Avaliação de modelos de linguagem compactos em execução offline: estudo do TinyLlama-1.1B-Chat

Vitor Amadeu Souza[1]; 0009-0002-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.
vitor.amadeu@foa.org.br

Resumo: Este estudo apresenta uma análise do modelo TinyLlama-1.1B-Chat-v1.0 operando em ambiente local completamente offline, utilizando VS Code e Python em máquina pessoal sem necessidade de conectividade de rede após o carregamento inicial. O trabalho investiga as capacidades conversacionais de um modelo de linguagem de pequeno porte (1.1 bilhão de parâmetros) em cenários de uso autônomo, avaliando sua precisão factual, coerência contextual e capacidade de manutenção de diálogo. A metodologia envolveu a implementação de um sistema de chat interativo com preservação de histórico e avaliação sistemática de respostas em domínios variados, executado localmente em ambiente de desenvolvimento integrado. Os resultados demonstraram que, apesar das limitações inerentes ao tamanho reduzido, o modelo apresentou desempenho satisfatório em questões factuais básicas, mantendo coerência conversacional e fornecendo respostas estruturadas. O sistema operou de forma autônoma após carregamento, demonstrando viabilidade para aplicações em ambientes com restrições de conectividade e necessidades de privacidade de dados. As análises revelaram precisão factual adequada em conhecimentos gerais, embora com limitações em detalhes específicos e contextos mais complexos.

Palavras-chave: Modelos de linguagem de pequeno porte. Sistemas conversacionais autônomos. Avaliação de desempenho. VS Code. Python.

INTRODUÇÃO

A democratização do acesso a modelos de linguagem de grande escala tem sido um dos principais desafios na área de Processamento de Linguagem Natural (PLN) nos últimos anos (Brown *et al*., 2020). Enquanto modelos como GPT-4 e LLaMA-2 demonstram capacidades impressionantes, suas demandas computacionais os tornam inacessíveis para muitas aplicações práticas (Touvron *et al*., 2023). Neste contexto, emerge a necessidade de modelos menores e mais eficientes que possam operar em ambientes locais com recursos limitados, mantendo capacidades conversacionais úteis e garantindo privacidade de dados.

O desenvolvimento de ambientes de programação integrados como o Visual Studio Code (VS Code) tem facilitado significativamente a implementação e teste de modelos de inteligência artificial em máquinas locais (Microsoft, 2023). A combinação de Python com bibliotecas especializadas como Transformers da Hugging Face permite que desenvolvedores e pesquisadores executem modelos de linguagem diretamente em suas estações de trabalho, sem depender de serviços em nuvem ou conectividade constante à internet (Wolf *et al*., 2020).

O TinyLlama-1.1B-Chat representa uma abordagem inovadora para esta problemática, oferecendo um modelo de linguagem com 1.1 bilhão de parâmetros especificamente otimizado para conversação (Zhang *et al*., 2024). Desenvolvido como uma versão compacta da arquitetura LLaMA, este modelo foi projetado para manter um equilíbrio entre eficiência computacional e qualidade de resposta, tornando-se uma alternativa viável para aplicações que requerem processamento local e autônomo.

A capacidade de operar completamente offline após o carregamento inicial representa uma vantagem significativa em cenários onde a conectividade é limitada ou onde questões de privacidade e segurança de dados são prioritárias (Goldberg, 2016). Aplicações em ambientes corporativos restritivos, dispositivos móveis com conectividade intermitente, ou situações que exigem processamento de informações sensíveis podem se beneficiar significativamente desta abordagem.

Este estudo tem como objetivo avaliar sistematicamente o desempenho do modelo TinyLlama-1.1B-Chat operando em ambiente local usando VS Code e Python, analisando suas capacidades conversacionais, precisão factual e coerência contextual em diferentes domínios do conhecimento. Através de uma metodologia estruturada de questionamentos e análise de respostas, busca-se compreender os limites e potencialidades deste modelo para aplicações práticas em cenários offline.

MÉTODOS

O experimento foi conduzido em uma estação de trabalho local equipada com GPU, utilizando o Visual Studio Code como ambiente de desenvolvimento integrado e Python 3.12.2 como linguagem principal. A escolha do VS Code justifica-se por sua ampla adoção na comunidade de inteligência artificial, oferecendo extensões especializadas para Python, integração com Jupyter notebooks e ferramentas avançadas de depuração (Gousios *et al*., 2015). Para a configuração do ambiente, foram instaladas as dependências principais: a biblioteca Transformers versão 4.55.4 da Hugging Face, PyTorch 2.8.0 com suporte CUDA para aceleração por GPU e Accelerate para otimização de carregamento de modelos. Além disso, adotou-se o tipo de dado bfloat16, de modo a reduzir o consumo de memória sem comprometer significativamente a precisão numérica, uma prática recomendada para execução de modelos de linguagem em hardware com limitações de VRAM (Dettmers *et al*., 2022).

O modelo utilizado foi o TinyLlama-1.1B-Chat-v1.0, carregado por meio do pipeline de geração de texto da biblioteca Transformers, com configuração de mapeamento automático de dispositivos para otimizar o uso da GPU. Com 1.1 bilhão de parâmetros distribuídos em uma arquitetura transformer modificada, o TinyLlama se baseia na estrutura do LLaMA, mas incorpora otimizações voltadas à redução de tamanho e à melhoria da eficiência (Zhang *et al*., 2024). Os parâmetros de geração foram ajustados cuidadosamente para equilibrar criatividade e coerência das respostas, sendo definidos como: máximo de 500 tokens novos por saída, sampling habilitado com temperatura de 0.7 para controle de aleatoriedade, top-p de 0.9 para nucleus sampling e top-k de 50 para restrição do conjunto de palavras

candidatas mais prováveis. Essa configuração segue as melhores práticas indicadas na literatura para modelos conversacionais (Holtzman *et al*., 2019).

O protocolo de avaliação foi estruturado a partir da formulação de três questões específicas em diferentes domínios: matemática (teorema de Pitágoras), ciências ambientais (energia renovável versus não-renovável) e história da arte (autoria da Mona Lisa). Essa seleção teve como objetivo avaliar tanto a capacidade do modelo em fornecer respostas factualmente corretas quanto sua habilidade em estruturar explicações em diferentes áreas do conhecimento. A análise considerou precisão factual, organização e clareza da resposta, presença de detalhes contextuais, coerência em relação ao histórico da conversa e adequação do nível de linguagem para o público geral. A avaliação foi conduzida de forma qualitativa, contemplando aspectos técnicos e comunicacionais.

O código-fonte está disponível para download através do link: https://github.com/vitor-souza-ime/tinyllama.

RESULTADOS E DISCUSSÃO

O sistema demonstrou capacidade de operação offline após o carregamento inicial do modelo, confirmando a viabilidade da abordagem para ambientes com restrições de conectividade. O tempo de carregamento inicial do modelo foi de aproximadamente dois minutos em hardware com GPU dedicada, após o qual todas as interações ocorreram sem necessidade de acesso à rede. Esse resultado é particularmente relevante para aplicações que requerem processamento local de dados sensíveis ou operação em ambientes com conectividade limitada. A utilização de memória GPU foi otimizada através do uso do tipo de dado bfloat16, resultando em consumo aproximado de 3.2 GB de VRAM durante a operação. Esse valor representa um uso eficiente de recursos considerando a capacidade do modelo de manter conversas coerentes e fornecer respostas factualmente precisas. A configuração de device_map="auto" permitiu distribuição automática do modelo entre GPU e CPU conforme necessário, garantindo operação estável mesmo em hardware com limitações de memória.

No que se refere à precisão factual, três questões foram utilizadas para avaliação. A primeira, sobre o teorema de Pitágoras, evidenciou que o modelo apresentou resposta exemplar ao

identificar corretamente Pitágoras como o matemático grego associado ao teorema e fornecer a formulação matemática precisa: “o quadrado da hipotenusa é igual à soma dos quadrados dos outros dois lados”. A resposta demonstrou compreensão adequada da terminologia geométrica, diferenciando hipotenusa de catetos e explicando sua aplicabilidade em triângulos retângulos. Além disso, o modelo apresentou contextualização histórica e aplicações práticas, confirmando observações de Zhao *et al*. (2023) sobre a competência de modelos de pequeno porte em lidar com domínios matemáticos bem estabelecidos.

Na segunda questão, referente às energias renováveis e não renováveis, o modelo conseguiu estabelecer distinção clara entre as duas categorias, identificando corretamente fontes renováveis como solar, eólica, hidrelétrica, geotérmica e biomassa, em contraste com carvão, petróleo e gás natural como não renováveis. A resposta trouxe uma explicação conceitual sólida ao afirmar que fontes renováveis estão sempre disponíveis e não se esgotam, enquanto as não renováveis podem se esgotar ao longo do tempo. O modelo incluiu ainda considerações ambientais relevantes, mencionando o impacto reduzido das renováveis e os problemas de poluição relacionados às não renováveis. Essa síntese e contextualização refletem a capacidade do modelo de articular conhecimentos multidisciplinares, em consonância com achados de Petroni *et al.* (2019) sobre retenção de conhecimento factual em modelos de linguagem.

A terceira questão, sobre a autoria da Mona Lisa, evidenciou novamente a precisão factual do TinyLlama. O modelo identificou corretamente Leonardo da Vinci como o autor, mencionou o período de criação da obra (1503-1506) e sua localização atual no Museu do Louvre, em Paris. De forma notável, incluiu também o nome alternativo “La Gioconda” e fez referência ao contexto histórico do período criativo de Da Vinci, descrevendo-o como um momento de intensa criatividade e curiosidade intelectual. Esse detalhamento sugere processamento sofisticado de conhecimento cultural e histórico, superando expectativas comuns para modelos de apenas 1.1 bilhão de parâmetros.

No que diz respeito à coerência conversacional, o sistema preservou adequadamente o histórico de interações e aplicou de forma consistente o template de chat ao longo do

experimento. Cada resposta foi produzida considerando o contexto acumulado, embora as questões fossem independentes entre si. As respostas seguiram uma estrutura consistente, composta por introdução, desenvolvimento detalhado e conclusão contextual, o que indica eficácia do treinamento conversacional em estruturar respostas apropriadas a diferentes tipos de pergunta. Além disso, a variação no comprimento das respostas, que oscilou entre 50 e 150 palavras, demonstrou adaptação proporcional à complexidade das questões apresentadas.

Por fim, as implicações práticas dos resultados sugerem que o TinyLlama-1.1B-Chat é viável para aplicações educacionais, sistemas de perguntas e respostas básicos e assistência em contextos com restrição de conectividade. Sua operação offline, combinada com precisão factual adequada, torna-o atrativo para cenários que demandam privacidade de dados ou ausência de internet constante. Em ambientes corporativos, pode atuar como assistente interno, base de conhecimento interativa ou ferramenta de treinamento para funcionários. A integração com VS Code, por sua vez, favorece a incorporação a fluxos de desenvolvimento já consolidados, permitindo customização e expansão conforme as necessidades de cada organização.

CONCLUSÕES

A capacidade de operação autônoma após carregamento inicial representa uma vantagem para aplicações que requerem processamento local, seja por questões de privacidade, conectividade limitada, ou restrições de segurança. A implementação em ambiente de desenvolvimento familiar como VS Code reduz barreiras de entrada para desenvolvedores e pesquisadores interessados em experimentar com modelos de linguagem locais.

As análises de precisão factual revelaram desempenho satisfatório em questões de conhecimento geral, com respostas estruturadas e contextualmente relevantes. O modelo demonstrou capacidade de elaboração além da mera repetição de fatos, incluindo contextualizações históricas e aplicações práticas que enriquecem o valor informacional das respostas.

Para trabalhos futuros, recomenda-se investigação de técnicas de fine-tuning específicas para domínios de aplicação, exploração de métodos de compressão adicional para reduzir

ainda mais os requisitos computacionais, e desenvolvimento de métricas automatizadas para avaliação sistemática de qualidade de resposta em modelos conversacionais de pequeno porte.

REFERÊNCIAS

BROWN, T. et al. Language models are few-shot learners. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 33 (NEURIPS 2020), 2020, Online. Proceedings... [S.l.: s.n.], 2020. p. 1877-1901.

TOUVRON, H. et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

WOLF, T. et al. Transformers: State-of-the-art natural language processing. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING: SYSTEM DEMONSTRATIONS, 2020, Online. Proceedings... Association for Computational Linguistics, 2020. p. 38-45.

ZHANG, P. et al. TinyLlama: An open-source small language model. arXiv preprint arXiv:2401.02385, 2024.

ZHAO, W. Z. et al. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023.

GOLDBERG, Yoav. A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research, v. 57, p. 345-420, 2016.

GOUSIOS, Georgios et al. Work practices and challenges in pull-based development: The integrator's perspective. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, 2015. p. 358-368.

DETTMERS, Tim et al. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. Advances in neural information processing systems, v. 35, p. 30318-30332, 2022.

HOLTZMAN, A. et al. The curious case of neural text degeneration. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, 7., 2019, New Orleans. Proceedings... [S.l.: s.n.], 2019.

PETRONI, F. et al. Language models as knowledge bases? In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING, 2019, Hong Kong. Proceedings... Association for Computational Linguistics, 2019. p. 2463-2473.