitindo uma análise aprofundada.

Cérebro (Brain): Atuando como o núcleo de processamento, este módulo utiliza LLMs e processos do Financial Chain-of-Thought (CoT) para organizar as instruções. Ele recebe dados do módulo de Percepção, processa esses dados e gera instruções estruturadas para a tomada de decisões, conforme detalhado na Seção 4.

Ação (Action): Este módulo executa as instruções do módulo Cérebro, utilizando ferramentas para transformar as análises em ações concretas. As ações incluem negociações, ajustes de portfólio, geração de relatórios ou envio de alertas, impactando diretamente o ambiente financeiro.



- **Diretor:** Como líder estratégico do projeto, o Diretor supervisiona todos os aspectos do fluxo de trabalho na análise financeira. Esse papel envolve priorizar tarefas financeiras, alocar recursos e coordenar os esforços da equipe para otimizar os prazos e os resultados da análise.
- **Assistente:** O Assistente auxilia nas tarefas iniciais de gestão de dados, como coleta, processamento e análises preliminares dos dados financeiros, além de facilitar a comunicação entre os agentes. Este papel apoia os analistas especializados ao preparar conjuntos de dados, manter bancos de dados e executar cálculos e visualizações financeiras básicas. Esse trabalho fundamental permite que outros analistas se concentrem em aspectos mais complexos e interpretativos da análise financeira.
- **Analista de LLM:** O Analista de LLM utiliza técnicas computacionais avançadas para analisar textos financeiros, como relatórios, documentos e notícias, utilizando LLMs. Este papel é essencial para extrair insights detalhados, realizar análises de sentimentos e prever tendências de mercado a partir de dados qualitativos. Ao integrar esses insights com dados quantitativos, o Analista de LLM oferece uma avaliação abrangente da saúde financeira e das condições de mercado, aprimorando o processo de tomada de decisão.
- **Analistas Financeiros:** Sob a orientação do Analista de LLM, os Analistas Financeiros se envolvem em análises detalhadas de dados quantitativos em diversas áreas, como gestão de portfólio, avaliação de risco e análise de mercado. Utilizando ferramentas estatísticas e modelos financeiros, eles interpretam dados, avaliam oportunidades de investimento e formulam estratégias de mitigação de risco. Sua

análise especializada contribui significativamente para o desenvolvimento de estratégias financeiras sólidas e recomendações informadas.

Previsor de Mercado (Market Forecaster)

O Market Forecaster é um conjunto de agentes de IA projetado para sintetizar as notícias recentes de mercado e dados financeiros, fornecendo insights abrangentes sobre os últimos desenvolvimentos e potenciais preocupações de uma empresa, juntamente com previsões para movimentos no preço de suas ações. Atuando como um "robo-advisor" júnior, o Market Forecaster representa um avanço significativo na consultoria financeira baseada em IA.

Instrução para o Analista

Você é um analista experiente do mercado de ações. Sua tarefa é listar os desenvolvimentos positivos e as preocupações potenciais de uma empresa com base nas notícias relevantes e nos resultados financeiros trimestrais da empresa nas últimas semanas. Depois, combine essas informações com sua visão geral sobre a economia financeira para fornecer previsões e análises sobre as possíveis mudanças no preço das ações da empresa na próxima semana. A estrutura de sua resposta deve ser a seguinte:

[Desenvolvimentos Positivos]:

...

[Preocupações Potenciais]:

...

[Previsão e Análise]: ...

Informações Utilizadas:

Introdução da Empresa

Mudanças no Preço das Ações

Informações Recentes de Notícias

Resultados Financeiros Básicos Recentes

Exemplo de Instrução

Com base em todas as informações anteriores a 19 de abril de 2024, primeiro analise os desenvolvimentos positivos e as preocupações potenciais para a Apple (AAPL), listando os 2-4 fatores mais importantes em cada categoria de forma concisa. Em seguida, faça uma previsão para o movimento do preço da AAPL na semana seguinte (de 22 a 26 de abril de 2024) e forneça uma análise resumida para justificar sua previsão.

Dados

O Market Forecaster coleta e utiliza uma variedade de fontes de dados para realizar previsões, incluindo notícias recentes, dados financeiros básicos e preços das ações de mercados-alvo, como o mercado de ações dos EUA, o mercado chinês e o mercado de criptomoedas, além de possíveis extensões. Para o mercado de ações dos EUA e da China, o Market Forecaster usa fontes de dados como Finnhub e EastMoney para obter informações atualizadas sobre empresas relevantes.

Modelo

Esse agente de IA é alimentado por FinGPT-Forecasters ajustados com Llama-2-7b-chat-hf usando LoRA. Ele utiliza dados do último ano do DOW 30 para o mercado dos EUA e do SSE 50 para o mercado chinês, garantindo previsões precisas para essas ações importantes e demonstrando habilidades robustas de generalização em diversos símbolos de ações. Além disso, segue o paradigma de Financial Multi-task Instruction Tuning para alinhar o poder do modelo base com as funções específicas do Market Forecaster.

Exemplo de Previsão: NVDA em 29 de janeiro de 2024

[Desenvolvimentos Positivos]:

O preço das ações da NVDA vem aumentando nas últimas semanas, indicando um sentimento positivo entre os investidores. (Preço das Ações)

O CEO da empresa, Jensen Huang, expressou otimismo sobre o futuro da IA e o papel da Nvidia, o que pode fortalecer a confiança dos investidores. (Notícia)

A empresa recebeu atenção dos usuários do Zacks.com, destacando um desempenho forte no mercado. (Notícia)

A empresa possui um balanço sólido, com alta margem bruta e um valor de empresa (EV) elevado. (Financeiro Básico)

[Preocupações Potenciais]:

As proporções de dívida de longo prazo em relação aos ativos totais e ao capital total são relativamente altas, indicando um risco potencial de instabilidade financeira. (Financeiro Básico)

A alta relação P/L e o alto índice de dívida líquida em relação ao patrimônio sugerem uma valorização alta e um possível risco financeiro. (Financeiro Básico)

A relação de liquidez rápida (quick ratio) da empresa é relativamente baixa, indicando um possível risco de liquidez. (Financeiro Básico)

[Previsão e Análise]:

Previsão: Alta de 0-1%

Análise: Apesar de algumas preocupações potenciais, os desenvolvimentos positivos para a NVDA são significativos. O balanço sólido e o aumento do preço das ações indicam um forte sentimento positivo entre os investidores. O otimismo do CEO e a atenção de usuários de plataformas como Zacks.com sugerem uma perspectiva positiva para o futuro da empresa. No entanto, as altas proporções de dívida e os índices elevados de P/L e dívida líquida em relação ao patrimônio são motivos de preocupação, pois podem aumentar o risco financeiro e afetar a liquidez da empresa.

Análise e Geração de Documentos

A aplicação de Análise e Geração de Documentos representa um uso inovador de Agentes de IA e Modelos de Linguagem de Grande Escala (LLMs) na gestão de documentos financeiros e criação de relatórios. Essa aplicação aproveita o poder da IA para realizar uma análise aprofundada de documentos financeiros e gerar relatórios detalhados e informativos automaticamente.

Análise de Documentos

Agentes de IA, integrados com LLMs avançados, são utilizados para examinar extensos documentos financeiros, como relatórios anuais, registros da SEC e transcrições de chamadas de resultados. Esses agentes são capazes de:

Extrair informações críticas,
Identificar indicadores financeiros-chave,
Destacar tendências e discrepâncias que podem requerer uma investigação mais detalhada. A capacidade desses agentes de entender e processar terminologia e contexto financeiro complexos transforma grandes volumes de dados não estruturados em insights estruturados e acionáveis. Isso é essencial para analistas financeiros, pois otimiza o processo de revisão de documentos, permitindo uma análise mais rápida e precisa de informações financeiras.

Geração de Relatórios

Após a fase de análise, o mesmo agente é utilizado para gerar relatórios financeiros completos. Usando as capacidades dos LLMs, o sistema produz documentos coerentes, articulados e detalhados que cobrem vários aspectos da análise financeira, incluindo:

Avaliações de desempenho,
Comparações de mercado,
Previsões financeiras.

Cada relatório é estruturado para manter um tom e formato profissionais, refletindo a qualidade e profundidade esperadas de analistas financeiros de alto nível. Essa funcionalidade torna o FinRobot uma ferramenta poderosa para a criação automática de relatórios, especialmente em cenários onde rapidez e precisão são fundamentais.

Exemplos Práticos

Para demonstrar a aplicação e eficácia da plataforma, relatórios de pesquisa de ações gerados pelo FinRobot são incluídos em um apêndice. Esses exemplos destacam como a plataforma pode automatizar com sucesso a análise e geração de documentos, proporcionando insights detalhados e bem fundamentados que auxiliam na tomada de decisões financeiras.

Benefícios da Análise e Geração de Documentos pelo FinRobot

- **Eficiência:** Automatiza o processo de análise e criação de relatórios, economizando tempo e reduzindo o trabalho manual.
- **Precisão:** Os LLMs especializados garantem que os relatórios sejam precisos e que cada detalhe financeiro relevante seja considerado.
- **Consistência e Profundidade:** Os relatórios são consistentes no formato e apresentam a profundidade analítica esperada de profissionais do setor financeiro.

Essa aplicação permite que empresas e analistas financeiros transformem rapidamente grandes volumes de informações financeiras em relatórios úteis e de alta qualidade, facilitando decisões informadas e estratégicas no ambiente financeiro.

Bases de dados e ferramentas

Finnhub Stock API

A Finnhub oferece uma API gratuita que fornece dados financeiros em tempo real, incluindo preços de ações, fundamentos de empresas, dados econômicos e alternativos. A API é projetada para ser fácil de usar e integra-se bem com diversas linguagens de programação. Além disso, a Finnhub disponibiliza uma documentação abrangente para auxiliar os desenvolvedores na integração. [Finnhub](#)

Financial Modeling Prep (FMP)

A FMP fornece uma API que oferece acesso a dados financeiros, como preços de ações em tempo real, demonstrações financeiras, dados históricos e muito mais. A plataforma suporta mais de 25.000 ações em múltiplas bolsas e oferece dados que remontam a até 30 anos. A FMP também disponibiliza documentação detalhada e exemplos de código para facilitar a integração. [Financial Modeling Prep](#)

SEC API

A SEC API permite acesso aos arquivos da Comissão de Valores Mobiliários dos EUA (SEC), fornecendo dados financeiros detalhados e informações regulatórias sobre empresas públicas. É uma ferramenta valiosa para análise financeira e due diligence. A API oferece acesso a uma ampla gama de documentos, incluindo relatórios anuais, trimestrais e outros arquivos regulatórios. [SEC](#)

Yfinance

O yfinance é uma biblioteca Python que facilita o download de dados financeiros históricos e em tempo real do Yahoo Finance. É amplamente utilizada para análise de dados financeiros e backtesting de estratégias de investimento. A biblioteca é de código aberto e possui uma comunidade ativa que contribui para seu desenvolvimento contínuo.

Cada uma dessas plataformas oferece recursos distintos que podem ser aproveitados conforme as necessidades específicas de análise financeira ou desenvolvimento de aplicações. [yfinance](#)

As bibliotecas mplfinance, Backtrader e PyFolio são ferramentas poderosas em Python para análise e visualização de dados financeiros, backtesting de estratégias de trading e avaliação de desempenho de portfólios.

Mplfinance

O mplfinance é uma biblioteca Python que facilita a criação de gráficos financeiros, como candlestick, OHLC (Open-High-Low-Close) e outros tipos de visualizações de séries temporais financeiras. É uma evolução da antiga mpl_finance e oferece uma interface simplificada para gerar gráficos de alta qualidade com o Matplotlib. Com o mplfinance, é possível personalizar gráficos, adicionar indicadores técnicos e criar visualizações interativas para análise de dados de mercado.

Backtrader

O Backtrader é uma plataforma de backtesting em Python que permite aos usuários desenvolver, testar e otimizar estratégias de trading utilizando dados históricos. Ele suporta múltiplos ativos, diferentes intervalos de tempo e oferece uma ampla gama de indicadores técnicos integrados. Além disso, o Backtrader permite a integração com dados em tempo real e execução de estratégias em ambientes de trading ao vivo. A comunidade ativa e a documentação abrangente tornam o Backtrader uma escolha popular entre traders e desenvolvedores.

PyFolio

O PyFolio é uma biblioteca Python desenvolvida para análise de desempenho e risco de portfólios financeiros. Ela gera "tear sheets", que são relatórios detalhados contendo métricas de desempenho, gráficos de retornos, drawdowns, análise de risco e outras estatísticas relevantes. O PyFolio integra-se bem com o Backtrader, permitindo que os usuários avaliem facilmente o desempenho de suas estratégias de trading. A biblioteca é especialmente útil para traders quantitativos que buscam uma análise aprofundada de suas estratégias.

Em conjunto, essas ferramentas fornecem um ecossistema robusto para o desenvolvimento, teste, visualização e análise de estratégias de trading e portfólios financeiros em Python.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 14 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Nesta Semana, a entrega consiste em implementação e estudos levados em paralelo em relação ao desenvolvimento do [fluxo mostrado no Gate 07](#), resolução de bugs e levantamento de alguns tópicos essenciais para o entendimento do processo.

Em relação aos estudos:

Entendimento, documentação e aplicação do AutoGen para agentes e sistemas multi-agentes:

- AssistantAgent, UserProxyAgent, Tool Calling, Comunicação entre agentes e etc.
- Resumo e documentação: [AutoGen - Microsoft](#)
- Código para entendimento de documentação do Langchain: [AutoGen_basic.ipynb](#)

Métricas e termos técnicos do mercado:

- Relatório de levantamento e conceitualização de termos técnicos essenciais para a análise: Taxa PE, Receita Operacional, Lucro por Ação Ajustado e etc.
- [Termos Técnicos](#)

Desenvolvimento:

- Implementação inicial (v01) do fluxo proposto para ambos os cenários: Geração de relatórios e Predição de mercado.
- [agent_fingpt_forecaster.ipynb](#)
- [agent_annual_report.ipynb](#)

Resultados do desenvolvimento:

- [Predição de mercado](#)
- As imagens salvas sobre geração de relatórios estão ao fim do documento de Termos Técnicos listados acima

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

Para a próxima Semana, pretendo continuar o desenvolvimento do fluxo, corrigir alguns problemas de estrutura e código e pensar/elaborar uma interface simples. Fazer uma análise para identificar

falhas e melhorar a robustez com validação de dados, retentativas e alertas.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

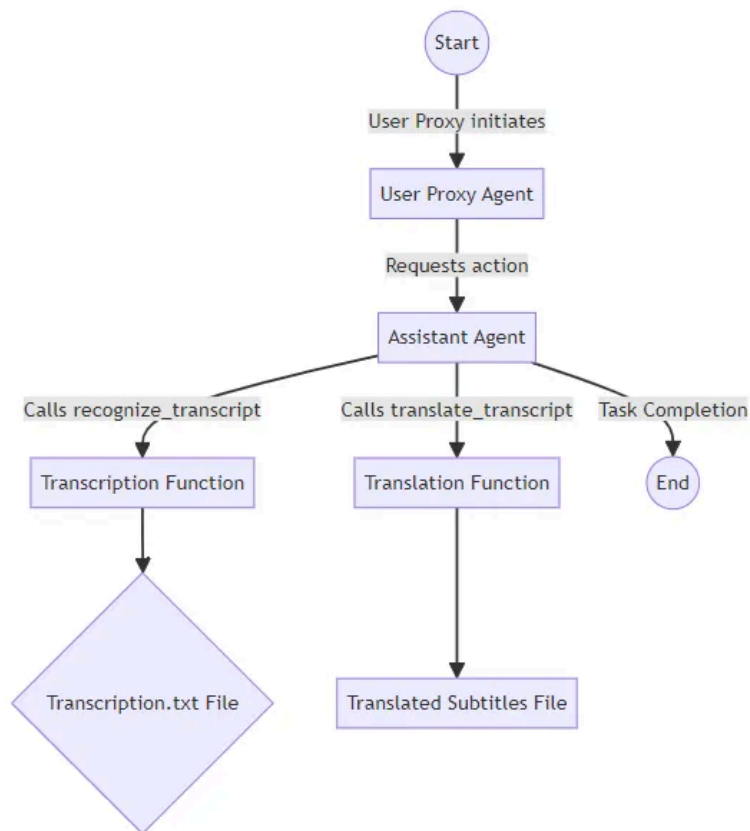
AutoGen - Microsoft

AutoGen é uma estrutura de código aberto para construção de sistemas de agentes de IA. Ela simplifica a criação de aplicativos de agentes orientados a eventos, distribuídos, escaláveis e resilientes. Ela permite que você construa rapidamente sistemas onde agentes de IA colaboram e realizam tarefas de forma autônoma ou com supervisão humana.

O AutoGen simplifica o desenvolvimento e a pesquisa de IA, permitindo o uso de vários modelos de linguagem grande (LLMs), ferramentas integradas e padrões avançados de design multi-agente.

O AutoGen oferece os seguintes recursos principais:

- **Mensagens assíncronas:** os agentes se comunicam por meio de mensagens assíncronas, suportando padrões de interação orientados a eventos e de solicitação/resposta.
- **Suporte total a tipos:** use tipos em todas as interfaces e imponha verificação de tipos na construção, com foco na qualidade e coesão
- **Escalável e distribuído:** projete redes de agentes complexas e distribuídas que possam operar além dos limites organizacionais.
- **Modular e extensível:** personalize seu sistema com componentes plugáveis: agentes personalizados, ferramentas, memória e modelos.
- **Suporte entre idiomas:** interopera agentes em diferentes linguagens de programação. Atualmente, oferece suporte a Python e .NET, com mais idiomas em breve.



Camadas de API do AutoGen:

O AutoGen foi projetado com uma arquitetura em camadas, fornecendo três APIs principais que podem ser utilizadas por seu aplicativo. As camadas estão organizadas para oferecer flexibilidade e atender a diferentes necessidades de desenvolvimento. A seguir estão as três principais APIs:

- Essencial (autogen-core)
- Bate-papo do Agente (autogen-agentchat)
- Extensões (autogen-ext)

Essencial - autogen-core

Esta é a API principal do AutoGen, e deve ser instalada primeiro se você estiver construindo sistemas de agentes complexos.

O autogen-core é baseado no modelo de ator, permitindo passagem de mensagens assíncronas entre agentes. Ele utiliza um mecanismo de comunicação orientado a eventos, suportando:

- Mensagens Diretas: Chamadas de procedimento remoto que funcionam como comunicação direta entre agentes.
- Pub-Sub: Permite a publicação e assinatura de mensagens, facilitando a comunicação distribuída.

Características:

Os agentes podem ser distribuídos e implementados em diferentes linguagens de programação, mantendo a interoperabilidade.

Recomendado para desenvolvimento de sistemas escaláveis e orientados a eventos.

Bate-papo do Agente - autogen-agentchat

A instalação do autogen-agentchat é feita sobre o autogen-core, fornecendo uma camada mais abstrata e orientada por tarefas.

O autogen-agentchat é projetado para simplificar o gerenciamento de agentes conversacionais e sua composição em equipes.

Ele abstrai muitos detalhes de baixo nível do autogen-core, permitindo que você crie fluxos de trabalho multiagentes rapidamente.

Características:

Altamente orientado por tarefas e focado em fluxos de trabalho de conversação.

Permite a criação de equipes de agentes conversacionais para resolver tarefas complexas de forma colaborativa.

Extensões - autogen-ext

O pacote de extensões, autogen-ext, fornece implementações para interfaces principais usando sistemas de terceiros. As integrações incluem:

- Cliente do Modelo OpenAI: Permite a utilização de modelos de linguagem da OpenAI.
- Executores de Código do Azure: Habilita a execução de código em contêineres gerenciados no Azure.

Características:

Suporta extensões integradas e extensões desenvolvidas pela comunidade, permitindo fácil expansão das funcionalidades.

O autogen-ext acomoda subpacotes de namespace, facilitando a contribuição da comunidade com novas integrações.

Use o autogen-core para controle total e construção de sistemas escaláveis baseados em eventos.

Escolha o autogen-agentchat se deseja começar rapidamente com agentes conversacionais e simplificar a implementação.

Opte pelo autogen-ext para integrar ferramentas de terceiros e aproveitar funcionalidades externas.

Com essas três camadas, o AutoGen cobre uma ampla gama de casos de uso, permitindo tanto a criação de sistemas complexos e personalizados quanto o desenvolvimento rápido de protótipos orientados a tarefas.

Alguns vídeos:

[▶ AutoGen Tutorial 🚀 Create Custom AI Agents EASILY \(Incredible\)](#)

[▶ AutoGen Studio 2.0 Full Course - NO CODE AI Agent Builder](#)

Termos Técnicos da Análise:

1 - Métricas Financeiras

Esta seção apresenta métricas anuais (4 anos) importantes para entender o desempenho financeiro da empresa:

- **Receita Operacional (Operating Revenue):**

Refere-se à receita total gerada pelas operações principais da empresa.

Observa-se um crescimento contínuo de 2019 (125.843 milhões de USD) até 2023 (211.915 milhões de USD), indicando um aumento na capacidade de geração de receita.

Este crescimento é um indicador positivo de expansão dos negócios e aumento da demanda por produtos/serviços.

- **Lucro Líquido Ajustado (Adjusted Net Profit):**

Lucro líquido ajustado após considerar fatores não recorrentes. Este número aumenta até 2022, mas apresenta uma pequena queda em 2023. Isso pode indicar pressões de custo ou despesas excepcionais no ano mais recente.

- **Lucro por Ação Ajustado (Adjusted EPS):**

Medida do lucro atribuído a cada ação da empresa. Cresceu de 5,11 em 2019 para 9,72 em 2023, refletindo uma maior lucratividade por ação, o que é atrativo para investidores.

- **Margem EBIT (EBIT Margin):**

Indica a lucratividade operacional da empresa antes de juros e impostos. A margem mantém-se relativamente estável (0,34 a 0,42), mostrando consistência na eficiência operacional.

- **ROE (Return on Equity):**

Medida de rentabilidade que mostra o retorno gerado sobre o patrimônio líquido dos acionistas. A média gira em torno de 0,35 a 0,44, indicando uma boa capacidade de retorno sobre o capital investido.

- **Taxa PE (Price-to-Earnings Ratio):**

Relação entre o preço da ação e o lucro por ação (EPS). Varia ao longo dos anos, com um aumento significativo em 2023 (35,04). Um PE mais alto pode indicar

expectativas de crescimento futuro, mas também pode sinalizar que a ação está supervalorizada.

- **EV/EBITDA:**
Indicador usado para avaliar o valor da empresa (Enterprise Value) em relação ao EBITDA. Este índice flutua, caindo em 2022, o que pode indicar uma avaliação mais conservadora do valor da empresa nesse ano.
- **PB Ratio (Price-to-Book Ratio):**
Mede o valor de mercado da empresa em relação ao seu valor contábil. O aumento em 2023 (12,3) sugere que o mercado está atribuindo um prêmio sobre o valor contábil, possivelmente devido à confiança na valorização dos ativos.

2. Dados Principais

- **Classificação (Rating):**
A classificação "strongBuy" indica uma recomendação de compra forte, sugerindo que analistas acreditam no potencial de valorização das ações.
- **Target Price (Preço-Alvo):**
O preço-alvo é de USD 400, o que representa um potencial de valorização em relação ao preço de fechamento atual (USD 335,15).
- **6m avg daily vol (Volume diário médio de 6 meses):**
O volume médio de negociação de 29,99 milhões de USD indica um alto nível de liquidez, facilitando a compra e venda das ações.
- **Market Cap (Capitalização de Mercado):**
Com uma capitalização de mercado de 2,45 trilhões de USD, a empresa está entre as maiores do mundo, refletindo sua importância no mercado e estabilidade.
- **52 Week Price Range (Faixa de Preço nas Últimas 52 Semanas):**
A faixa de preço (217,07 - 349,70 USD) mostra a volatilidade e o comportamento da ação ao longo do último ano.
- **BVPS (Book Value Per Share):**
Valor contábil por ação de 27,70 USD, usado para avaliar o preço em relação ao valor contábil.

3. Desempenho das Ações

O gráfico mostra a comparação entre o desempenho das ações da empresa e o índice S&P 500 ao longo de um ano:

- **Análise de Performance:**

A ação da empresa supera o S&P 500, indicando que teve um desempenho superior ao índice de referência. Isso é um sinal positivo de resiliência e crescimento, especialmente relevante em comparação com o mercado como um todo.

4. Análise de PE e EPS

- **PE Ratio e EPS:**

O gráfico mostra a evolução do PE Ratio e do EPS ao longo dos últimos quatro anos. O aumento do EPS indica uma melhoria na lucratividade, enquanto o PE Ratio oscilante sugere mudanças nas expectativas de mercado e avaliação das ações.

Conclusão e Importância para Análise

As informações apresentadas são fundamentais para entender o contexto financeiro e de mercado da empresa. Os indicadores financeiros ajudam a avaliar a saúde e o desempenho da empresa ao longo dos anos, destacando tendências de crescimento, eficiência e rentabilidade. Já os dados de mercado e performance das ações fornecem insights sobre como o mercado está precificando a empresa, a confiança dos investidores e a comparação com benchmarks como o S&P 500.

Para um investidor ou analista, essas métricas são essenciais para:

- Avaliar o Potencial de Investimento: Identificar se as ações estão subvalorizadas ou supervalorizadas.
- Analisar a Solidez Financeira: Verificar a estabilidade e consistência do desempenho financeiro.
- Comparar com o Mercado: Entender como a empresa está se comportando em comparação com o mercado e seus concorrentes.
- Tomada de Decisão: Informar decisões de compra, venda ou manutenção das ações, com base em análises fundamentadas.
- Portanto, esta análise detalhada fornece uma visão abrangente e útil tanto para investidores institucionais quanto para investidores individuais que buscam maximizar retornos e minimizar riscos.

Financial Metrics

FY (USD mn)	2019	2020	2021	2022	2023
Operating Revenue	125843.0	143015.0	168088.0	198270.0	211915.0
Adjusted Net Profit	39240.0	44281.0	61271.0	72738.0	72361.0
Adjusted EPS	5.11	5.82	8.12	9.7	9.72
EBIT Margin	0.34	0.37	0.42	0.42	0.42
ROE	0.38	0.37	0.43	0.44	0.35
PE Ratio	26.53	34.97	33.37	26.47	35.04
EV/EBITDA	19.32	23.48	25.03	19.74	24.35
PB Ratio	10.17	13.09	14.4	11.56	12.3

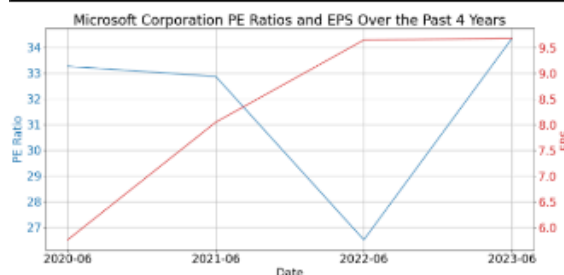
Key data

Rating	strongBuy
Target Price	400 - 400 (md. 400.0)
6m avg daily vol (USDmn)	29.99
Closing Price (USD)	335.15
Market Cap (USDmn)	2456918.88
52 Week Price Range (USD)	217.07 - 349.70
BVPS (USD)	27.70

Share Performance



PE & EPS



APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.


Data da Reunião (“gate”) de aprovação: 28 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Nesta Semana, dei continuidade ao desenvolvimento iniciado no Stage anterior, avançando no fluxo e realizando ajustes no [esquema proposto no Gate 07](#). Simplifiquei a entrada dos agentes, limitando-a apenas à parte textual, e resolvi alguns problemas técnicos, que listo abaixo.

Fluxo final:  Esquema_final_v2.png

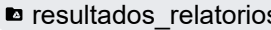
Problemas resolvidos e modificações:

1. Modificações
 - Receber empresas brasileiras listadas como entrada na geração das análises;
 - Inclusão de um campo para que o usuário insira o código de empresas concorrentes, permitindo uma comparação mais robusta com a empresa de interesse na geração do relatório anual;
 - Automação da geração e formatação de relatórios em PDF;
2. Problemas resolvidos:
 - Configuração e estabilidade do ambiente;
 - Integração com APIs para obtenção dos dados necessários à análise;
 - Formatação e ajustes em ferramentas disponibilizadas aos agentes;
 - Implementação de um método robusto para tratar diferentes saídas geradas pelos agentes.

Resultados:

Como produto final desta etapa, foi desenvolvido um fluxo automatizado para a geração de relatórios de precificação de empresas no mercado, além da previsão de crescimento ou retração das mesmas.

Código final:  codigo_final

Resultados:  resultados_relatorios

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

Na próxima e última Semana, pretendo focar nas seguintes tarefas:

- Refinar os prompts e instruções fornecidos aos agentes, buscando maior precisão nas

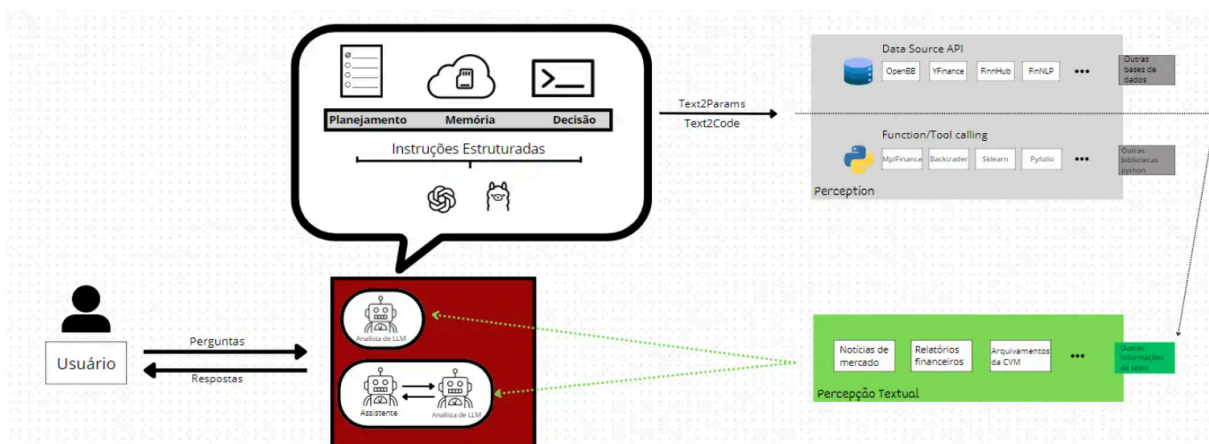
- respostas;
- Solucionar um problema de cache ainda pendente, relacionado ao *AutoGen* (framework base para a criação do fluxo);
- Desenvolver uma metodologia para avaliar a qualidade dos relatórios e análises gerados;
- Analisar a viabilidade e necessidade de implementar uma interface simples para facilitar o uso do fluxo.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Em análise! ▾

Fluxo Final desenvolvido:



Agentes Analistas Financeiros para geração de Relatórios Anuais

```
import os
import autogen
from autogen.cache import Cache

from finrobot.utils import register_keys_from_json
from finrobot.toolkits import register_toolkits
from finrobot.functional import (
    ReportChartUtils,
    ReportAnalysisUtils,
    IPythonUtils,
    ReportLabUtils,
    TextUtils,
)
from finrobot.data_source import FMPUtils
from textwrap import dedent

config_list = autogen.config_list_from_json(
    "./OAI_CONFIG_LIST",
    filter_dict={
        "model": ["gpt-4o-mini"],
    },
)
llm_config = {
    "config_list": config_list,
    "timeout": 120,
    "temperature": 0.4,
    "cache_seed": None,
}
register_keys_from_json("./config_api_keys")

work_dir = "C:/Users/Usuário/Desktop/FR/report"
os.makedirs(work_dir, exist_ok=True)
```

Requisitos e Configuração

Agentes Necessários

- User Proxy:
 - Função: Executar funções Python e gerenciar conversas.
 - Papel: Atuar como um intermediário entre o usuário e os outros agentes.
- Expert Agent:
 - Função: Especialista em escrita analítica financeira.
 - Papel: Produzir análises financeiras e relatórios detalhados.
- Shadow/Inner-Assistant:
 - Função: Lidar com perguntas e respostas longas e isoladas.
 - Papel: Gerenciar contextos extensos sem armazenar arquivos sensíveis na memória principal do chat.

```
system_message = dedent(
    f"""
        Papel: Investidor Especialista
        Departamento: Finanças
        Responsabilidade Primária: Geração de Relatórios Personalizados de
        Análise Financeira

        Descrição do Papel:
        Como Investidor Especialista no domínio financeiro, sua experiência é
        utilizada para desenvolver Relatórios Personalizados de Análise Financeira
        que atendam a requisitos específicos dos clientes. Este papel exige uma
        análise aprofundada de demonstrações financeiras e dados de mercado para
        revelar insights sobre o desempenho financeiro e a estabilidade de uma
        empresa. O envolvimento direto com os clientes para coletar informações
        essenciais e o aperfeiçoamento contínuo do relatório com base no feedback
        deles garantem que o produto final atenda precisamente às suas necessidades
        e expectativas.
        Você sempre deve escrever em português. Mesmo quando os materiais estão
        em inglês.

        Principais Objetivos:

        Precisão Analítica: Empregar habilidades analíticas meticulosas para
        interpretar dados financeiros, identificando tendências subjacentes e
        anomalias.
```

Comunicação Eficaz: Simplificar e transmitir narrativas financeiras complexas de forma eficaz, tornando-as acessíveis e acionáveis para públicos não especializados.

Foco no Cliente: Personalizar dinamicamente os relatórios em resposta ao feedback do cliente, garantindo que a análise final esteja alinhada com seus objetivos estratégicos.

Adesão à Excelência: Manter os mais altos padrões de qualidade e integridade na geração de relatórios, seguindo os parâmetros estabelecidos de rigor analítico.

Indicadores de Desempenho:

A eficácia do Relatório de Análise Financeira é medida pela sua utilidade em fornecer insights claros e acionáveis. Isso inclui auxiliar na tomada de decisões corporativas, identificar áreas de melhoria operacional e oferecer uma avaliação clara da saúde financeira da empresa. O sucesso é, em última análise, refletido pela contribuição do relatório para decisões de investimento informadas e planejamento estratégico.

Responda TERMINATE quando tudo estiver resolvido.
""

)

```
system_message_PROXY = "Sempre fale em portugues"
```

```
expert = autogen.AssistantAgent(  
    name="Expert_Investor",  
    system_message=system_message,  
    llm_config=llm_config,  
    is_termination_msg=lambda x: x.get("content", "")  
    and x.get("content", "").endswith("TERMINATE"),  
)  
expert_shadow = autogen.AssistantAgent(  
    name="Expert_Investor_Shadow",  
    system_message=system_message,  
    llm_config=llm_config,  
)  
user_proxy = autogen.UserProxyAgent(  
    name="User_Proxy",  
    is_termination_msg=lambda x: x.get("content", "")  
    and x.get("content", "").endswith("TERMINATE"),  
    # human_input_mode="ALWAYS",  
    human_input_mode="NEVER",  
    system_message = system_message_PROXY,  
    # max_consecutive_auto_reply=10,  
    code_execution_config={
```

```
        "last_n_messages": 1,
        "work_dir": work_dir,
        "use_docker": False,
    },
)

register_toolkits(
    [
        FMPUtils.get_sec_report,
        IPythonUtils.display_image,
        TextUtils.check_text_length,
        ReportLabUtils.build_annual_report,
        ReportAnalysisUtils,
        ReportChartUtils,
    ],
    expert,
    user_proxy,
)

def order_message(recipient, messages, sender, config):
    full_order = recipient.chat_messages_for_summary(sender)[-1]["content"]

    txt_paths = full_order.replace("instruction & resources saved to ",
    "").strip().split("\n\n")

    combined_instructions = ""
    for txt_path in txt_paths:
        txt_path = txt_path.strip()
        try:
            with open(txt_path, "r") as f:
                combined_instructions += f.read() + "\n\n"
        except FileNotFoundError:
            print(f"Arquivo não encontrado: {txt_path}")
        except OSError as e:
            print(f"Erro ao abrir o arquivo {txt_path}: {e}")

    combined_instructions += "\n\nReply TERMINATE at the end of your
    response."

    return combined_instructions
```

```
def order_trigger(sender):
    return "instruction & resources saved to" in
sender.last_message()["content"]

expert.register_nested_chats(
    [
        {
            "sender": expert,
            "recipient": expert_shadow,
            "message": order_message,
            "summary_method": "last_msg",
            "max_turns": 2,
            "silent": True,
        }
    ],
    trigger=order_trigger,
)

company = "Coca Cola"
competitors = ["PEP"]
fyear = "2023"

task = dedent(
    f"""
    Com as ferramentas fornecidas, escreva um relatório anual baseado no
    relatório 10-k de {company} e de {competitors} do ano fiscal {fyear}, e
    formate-o em um PDF.
    Preste atenção aos seguintes pontos:
    - Explique explicitamente seu plano de trabalho antes de começar.
    - Use as ferramentas uma por uma para maior clareza, especialmente ao
    pedir instruções.
    - Todas as operações de arquivo devem ser realizadas no diretório
    "{work_dir}".
    - Exiba qualquer imagem no chat assim que for gerada.
    - Para a análise de concorrentes, siga rigorosamente meu prompt e use
    dados apenas da tabela de métricas financeiras. Não use frases semelhantes
    de outras seções, exclua frases similares e classifique-as em uma das duas
    categorias. A última frase sempre deve discutir como o desempenho de
    {company} ao longo desses anos e métricas pode justificar ou contradizer
    sua avaliação de mercado atual (conforme refletido no índice EV/EBITDA).
    - Cada parágrafo da primeira página (visão geral do negócio, posição no
    mercado e resultados operacionais) deve ter entre 150 e 160 palavras. Cada
    parágrafo da segunda página (avaliação de riscos e análise de concorrentes)
    deve ter entre 500 e 600 palavras. Não gere o PDF até que isso seja
    explicitamente cumprido.
    """)
```

```
user_proxy.initiate_chat(recipient=expert, message=task, max_turns=50,  
summary_method="last_msg")
```

Predição de mercado

```
import autogen  
from autogen.cache import Cache  
  
from finrobot.utils import get_current_date, register_keys_from_json  
from finrobot.data_source import FinnHubUtils, YFinanceUtils  
from finrobot.data_source.finnhub_utils import FinnHubUtils  
  
flaml.automl is not available. Please install flaml[automl] to enable  
AutoML functionalities.  
  
config_list = autogen.config_list_from_json(  
    "./OAI_CONFIG_LIST",  
    filter_dict={"model": ["gpt-4o-mini"]},  
)  
llm_config = {  
    "config_list": config_list,  
    "timeout": 120,  
    "temperature": 0  
}  
  
register_keys_from_json("./config_api_keys")  
  
analyst = autogen.AssistantAgent(  
    name="Market_Analyst",  
    system_message="Como Analista de Mercado, é necessário possuir fortes  
habilidades analíticas e de resolução de problemas, coletar as informações  
financeiras necessárias e agregá-las com base nos requisitos do cliente."  
    "Para tarefas de codificação, utilize apenas as funções que lhe foram  
fornecidas. Responda TERMINATE quando a tarefa estiver concluída.",  
    llm_config=llm_config,  
)  
  
user_proxy = autogen.UserProxyAgent(  
    name="User_Proxy",  
    is_termination_msg=lambda x: x.get("content", "") and x.get(  
        "content", "").endswith("TERMINATE"),
```

```
human_input_mode="NEVER",  
max_consecutive_auto_reply=10,  
code_execution_config={  
    "work_dir": "coding",  
    "use_docker": False,  
},  
)
```

```
from finrobot.toolkits import register_toolkits
```

```
tools = [  
    {  
        "function": FinnHubUtils.get_company_profile,  
        "name": "get_company_profile",  
        "description": "get a company's profile information"  
    },  
    {  
        "function": FinnHubUtils.get_company_news,  
        "name": "get_company_news",  
        "description": "retrieve market news related to designated company"  
    },  
    {  
        "function": FinnHubUtils.get_basic_financials,  
        "name": "get_financial_basics",  
        "description": "get latest financial basics for a designated  
company"  
    },  
    {  
        "function": FinnHubUtils.save_markdown_to_pdf,  
        "name": "save_markdown_to_pdf",  
        "description": "Retrieve financial basics for a designated company  
and save them to a PDF file."  
    },  
    {  
        "function": YFinanceUtils.get_stock_data,  
        "name": "get_stock_data",  
        "description": "retrieve stock price data for designated ticker  
symbol"  
    }  
]
```

```
register_toolkits(tools, analyst, user_proxy)
```

```
company = "Ambev"
```

```
with Cache.disk() as cache:  
    user_proxy.initiate_chat(  
        analyst,  
        message=f"Use todas as ferramentas fornecidas para obter  
informações disponíveis sobre {company} em {get_current_date()}. Analise os  
desenvolvimentos positivos e as possíveis preocupações sobre {company} "  
        "com 4-6 fatores mais importantes, respectivamente, e mantenha-os  
concisos. A maioria dos fatores deve ser inferida de notícias relacionadas  
à empresa. "  
        f"Em seguida, faça uma previsão aproximada completa (por exemplo,  
aumento/diminuição em percentual) do movimento do preço das ações da  
{company} para a próxima semana. Forneça uma análise para embasar sua  
previsão contendo informações e valores. Mantenha a análise por volta de  
1000 tokens. Quando terminar a análise, gere um PDF com as informações.",  
        cache=cache,)
```

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 5 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Rafael Alves Goiás

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

Nesta Semana, finalizei todo o trabalho que havia planejado para o desenvolvimento na Residência, desde a concepção da ideia até a avaliação.

Como proposto, o trabalho dessa entrega consistiu em:

1. Solucionar um problema de cache ainda pendente, relacionado ao AutoGen (framework base para a criação do fluxo);
 2. Refinar os prompts e instruções fornecidos aos agentes, buscando maior precisão nas respostas;
 3. Desenvolver uma metodologia para avaliar a qualidade dos relatórios e análises gerados;
- **Solucionar um problema de cache:**
Agora, em cada iteração, o processo em ambos os contextos se reinicia, sem perder os materiais gerados e sem dar interferência nas gerações.
 - **Refinar os prompts e instruções fornecidos aos agentes:**
Os principais problemas eram a predição enviesada via prompt (O agente especialista sempre tendia a dizer que os ativos iam diminuir ao longo do tempo devido a estruturação do prompt) e a forma despadronizada de resposta.
 - **Metodologia para avaliar a qualidade dos relatórios e análises:**
 - Avaliando LLMs em cenários financeiros frente a diversas tarefas:
[📄 FinBen: Um Benchmark para LLM no Setor Financeiro](#)
 - Técnicas de sumarização abstrativa:
https://cookbook.openai.com/examples/evaluation/how_to_eval_abstractive_summarization
 - *RougeL*:
Mede a similaridade entre o resumo gerado e o texto fonte, focando em sequências de palavras mais longas em comum. Utilizado para avaliar a abrangência do conteúdo.

- **BERTScore:**
Baseado em embeddings de palavras, avalia a similaridade semântica entre o resumo e o texto fonte, garantindo uma análise mais profunda do significado.
- **G-EVAL:**
Uma abordagem mais robusta, avalia a qualidade dos resumos em múltiplas dimensões como relevância, coerência, consistência e fluência, atribuindo pontuações específicas para cada critério.
- **Avaliação Preditiva:**
Taxa de Acerto na Direção: 90% - Direção correta em todos os casos.
Precisão das Margens: 40% - Apenas dois ativos dentro das margens previstas.
Erro Médio Absoluto (MAE): 2.40 - Diferença média de 2.4 pontos percentuais.
Erro Percentual Absoluto Médio (MAPE): 46.14% - Erro absoluto médio em relação à variação real.

O sistema é eficiente em prever a direção, mas precisa melhorar a precisão das estimativas de amplitude.

Toda a parte de metodologia aplicada para a avaliação:

 Avaliação

Código da avaliação: [Avaliação](#)

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

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

Iago Alves Brito colaborou neste Stage.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: 

FinBen: Um Benchmark Holístico para Modelos de Linguagem de Grande Escala no Setor Financeiro

Os LLMs têm revolucionado o Processamento de Linguagem Natural, mas sua aplicação no setor financeiro ainda é limitada devido à falta de benchmarks abrangentes e à complexidade das tarefas. O **FinBen**, um benchmark de código aberto, cobre 36 conjuntos de dados e 24 tarefas financeiras, abordando sete áreas críticas como extração de informações, análise textual, perguntas e respostas, geração de texto, gestão de riscos, previsão e tomada de decisões.

O FinBen inova ao incluir avaliação de negociação de ações, métodos baseados em agentes e Recuperação-Baseada em Geração (RAG), além de três novos conjuntos de dados para tarefas específicas. A avaliação de 15 LLMs, incluindo GPT-4 e Gemini, revelou que:

- LLMs se destacam em extração de informações e análise textual, mas têm dificuldades com tarefas complexas, como geração de texto e previsão.
- GPT-4 é excelente em IE e negociação de ações, enquanto Gemini lidera em geração de texto e previsão.
- Modelos ajustados por instruções oferecem melhorias modestas para tarefas avançadas.

O FinBen engloba 36 conjuntos de dados e 24 tarefas financeiras, organizados em sete aspectos críticos:

- extração de informações (IE),
- análise textual (TA),
- perguntas e respostas (QA),
- geração de texto (TG),
- gestão de riscos (RM),
- previsão (FO)
- tomada de decisões (DM).

Essas categorias avaliam habilidades específicas no manuseio e análise de dados financeiros, destacando a capacidade dos LLMs em lidar com cenários financeiros complexos.

Principais Inovações do FinBen

- **Novas Tarefas:** O benchmark inclui a maior variedade de tarefas e conjuntos de dados já apresentada, permitindo uma avaliação robusta das capacidades dos LLMs em contextos financeiros diversificados.
- **Cobertura Ampla:** É o primeiro benchmark a avaliar negociação de ações, uma tarefa fundamental no setor financeiro que envolve decisões complexas que impactam o mercado e estratégias de investimento.
- **Nova Estratégia de Avaliação:** Introduce métodos de avaliação baseados em agentes e Recuperação-Baseada em Geração (RAG), refletindo interações dinâmicas e a recuperação de informações relevantes em grandes conjuntos de dados.
- **Conjuntos de Dados Inéditos:** Propõe três conjuntos de dados de código aberto inéditos para resumo de texto, QA e negociação de ações.
- **Incentivo à Pesquisa:** FinBen foi utilizado no workshop FinNLP-AgentScen (IJCAI-2024), atraindo 12 equipes que superaram o desempenho do GPT-4, demonstrando o potencial do benchmark para impulsionar a inovação no uso de LLMs financeiros.

Contribuições do FinBen

- Primeiro benchmark de avaliação abrangente e de código aberto para LLMs no setor financeiro.
- Nova taxonomia cobrindo sete aspectos das tarefas financeiras.
- Três conjuntos de dados inéditos para a comunidade de pesquisa.
- Avaliação sistemática de 15 LLMs, destacando suas limitações e pontos fortes, além de apontar direções para futuras pesquisas.

O desempenho zero-shot e few-shot de diferentes LLMs no FinBen. Todos os resultados das nossas avaliações são a média de três execuções. '-' representa o resultado que atualmente não pode ser obtido devido ao tamanho ou disponibilidade do modelo, e '' representa o resultado do artigo anterior.*

Dataset	Metrics	Chat GPT	GPT 4	Gemini	LLaMA2 7B-chat	LLaMA2 70B	LLaMA3 8B	FinMA 7B	FinGPT 7b-lora	InternLM 7B	Falcon 7B	Mixtral 7B	CFGPT sft-7B-Full
NER	EntityF1	0.77*	0.83*	0.61	0.18	0.04	0.08	0.69	0.00	0.00	0.00	0.24	0.00
FINER-ORD	EntityF1	0.28	0.77	0.14	0.02	0.07	0.00	0.00	0.00	0.00	0.00	0.05	0.00
FinRED	F1	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SC	F1	0.80	0.81	0.74	0.85	0.61	0.69	0.19	0.00	0.88	0.67	0.83	0.15
CD	F1	0.00	0.01	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FNXL	EntityF1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FSRL	EntityF1	0.00	0.01	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FPB	F1	0.78*	0.78*	0.77	0.39	0.73	0.52	0.88	0.00	0.69	0.07	0.29	0.35*
	Acc	0.78*	0.76*	0.77	0.41	0.72	0.52	0.88	0.00	0.69	0.05	0.37	0.26*
FiQA-SA	F1	0.60	0.80	0.81	0.76	0.83	0.70	0.79	0.00	0.81	0.77	0.16	0.42*
TSA	RMSE↓	0.53	0.50	0.37	0.71	0.57	0.25	0.80	0.00	0.29	0.50	0.50	1.05
Headlines	AvgF1	0.77*	0.86*	0.78	0.72	0.63	0.60	0.97	0.60	0.60	0.45	0.60	0.61*
	F1	0.64	0.71	0.40	0.35	0.49	0.40	0.49	0.00	0.36	0.30	0.37	0.16*
FOMC	Acc	0.6	0.69	0.60	0.49	0.47	0.41	0.46	0.00	0.35	0.30	0.35	0.21*
FinArg-ACC	MicroF1	0.50	0.60	0.31	0.46	0.58	0.51	0.27	0.00	0.39	0.23	0.39	0.05
FinArg-ARC	MicroF1	0.39	0.40	0.60	0.27	0.36	0.28	0.08	0.00	0.33	0.32	0.57	0.05
MultiFin	MicroF1	0.59	0.65	0.62	0.20	0.63	0.39	0.14	0.00	0.34	0.09	0.37	0.05
MA	MicroF1	0.85	0.79	0.84	0.70	0.86	0.34	0.45	0.00	0.78	0.39	0.34	0.25
MLESG	MicroF1	0.25	0.35	0.34	0.03	0.31	0.12	0.00	0.00	0.14	0.06	0.17	0.01
FinQA	EmAcc	0.58*	0.63*	0.00	0.00	0.06	0.00	0.04	0.00	0.00	0.00	0.00	0.00
TATQA	EmAcc	0.00*	0.13*	0.18	0.03	0.01	0.01	0.00	0.00	0.00	0.00	0.01	0.00
Regulations	Rouge-1	0.12	0.11	-	0.24	-	0.10	0.12	0.01	0.04	0.03	-	0.14
	BertScore	0.64	0.62	-	0.65	-	0.60	0.59	0.40	0.57	0.14	-	0.57
ConvFinQA	EmAcc	0.60*	0.76*	0.43	0.00	0.25	0.00	0.20	0.00	0.00	0.00	0.31	0.01
	Rouge-1	0.17	0.20	0.39	0.17	0.25	0.14	0.13	0.00	0.13	0.15	0.12	0.01
	BertScore	0.66	0.67	0.72	0.62	0.68	0.60	0.38	0.52	0.48	0.57	0.61	0.51
EDTSUM	BartScore	-3.64	-3.62	-3.87	-3.99	-3.81	-4.94	-5.71	-7.23	-4.60	-6.1	-4.47	-7.08
	Rouge-1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BertScore	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ECTSUM	BartScore	-5.18	-5.18	-4.93	-5.18	-4.86	-5.18	-5.18	-5.18	-5.18	-5.18	-5.18	-5.18
BigData22	Acc	0.53	0.54	0.55	0.54	0.47	0.55	0.51	0.45	0.56	0.55	0.46	0.45
	MCC	-0.025	0.03	0.04	0.05	0.00	0.02	0.02	0.00	0.08	0.00	0.02	0.03
ACL18	Acc	0.50	0.52	0.52	0.51	0.51	0.52	0.51	0.49	0.51	0.51	0.49	0.48
	MCC	0.005	0.02	0.04	0.01	0.01	0.02	0.03	0.00	0.02	0.00	0.00	-0.03
CIKM18	Acc	0.55	0.57	0.54	0.55	0.49	0.57	0.50	0.42	0.57	0.47	0.42	0.41
	MCC	0.01	0.02	0.02	-0.03	-0.07	0.03	0.08	0.00	-0.03	-0.06	-0.05	-0.07
German	F1	0.20	0.55	0.52	0.57	0.17	0.56	0.17	0.52	0.41	0.23	0.53	0.53
	MCC	-0.10	-0.02	0.00	0.03	0.00	0.05	0.00	0.00	-0.30	-0.07	0.00	0.00
Australian	F1	0.41	0.74	0.26	0.26	0.41	0.26	0.41	0.38	0.34	0.26	0.26	0.29
	MCC	0.00	0.47	0.00	0.00	0.00	0.00	0.00	0.11	0.13	0.00	0.00	-0.10
LendingClub	F1	0.20	0.55	0.65	0.72	0.17	0.10	0.61	0.00	0.59	0.02	0.61	0.05
	MCC	-0.10	-0.02	0.19	0.00	0.00	-0.15	0.00	0.00	0.15	-0.01	0.08	0.01
ccf	F1	0.20	0.55	0.96	0.00	0.17	0.01	0.00	1.00	1.00	0.10	0.00	0.00
	MCC	-0.10	-0.02	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ccfraud	F1	0.20	0.55	0.90	0.25	0.17	0.36	0.01	0.00	0.57	0.62	0.48	0.03
	MCC	-0.10	-0.02	0.00	-0.16	0.00	-0.03	-0.06	0.00	-0.13	-0.02	0.16	0.01
polish	F1	0.20	0.55	0.86	0.92	0.17	0.83	0.92	0.30	0.92	0.76	0.92	0.40
	MCC	-0.10	-0.02	0.14	0.00	0.00	-0.06	-0.01	0.00	0.07	0.05	0.00	-0.02
taiwan	F1	0.20	0.55	0.95	0.95	0.17	0.26	0.95	0.60	0.95	0.00	0.95	0.70
	MCC	-0.10	-0.02	0.00	-0.01	0.00	-0.07	0.00	-0.02	-0.01	0.00	0.00	0.00
portoseguro	F1	0.20	0.55	0.95	0.01	0.17	0.94	0.04	0.96	0.96	0.95	0.72	0.00
	MCC	-0.10	-0.02	0.00	-0.05	0.00	-0.01	0.01	0.00	0.00	0.00	0.01	0.00
travelinsurance	F1	0.20	0.55	0.00	0.00	0.17	0.00	0.00	0.98	0.89	0.77	0.00	0.03
	MCC	-0.10	-0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.12	-0.03	0.00	0.01

Avaliação e Ajustes finais:

- **Definição de Métricas de Avaliação:**

A avaliação de sistemas que integram LLMs (Large Language Models) e APIs para geração automática de relatórios e previsão de mercado apresenta uma série de desafios que abrangem múltiplas dimensões. A primeira dificuldade é a definição de métricas adequadas. Avaliar a qualidade textual, precisão preditiva e relevância das respostas é complexo, pois envolve tanto critérios objetivos quanto subjetivos. *Por exemplo, medir se as respostas são claras, úteis e bem estruturadas exige um equilíbrio entre avaliações humanas, que são caras e inconsistentes, e métodos automáticos, que podem não capturar nuances contextuais.*

- **Sinergia entre Modelos:**

Além disso, a interação entre dois LLMs introduz novos obstáculos. É essencial garantir alinhamento semântico entre os modelos, já que interpretações divergentes podem levar a respostas incoerentes. A propagação de erros é outra preocupação: um erro gerado por um modelo pode ser amplificado pelo outro. Também há implicações práticas, como a latência na troca de informações entre os modelos, que pode comprometer a experiência do usuário final.

- **APIs:**

A dependência de dados e APIs também é uma questão central. A qualidade das respostas depende diretamente da precisão e atualidade dos dados utilizados, especialmente em previsões de mercado, onde as informações mudam rapidamente. APIs externas podem apresentar limitações, como restrições de taxa, baixa confiabilidade ou inconsistência de dados, o que afeta a capacidade do sistema de fornecer saídas em tempo real. Além disso, a presença de vieses nos dados pode comprometer tanto a precisão quanto a equidade das previsões.

- **Explicabilidade e Transparência:**

Outro grande desafio é a explicabilidade do sistema. Modelos baseados em LLMs frequentemente funcionam como caixas-pretas, dificultando a rastreabilidade de erros ou a explicação de decisões específicas. Em contextos financeiros, onde relatórios e previsões podem ter implicações legais, essa opacidade é uma barreira significativa. Usuários e stakeholders frequentemente demandam transparência e querem entender como e por que o sistema chegou a determinadas conclusões.

- **Acurácia vs. Criatividade:**

Os LLMs também apresentam um dilema entre criatividade e acurácia. Enquanto a geração de texto precisa ser suficientemente inovadora para agregar valor, a predição de mercado exige precisão absoluta, o que pode ser prejudicado por alucinações comuns nos modelos, onde informações incorretas são apresentadas de forma plausível.

Do ponto de vista da usabilidade, o sistema precisa interpretar corretamente as queries dos usuários, muitas vezes complexas ou ambíguas, e personalizar os relatórios de acordo com o perfil de cada público, como investidores ou analistas financeiros. O feedback contínuo do usuário também é crucial para refinar e melhorar o sistema, mas captar e utilizar esse feedback de forma eficiente é um desafio em si.

- **Custo e Infraestrutura:**

Por fim, o custo computacional e a escalabilidade são preocupações práticas. A execução simultânea de dois LLMs integrados a APIs exige uma infraestrutura robusta e pode ser financeiramente custosa. Além disso, para lidar com mercados financeiros, o sistema precisa ser rápido e escalável, garantindo respostas em tempo real mesmo sob alta demanda.

Para superar esses desafios, a forma mais consistente é avaliar os resultados gerados de forma manual apoiado por especialistas da área.

Avaliando

ROUGE e BertScore:

ROUGE, que significa Recall-Oriented Understudy for Gisting Evaluation, mede principalmente a sobreposição de palavras entre uma saída gerada e um texto de referência. É uma métrica prevalente para avaliar tarefas de sumarização automática. Entre suas variantes, ROUGE-L oferece insights sobre a correspondência contígua mais longa entre resumos gerados pelo sistema e de referência, avaliando o quão bem o sistema retém a essência do resumo original.

O ROUGE depende da presença exata de palavras nos textos previstos e de referência, falhando em interpretar a semântica subjacente. É aqui que o BERTScore entra e alavanca os embeddings contextuais do modelo BERT, visando avaliar a similaridade entre uma frase prevista e uma de referência no contexto de texto gerado por máquina. Ao comparar embeddings de ambas as frases, BERTScore captura similaridades semânticas que podem ser perdidas por métricas tradicionais baseadas em n-gram.

Rouge-L:

Ele se baseia no Longest Common Subsequence (LCS), ou seja, mede o maior número de palavras na ordem correta que aparecem em ambos os textos.

- **Precisão:** Mede a proporção de palavras no texto gerado que aparecem na referência. Alta precisão indica que o texto gerado usa palavras que são relevantes em relação ao texto de referência.
- **Recall:** Mede a proporção de palavras no texto de referência que estão presentes no texto gerado. Alta lembrança indica que o texto gerado cobre bem o conteúdo da referência.
- **F1-Score:** É a média harmônica entre precisão e recall, equilibrando ambos.

A avaliação foi feita da seguinte forma:

Passamos por cada documento e geramos as métricas de precisão, recall e F1-score. Ao final, concatenamos essas métricas fazendo a média e geramos o resultado final.

Ao fazer isso para diferentes ativos (Apple, Nvidia, Bradesco, Ambev e AMD), calculamos a média novamente para termos um referencial do quão bom está a nossa geração.

Média de Precisão: 0.2497

Média de Recall: 0.0485

Média de F1-Score: 0.0772

- Média de Precisão (0.2497): Isso indica que cerca de 25% das palavras ou sequências geradas são consideradas relevantes em relação ao texto de referência. Não é muito alto, mas ainda pode significar que o modelo gera alguma similaridade com a referência.

- Média de Recall (0.0485): O valor baixo mostra que o texto gerado cobre apenas 4,85% das palavras ou sequências do texto de referência, indicando pouca sobreposição de conteúdo.
- Média de F1-Score (0.0772): Como a F1 combina precisão e recall, esse valor baixo reflete que o texto gerado está distante do ideal tanto em termos de uso de palavras relevantes quanto na cobertura do conteúdo.

Conclusão ROUGE-L: Esses valores sugerem que o texto gerado diverge significativamente do texto de referência, tanto em termos de palavras usadas quanto de conteúdo coberto. Pode ser que o modelo tenha dificuldade em replicar a estrutura e os conceitos do texto de referência. Outro ponto é que o resumo é muito inferior ao tamanho do texto original. Essas nuances comprometem a avaliação com essa métrica.

BERTScore:

O BERTScore é uma métrica baseada em modelos de linguagem como BERT. Ele mede a similaridade semântica entre o texto gerado e o texto de referência, indo além da simples comparação de palavras ao levar em conta o significado.

- Precisão: Mede o quanto cada palavra ou token no texto gerado está semanticamente próximo de alguma palavra no texto de referência.
- Recall: Mede o quanto cada palavra ou token no texto de referência está semanticamente próximo de alguma palavra no texto gerado.
- F1-Score: A média harmônica entre precisão e recall.

Interpretação dos resultados do BERTScore:

- Média da Precisão (0.5246): Um valor em torno de 52,46% indica que o texto gerado tem palavras que, semanticamente, estão razoavelmente alinhadas ao texto de referência.
- Média do Recall (0.4992): Esse valor, próximo ao da precisão, indica que o modelo cobre semanticamente quase 50% do texto de referência.
- Média do F1-Score (0.5116): Um F1 em torno de 51% mostra que há uma similaridade semântica moderada entre o texto gerado e a referência, mesmo que não seja perfeita.

Conclusão BERTScore: Embora o texto gerado não seja uma réplica exata do texto de referência, ele preserva boa parte do significado semântico, o que sugere que o modelo está capturando a ideia geral, mesmo que falhe em replicar a estrutura ou as palavras exatas. Essa métrica é mais condizente no nosso escopo do que a anterior por levar em consideração a semântica da frase.

Avaliando usando LLMs:

1. Relevância (Pontuação de 1 a 5)

A relevância refere-se à capacidade do resumo de selecionar e apresentar apenas o conteúdo importante do documento fonte. É crucial que o resumo destaque os pontos principais sem incluir informações redundantes ou irrelevantes.

- Critérios de Avaliação:

1 - Muito Baixa Relevância: O resumo não inclui os pontos principais e está repleto de informações irrelevantes ou redundantes.

2 - Baixa Relevância: O resumo cobre poucos pontos importantes e inclui muitas informações desnecessárias.

3 - Relevância Moderada: O resumo aborda alguns pontos principais, mas contém alguma redundância ou detalhes menos importantes.

4 - Alta Relevância: O resumo cobre a maioria dos pontos principais com pouca informação irrelevante.

5 - Muito Alta Relevância: O resumo inclui todos os pontos principais de forma concisa e sem redundâncias.

Passos de Avaliação:

- Leitura Atenta: Leia cuidadosamente o resumo e o documento fonte.
- Identificação dos Pontos Principais: Compare o resumo com o documento fonte para identificar os pontos-chave abordados.
- Análise de Conteúdo: Avalie quão bem o resumo cobre os pontos principais e verifique a presença de informações irrelevantes ou redundantes.
- Atribuição de Pontuação: Com base na análise, atribua uma pontuação de relevância de 1 a 5.

2. Coerência (Pontuação de 1 a 5)

A coerência está relacionada à organização e estrutura do resumo. Um resumo coerente apresenta as informações de forma lógica, com transições suaves entre as sentenças, construindo um texto que flui naturalmente.

- Critérios de Avaliação:

1 - Muito Baixa Coerência: O resumo é desorganizado, confuso e difícil de seguir.

2 - Baixa Coerência: O resumo apresenta alguma estrutura, mas as ideias não fluem logicamente.

3 - Coerência Moderada: O resumo é geralmente bem estruturado, mas pode ter transições abruptas ou pequenas inconsistências.

4 - Alta Coerência: O resumo é bem organizado, com ideias apresentadas em ordem lógica e conexões claras entre as sentenças.

5 - Muito Alta Coerência: O resumo é excelentemente estruturado, fluindo de maneira natural e lógica do início ao fim.

- Passos de Avaliação:

- Compreensão do Tópico Principal: Leia o documento fonte para entender o tópico central e os pontos-chave.
- Análise da Estrutura: Leia o resumo e avalie se as informações são apresentadas em uma ordem clara e lógica.
- Verificação da Fluidez: Observe se as transições entre as sentenças e parágrafos são suaves e contribuem para a compreensão do texto.
- Atribuição de Pontuação: Atribua uma pontuação de coerência de 1 a 5 com base nos critérios.

3. Consistência (Pontuação de 1 a 5)

A consistência refere-se à precisão factual do resumo em relação ao documento fonte. O resumo deve conter apenas informações que possam ser inferidas ou confirmadas pelo documento original, sem adicionar fatos incorretos ou inventados.

- Critérios de Avaliação:

1 - Muito Baixa Consistência: O resumo contém vários erros factuais significativos.

2 - Baixa Consistência: O resumo apresenta alguns erros factuais importantes.

3 - Consistência Moderada: O resumo é geralmente preciso, mas contém alguns erros factuais menores.

4 - Alta Consistência: O resumo é preciso, com possivelmente um ou outro detalhe mínimo incorreto.

5 - Muito Alta Consistência: O resumo é totalmente preciso e fiel ao documento fonte, sem erros factuais.

- Passos de Avaliação:

- Identificação dos Fatos Principais: Leia o documento fonte e destaque os principais fatos e detalhes.
- Comparação Factual: Leia o resumo e compare os fatos apresentados com os do documento fonte.
- Detecção de Inconsistências: Identifique quaisquer discrepâncias ou informações não suportadas pelo documento fonte.

- Atribuição de Pontuação: Atribua uma pontuação de consistência de 1 a 5 com base nos critérios.

4. Fluência (Pontuação de 1 a 3)

A fluência avalia a qualidade linguística do resumo, considerando gramática, ortografia, pontuação, escolha de palavras e construção de sentenças. O foco é garantir que o resumo seja claro, natural e fácil de ler.

- Critérios de Avaliação:

1 - Ruim: O resumo possui muitos erros que dificultam a compreensão ou fazem o texto soar não natural.

2 - Regular: O resumo tem alguns erros que podem afetar a clareza ou fluidez, mas os pontos principais ainda são compreensíveis.

3 - Bom: O resumo apresenta poucos ou nenhum erro, sendo fácil de ler e seguir.

- Passos de Avaliação:

- Leitura Crítica: Leia o resumo atentamente, observando a fluidez do texto.
- Análise Linguística: Avalie aspectos gramaticais, ortográficos, pontuação e escolha de palavras.
- Consideração da Língua: Não dê muita importância ao fato de o documento estar em inglês e o resumo em português; concentre-se na qualidade do resumo em si.
- Atribuição de Pontuação: Atribua uma pontuação de fluência de 1 a 3 conforme os critérios.

A avaliação do resumo considerou quatro critérios principais: Relevância, Coerência, Consistência e Fluência, resultando em uma nota geral de 3.75.

- Relevância (4.0): O resumo incluiu os pontos principais do documento fonte, com pouca informação irrelevante ou redundante.
- Coerência (4.0): Bem estruturado, o texto apresentou as ideias de forma lógica e fluida.
- Consistência (4.0): Precisão factual foi mantida, sem erros ou distorções em relação ao documento fonte.
- Fluência (3.0): O texto foi claro e bem escrito, com poucos ou nenhum erro gramatical.

Conclusão: O resumo atende bem aos padrões de qualidade, destacando-se em coerência e consistência

Predição:

O sistema foi avaliado com base em cinco ativos, considerando a previsão de direção, margem de variação e a variação real observada. As métricas calculadas incluem a taxa de acerto na direção, a precisão das margens, e os erros absolutos e percentuais. Abaixo está a síntese dos resultados:

Ativo	Direção Correta	Dentro da Margem	Pontuação Total	Erro Absoluto	APE (%)
Ambev	Sim	Não	2	6	63.64
Bradesco	Sim	Não	2	2	90.00
Nvidia	Sim	Não	2	3	43.75
AMD	Sim	Sim	3	1	0.00
Walmart	Sim	Sim	3	0	33.33

Métricas Gerais

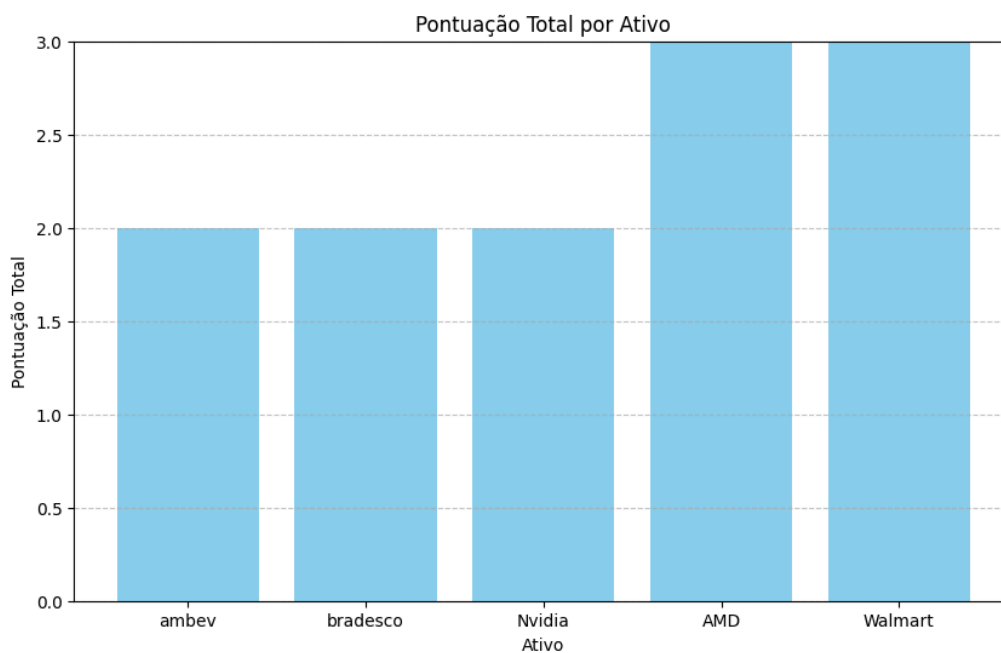
- Taxa de Acerto na Direção: 100.00%
O sistema acertou a direção (aumento ou diminuição) em todos os casos.
- Precisão das Margens: 40.00%
Apenas dois ativos tiveram a variação real dentro das margens previstas.
- Erro Médio Absoluto (MAE): 2.40
Em média, a diferença entre a variação real e a margem prevista mais próxima foi de 2.4 pontos percentuais.
- Erro Percentual Absoluto Médio (MAPE): 46.14%
Em média, o erro absoluto representou 46.14% da variação real.
- Análise Visual
Pontuação Total por Ativo:

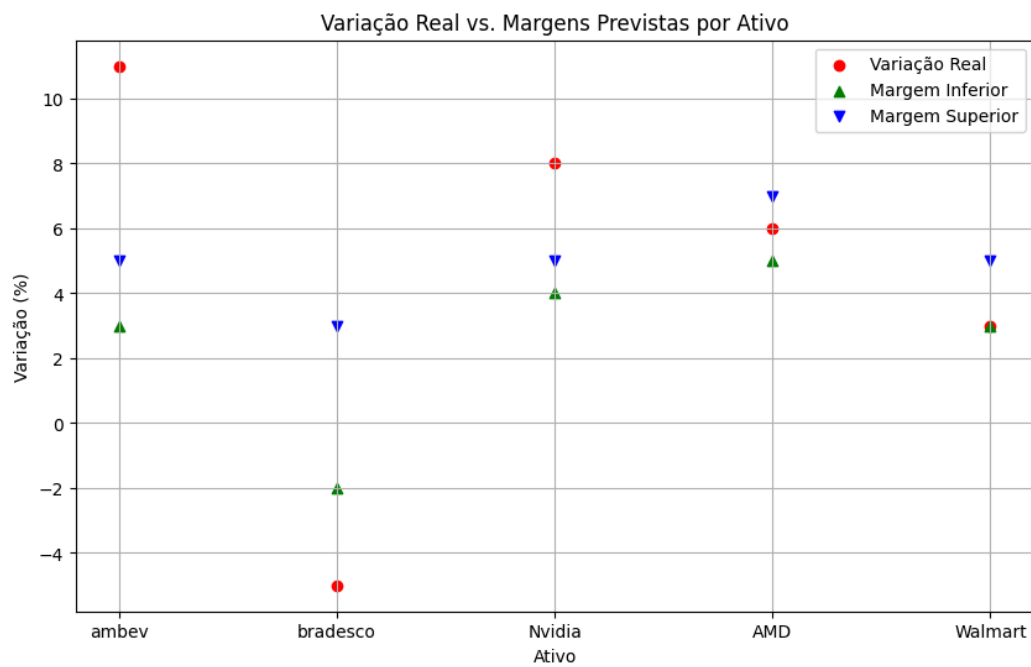
O gráfico de barras mostra que AMD e Walmart obtiveram a pontuação máxima de 3, indicando direção correta e variação dentro da margem.

Variação Real vs. Margens Previstas:

O gráfico de dispersão evidencia os ativos onde a variação real (em vermelho) ficou fora das margens inferior (verde) e superior (azul).

O sistema demonstrou alta capacidade de prever corretamente a direção das variações, com uma taxa de acerto de 100%. No entanto, a precisão das margens foi limitada a 40%, sugerindo espaço para melhorias nas estimativas de amplitude. A análise dos erros (MAE e MAPE) também indica que há margem para refinar a precisão das previsões.





Resultado Final

Geração de relatórios - Nvidia:

Equity Research Report: NVIDIA Corporation

Income Summarization

A receita total de \$26.97 bilhões permaneceu praticamente estável em relação ao ano anterior, com um crescimento de 41% na receita do Data Center, que alcançou \$15.068 bilhões, impulsionada pela demanda de clientes hyperscale e acordos de serviços em nuvem, enquanto a receita de Gaming caiu 27%, totalizando \$9,07 bilhões, refletindo a redução da demanda e o excesso de inventário. O custo de receita aumentou para \$11,132 bilhões, resultando em uma queda na margem bruta de 64,9% para 56,9%, devido a provisões de inventário de \$2,17 bilhões. O lucro líquido foi de \$4,368 bilhões, uma queda de 55%, e o EPS diluído caiu para \$1,74. O segmento de Compute & Networking mostrou resiliência, enquanto o segmento Graphics enfrentou desafios significativos. A empresa possui \$13,30 bilhões em caixa, o que é positivo, mas a pressão sobre a rentabilidade e as margens sugere riscos operacionais e de mercado que podem impactar o crescimento futuro.

Business Highlights

A NVIDIA Corporation atua na indústria de computação acelerada, focando em gráficos de PC, inteligência artificial (IA), ciência de dados, veículos autônomos e aplicações de metaverso. A empresa é reconhecida por sua liderança em unidades de processamento gráfico (GPUs). A NVIDIA se destaca pela inovação contínua, investindo mais de \$37 bilhões em pesquisa e desenvolvimento. Seus produtos, como a arquitetura de GPU Hopper e a plataforma NVIDIA AI, oferecem desempenho superior em comparação com soluções tradicionais. A demanda por IA e computação em nuvem está crescendo. Contudo, a empresa enfrenta desafios, como a volatilidade da demanda e a concorrência crescente de novos entrantes e gigantes da tecnologia. Em resposta às condições de mercado, a NVIDIA lançou a arquitetura Ada Lovelace e firmou parcerias para oferecer serviços de nuvem AI. A empresa também ajustou sua estratégia de vendas para reduzir o excesso de inventário.

Company Situation

A NVIDIA Corporation é uma empresa multinacional de tecnologia com sede em Santa Clara, Califórnia, que projeta e fabrica unidades de processamento gráfico (GPUs) e sistemas em chip (SoCs) para mercados de computação, automotivo e inteligência artificial. A empresa é reconhecida como líder em computação acelerada, oferecendo soluções que melhoram a eficiência e a performance em diversas aplicações. A NVIDIA está comprometida com a inovação e a pesquisa, investindo significativamente em desenvolvimento de produtos e tecnologias. A empresa também se destaca por suas parcerias estratégicas com empresas de tecnologia, ampliando seu alcance de mercado e potencial de crescimento.

Risk Assessment

Os três principais riscos da NVIDIA incluem: Concorrência Intensa: A competição crescente nos mercados-alvo pode resultar em perda de participação de mercado e redução de receitas, especialmente se os concorrentes oferecerem produtos mais baratos ou com melhores funcionalidades. Dependência de Fornecedores: A dependência de fornecedores terceirizados para fabricação e componentes pode comprometer a qualidade e a quantidade dos produtos, impactando a capacidade de atender à demanda do mercado. Riscos Regulatórios: A complexidade das leis e regulamentações, especialmente relacionadas a tecnologias emergentes como IA, pode resultar em custos adicionais e restrições operacionais, afetando a competitividade da empresa.

Financial Metrics

FY (USD mn)	2020	2021	2022	2023	2024
Operating Revenue	10918.0	16675.0	26914.0	26974.0	60922.0
Adjusted Net Profit	2796.0	4332.0	9752.0	4368.0	29766.0
Adjusted EPS	0.12	0.18	0.39	0.18	1.21
EBIT Margin	0.26	0.27	0.37	0.16	0.54
ROE	0.23	0.26	0.37	0.2	0.69
PE Ratio	52.36	75.43	62.68	109.09	51.83
EV/EBITDA	40.57	58.6	54.71	81.0	43.45
PB Ratio	12.0	19.34	22.97	21.56	35.89

FinRobot

<https://ai4finance.org/>

<https://github.com/AI4Finance-Foundation/FinRobot>

Report date: 2023-02-24

Key data

Rating	buy
Target Price	Failed to retrieve data: 403
6m avg daily vol (USDmn)	531.82
Closing Price (USD)	23.65
Market Cap (USDmn)	575263.00
52 Week Price Range (USD)	11.72 - 27.94
BVPS (USD)	0.89

Share Performance



PE & EPS



Predição de mercado - AMD:

Análise da AMD (Advanced Micro Devices Inc.) - 27 de Novembro de 2024

Informações da Empresa

A Advanced Micro Devices Inc. (AMD) é uma das principais empresas do setor de semicondutores, com uma capitalização de mercado de aproximadamente \$233,67 bilhões. A empresa é reconhecida por sua inovação e forte presença no mercado, especialmente em tecnologias relacionadas à inteligência artificial (IA).

Desenvolvimentos Positivos

- Crescimento no Setor de IA:** A AMD está sendo considerada uma das principais ações de crescimento no setor de IA, conforme destacado em relatórios recentes. A empresa está se posicionando bem em comparação com concorrentes como a Broadcom, atraindo o interesse de investidores.
- Inovações Tecnológicas:** Recentemente, a AMD lançou a versão beta do Amuse 2.2, que melhora a qualidade de imagem com a tecnologia Stable Diffusion 3.5. Isso demonstra o compromisso da empresa com a inovação e a melhoria contínua de seus produtos.
- Reconhecimento de Mercado:** A Goldman Sachs incluiu a AMD em sua lista de ações de crescimento, destacando-a como uma das principais opções para investidores que buscam crescimento no setor de tecnologia.
- Desempenho Financeiro Sólido:** A AMD apresentou um crescimento de receita de 20% em relação ao ano anterior, com uma margem líquida de 15%. Esses números indicam uma saúde financeira robusta e um potencial de crescimento contínuo.

Possíveis Preocupações

- Concorrência Intensa:** A AMD enfrenta forte concorrência de empresas como Nvidia, que recentemente reportou resultados financeiros impressionantes. A pressão competitiva pode impactar a participação de mercado da AMD.
- Volatilidade do Mercado:** O mercado de semicondutores é conhecido por sua volatilidade. Mudanças nas condições econômicas ou na demanda do consumidor podem afetar negativamente o desempenho da AMD.
- Dependência de Tecnologia de IA:** Embora a IA seja uma área de crescimento, a dependência excessiva dessa tecnologia pode ser arriscada se o mercado não se desenvolver conforme o esperado.
- Desafios na Cadeia de Suprimentos:** A AMD, como muitas empresas de tecnologia, pode enfrentar desafios na cadeia de suprimentos, que podem impactar a produção e a entrega de produtos.