Stack Overflow na Era das LLMs: Tendências e Intenção do usuário

1º Aline Pinto
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
aline.cristina@dcc.ufmg.br

2º Michele Brandão
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
michele@dcc.ufmg.br

***Resumo*—O lançamento do ChatGPT no final de 2022 marcou um ponto de virada para as plataformas colaborativas de conhecimento. Este trabalho investiga o impacto dos Grandes Modelos de Linguagem (LLMs) no Stack Overflow, analisando dois grupos de linguagens de programação: Alto Recurso (Python, JavaScript, C#) e Moderado Recurso (Bash, Dart, R). Utilizando Séries Temporais Interrompidas (ITS) e modelagem de tópicos (BERTopic), são avaliadas as mudanças no volume de interações, na complexidade das perguntas e na evolução temática entre os períodos pré e pós-LLM. Os resultados mostram um declínio significativo na atividade, proporcionalmente maior nas linguagens de Alto Recurso, acompanhado de um leve aumento na extensão das perguntas em ambos os grupos. A análise temática revela estabilidade nos tópicos entre os períodos, com diferenças concentradas no volume de cada tema: tópicos triviais foram parcialmente absorvidos nas linguagens de Alto Recurso, enquanto o grupo de Moderado Recurso se manteve mais estável, sugerindo uma mudança na intenção do usuário em direção a perguntas mais complexas. Em conjunto, esses achados indicam que o Stack Overflow não está desaparecendo, mas sim se reestruturando.**

***Index Terms*—Stack Overflow, LLM, ChatGPT, Time Series Analysis, Topic Modeling.**

I. Introdução

Historicamente, o Stack Overflow desempenhou um papel central na engenharia de software como o principal local de conhecimento colaborativo [1]. No entanto, o lançamento do ChatGPT pela OpenAI em 2022 [2] introduziu uma mudança na área [3]. Capazes de sintetizar soluções e explicar código instantaneamente, os Grandes Modelos de Linguagem (LLMs) [4] apresentaram uma alternativa eficiente ao modelo assíncrono de fóruns, levando a reduções no tráfego dessas plataformas [5].

Entretanto esse fenômeno levanta uma questão mais sutil: as LLMs afetam todas as comunidades de programação de forma uniforme? Estudos indicam que a proficiência das LLMs é dependente do volume de dados de treinamento disponíveis, sendo elevada em linguagens populares como Python, mas inferior em linguagens de nicho, como Bash [6]. Isso sugere que comunidades menores podem ser impactadas de forma distinta, hipótese que este trabalho investiga.

Diante desse contexto, este trabalho busca analisar o impacto das LLMs para diferentes perfis de linguagens, conectando a análise quantitativa de tendências à investigação qualitativa dos tópicos. Por meio de Séries Temporais Interrompidas (ITS) e de modelagem de tópicos (BERTopic), são avaliadas as variações no volume de interações, as mudanças na complexidade das postagens e as categorias temáticas que continuam movimentando as comunidades a fim de entender se o Stack Overflow está desaparecendo ou se reinventando.

II. Trabalhos Relacionados

A ascensão dos Grandes Modelos de Linguagem (LLMs) impulsiona uma série de estudos sobre o futuro das plataformas de conhecimento colaborativo. Kabir et al. [7], por exemplo, comparam respostas da comunidade com as geradas pelo ChatGPT, concluindo que a IA frequentemente oferece soluções competitivas para problemas gerais. Esse fenômeno é abordado por Burtch et al. [8], que demonstram estatisticamente a redução na participação em comunidades online à medida que ferramentas generativas se tornam acessíveis.

Contudo, esse impacto pode não ser uniforme, visto que Cassano et al. [6] destacam que a proficiência das LLMs é fortemente dependente do volume de dados de treinamento disponíveis, sendo alta em linguagens populares (*High-Resource*) como Python, mas significativamente inferior em linguagens de menor recurso (*Low-Resource*). Essa limitação sugere que comunidades de nicho podem sofrer menos impacto, uma hipótese que este trabalho investiga ao segmentar a análise por grupos de linguagens.

Contrapondo a narrativa pessimista sobre o fim dessas comunidades, trabalhos mais recentes sugerem uma especialização do conteúdo remanescente. No artigo *"Stack Overflow Is Not Dead Yet"* [9] os autores argumentam que, embora o volume de perguntas triviais tenha diminuído, a necessidade de *crowd wisdom* permanece vital para problemas de alta complexidade e nuances contextuais que as LLMs falham em capturar.

Sendo assim, este estudo busca avançar nesse debate ao investigar o impacto das LLMs para diferentes tipos de linguagens, conectando a análise quantitativa de tendências com a investigação qualitativa dos tópicos para entender as mudanças do Stack Overflow em diferentes comunidades.

III. METODOLOGIA

A. Coleta e categorização dos dados

Os dados utilizados foram extraídos do *Stack Exchange Data Dump* [10], dataset público em XML disponibilizado pela *Stack Exchange Community* via *Archive.org*, cobrindo o período de janeiro de 2018 a junho de 2025 após limpeza e filtragem. O início em 2018 permite observar o comportamento da plataforma antes do impacto atípico da pandemia (2020), que foi excluído das regressões mas utilizado como referência visual, detalhado na Seção III-C.

Para avaliar o impacto das LLMs em diferentes contextos, as linguagens foram divididas em dois grupos com base na popularidade (volume de tags no Stack Overflow) [11] e na proficiência comparativa de LLMs na geração de código [12].

1) **Linguagens de Alto Recurso (Python, JavaScript, C#):** Linguagens com maior volume de tags na plataforma [11], com ampla representação nos dados de treinamento de LLMs e alta proficiência associada [12].
2) **Linguagens de Moderado Recurso (Bash, Dart, R):** Linguagens de nicho com volume de tags inferior [11] e desempenho de LLMs comparativamente menor [12].

B. Métricas quantitativas

Para analisar mudanças de tendência em perguntas e respostas, foram definidas três métricas principais: o volume semanal de novas perguntas (Y_{vol}), a extensão de código (L_{code}), medida pelo número de quebras de linha dentro da tag `<code/>`, e a extensão de texto (L_{text}), medida pelo número de quebras de linha sem as tags `<code/>`. As duas métricas de extensão foram aplicadas tanto a perguntas quanto a respostas, visando medir complexidade, sob o apoio de que perguntas mais elaboradas tendem a ter maior volume textual e de código [13]. Por apresentarem distribuição assimétrica, L_{code} e L_{text} foram normalizadas via $\log_{10}$ antes da análise estatística, reduzindo a influência de valores extremos na regressão.

C. Séries Temporais Interrompidas (ITS)

O modelo de Séries Temporais Interrompidas [14] (ITS, Eq. 1) foi aplicado a todas as métricas quantitativas:

$$Y_t = \beta_0 + \beta_1 T_t + \beta_2 X_t + \beta_3 (T_t \cdot X_t) + \varepsilon_t \qquad (1)$$

onde T_t é a tendência pré-intervenção, X_t é a variável binária de intervenção (0 = pré-LLM, 1 = pós-LLM) e ε_t o erro aleatório. Os coeficientes de interesse são a Mudança de Nível (β_2), que estima o impacto imediato do lançamento do ChatGPT, e a Mudança de Inclinação (β_3), que indica alteração na taxa de crescimento ao longo do tempo. A estimação foi feita por Mínimos Quadrados Ordinários com erros-padrão robustos a heterocedasticidade e autocorrelação (estimador de Newey-West [15], `maxlags=4`), valor escolhido pela regra de bolso de um lag por semana de sazonalidade mensal. As análises foram restritas a 2021-2024, excluindo o pico atípico da pandemia (2020) e equilibrando o intervalo pré e pós-intervenção em torno do lançamento do ChatGPT. Para respostas, o modelo foi aplicado às *respostas aceitas* e *todas as respostas*, visando distinguir se mudanças na extensão ocorrem de forma ampla ou apenas nas respostas aceitas pela comunidade.

D. Menção a IAs em respostas

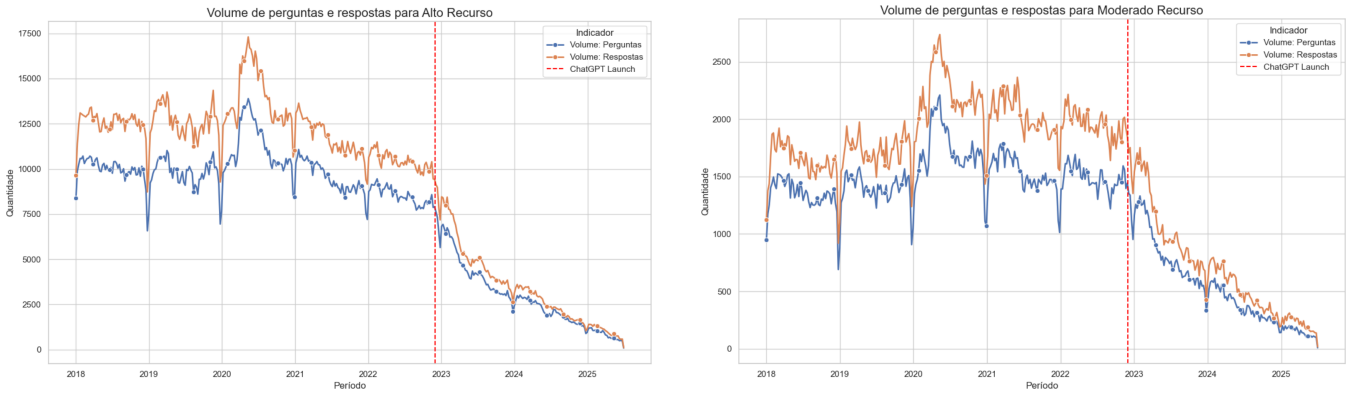
A influência das LLMs no conteúdo de respostas foi medida por meio de citações explícitas de ferramentas de IA, buscando observar variações entre linguagens e períodos (expresso em pontos percentuais). Para isso, foi realizada uma busca por expressão regular (*chatgpt*, variações de *gpt-3/4/5*, *copilot*, *openai*, *bard*, *gemini*, *claude*, *llm*/"large language model"), classificando cada resposta com valor 1 caso contivesse pelo menos uma ocorrência. Termos ambíguos como "bard" e "claude" foram exportados para revisão manual. Essa métrica não é encarada como indicador de geração de conteúdo por LLM, mas sim como uma possível temática em respostas.

E. Modelagem de tópicos e análise semântica

O BERTopic [16], técnica de modelagem de tópicos em Python, foi utilizado para investigar mudanças qualitativas de tópicos entre os períodos pré e pós-LLM. Para isso, foram amostradas de forma aleatória estratificada [17] 60.000 perguntas por linguagem (50% pré-LLM, 50% pós-LLM), restritas à janela 2020/1–2025/1 para equilibrar o volume de dados em torno da intervenção. Os embeddings foram gerados pelo modelo `all-mpnet-base-v2`, após remoção de HTML, código e stop words específicas de fóruns online. Por ser um modelo de propósito geral, `all-mpnet-base-v2` pode capturar com menor precisão as nuances de programação, limitação considerada na interpretação dos resultados.

Para os hiperparâmetros, o valor mínimo do cluster foi ajustado via busca em grade por linguagem e período, buscando o valor que produzisse o número de tópicos mais próximo de 10 antes da etapa de redução, otimizando o Silhouette Score [18]. Os tópicos foram então reduzidos [19] a no máximo 10 por modelo para viabilizar comparações entre linguagens. A nomeação foi gerada via *KeyBERTInspired* e refinada pelo *gpt-4o-mini*.

A evolução temporal foi analisada por abordagem seccionada: os tópicos foram gerados separadamente por período e comparados via similaridade de cosseno [20], após tentativas com Dynamic Topic Modeling [21] não produzirem resultados satisfatórios por priorizar tópicos dominantes no geral. Para identificar padrões transversais, todas as perguntas amostradas foram classificadas conjuntamente em um modelo unificado com $k = 8$ macro-grupos (busca em grade de $k = 4$ a 14). A qualidade dos modelos foi avaliada pelo Global Coherence Score (C_v) [22], que mede a coesão semântica entre os termos de um mesmo tópico, e pelo Silhouette Score, que avalia a separação entre clusters no espaço vetorial (valores próximos de 1 indicam separação clara).



(a) Grupo de Alto Recurso (Python, JavaScript, C#)

(b) Grupo de Moderado Recurso (R, Bash, Dart)

Figura 1: Comparativo do volume semanal de perguntas. A linha tracejada vertical marca o lançamento do ChatGPT (Nov/2022). Em ambos os grupos há queda no volume de perguntas e respostas, evidenciando diminuição de atividade na plataforma.

IV. Resultados e Discussão

A. O Impacto no Volume: Queda acentuada em linguagens de Alto Recurso

A Figura 1 apresenta a evolução do volume de perguntas e respostas para cada grupo entre 2018 e 2025. Visualmente, observa-se o aumento característico da pandemia (2020), seguido por estabilização e uma queda acentuada em ambos os grupos ao final de 2022. A aproximação entre as curvas de perguntas e respostas indica redução na quantidade de respostas por pergunta, padrão corroborado pelas taxas da Tabela A.1.

Tabela I: Regressão ITS: Volume de Perguntas (Y_{vol}).

Linguagem	**Nível** (β_2)	**Tendência** (β_3)	R^2_{adj}
Grupo Alto Recurso	**-1.840,14***	**-23,40***	**0,967**
Python	-1.048,77*	-13,06*	0,970
JavaScript	-631,27*	-5,56*	0,979
C#	-160,10*	-4,78*	0,954
Grupo Mod. Recurso	**-277,90***	**-7,57***	**0,927**
R	-120,99*	-3,78*	0,942
Dart	-110,87*	-2,72*	0,931
Bash	-46,04*	-1,07*	0,908

* Significativo a $p < 0.05$.

A regressão ITS (Tabela I) revela um contraste entre os grupos. O grupo de **Alto Recurso** sofreu queda imediata de ≈ -1.840 perguntas/semana, com aceleração de declínio ($\beta_3 = -23,40$) proporcionalmente maior. Python, que concentrava $\approx 51\%$ das perguntas do grupo no período pré-intervenção, respondeu sozinho por mais da metade da queda absoluta. O **Moderado Recurso**, embora também impactado, apresentou a retração mais suave ($\beta_3 = -7,57$), sugerindo substituição mais lenta.

A queda no volume é acompanhada por declínio no engajamento (Tabela A.1). No grupo de **Alto Recurso**, a taxa de resposta recuou em média $-12{,}8\%$, enquanto a taxa de resolução caiu aproximadamente o dobro, principalmente em Python ($-24{,}7\%$) e JavaScript ($-25{,}2\%$). No grupo de **Moderado Recurso**, as reduções foram mais contidas (média $-9{,}3\%$ na taxa de resposta), com exceção de Dart, que concentrou as maiores quedas do grupo tanto em resposta ($-16{,}4\%$) quanto em resolução ($-33{,}1\%$), sendo a maior do estudo. Contudo, esse resultado deve ser interpretado com cautela, visto que enquanto todas as demais linguagens já apresentavam redução no período pré-intervenção (2021–2022), Dart registrou crescimento de $+6{,}9\%$, reflexo da expansão da comunidade Flutter. Esse crescimento pode ter inflado a linha de base, amplificando as quedas observadas e não necessariamente por efeito direto das LLMs.

Assim, esses resultados sugerem que a intervenção não apenas reduziu o volume de interações, mas também enfraqueceu o ciclo de resolução, com impacto proporcionalmente maior nas linguagens de **Alto Recurso**.

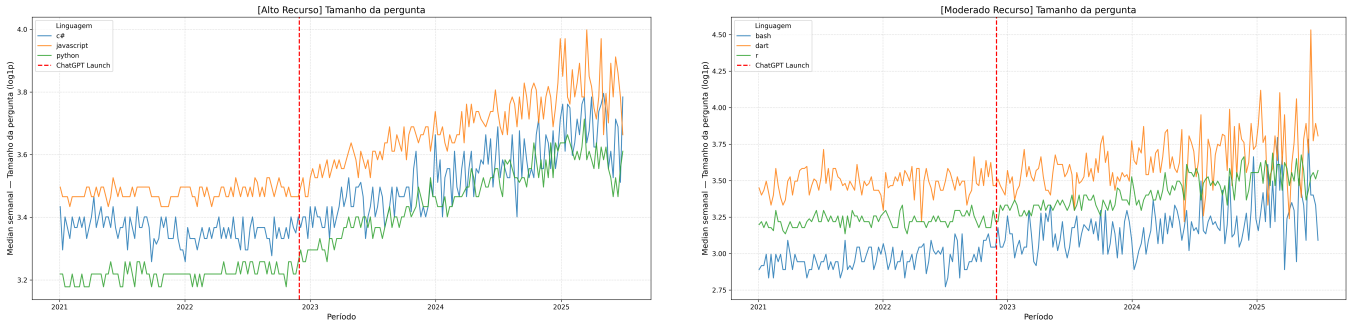
B. O Aumento da densidade de informação

Tabela II: Regressão ITS: Extensão de texto e código em perguntas

	Mudança imediata		Tendência pós-ChatGPT		R² ajustado	
Contexto / Linguagem	β_2 texto	β_2 código	β_3 texto	β_3 código	texto	código
Grupo Alto Recurso						
Python	0.0755*	0.0951*	0.0023*	0.0031*	0.9541	0.9381
C#	0.0362*	0.0410*	0.0030*	0.0033*	0.8909	0.7687
JavaScript	0.0435*	0.0671*	0.0026*	0.0024*	0.9213	0.8531
Grupo Moderado Recurso						
Bash	0.0470*	0.0862*	0.0015*	0.0020*	0.5431	0.5681
Dart	0.0015	−0.0088	0.0025*	0.0015*	0.4650	0.2224
R	0.0382*	0.0480*	0.0017*	0.0036*	0.7699	0.8677

* $p < 0.05$

Enquanto o volume de atividade entrou em queda, o conteúdo das perguntas mostrou uma leve tendência de aumento, como demonstrado na Figura 2. No grupo de **Alto Recurso**, todas as linguagens apresentaram aumento imediato significativo em texto e código, com ótimo ajuste de modelo ($R^2 = 0{,}77$–$0{,}95$). No grupo de **Moderado Recurso**, Bash e R também apresentaram mudança imediata significativa, enquanto Dart foi a única linguagem sem significância estatística



(a) Grupo de Alto Recurso (Python, JavaScript, C#)

(b) Grupo de Moderado Recurso (R, Bash, Dart)

Figura 2: Comparativo da extensão total das perguntas. A linha tracejada vertical marca o lançamento do ChatGPT (Nov/2022). Em ambos os grupos há aumento da extensão das perguntas em todas as linguagens.

e com ajuste inferior ($R^2_{\text{txt}} = 0{,}47$; $R^2_{\text{cod}} = 0{,}22$), comportamento consistente com o efeito de maturação discutido na Seção IV-A.

Em relação à tendência pós-intervenção, todas as linguagens apresentaram leve crescimento semanal contínuo e significativo ($p < 0{,}05$), com taxas médias de $0{,}26\%$–$0{,}29\%$ no **Alto Recurso** e $0{,}15\%$–$0{,}36\%$ no **Moderado Recurso**.

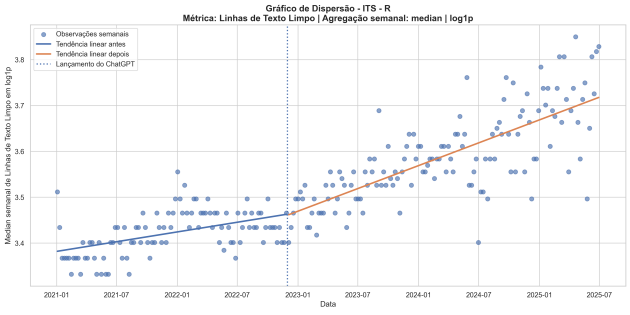


Figura 3: ITS para respostas agregadas por semana — linguagem R. Apesar de não haver mudança imediata (não há desconexão das retas), a inclinação da reta muda positivamente após o lançamento do ChatGPT, indicando tendência de aumento a longo prazo.

Para as respostas, o padrão observado é distinto. Mudanças imediatas foram mais pontuais, visto que no grupo de **Alto Recurso**, apenas Python (texto: $+2{,}49\%$) e C# (texto: $+5{,}08\%$; código: $+4{,}97\%$) em respostas aceitas, e C# (texto: $+3{,}78\%$) e JavaScript (texto: $+2{,}39\%$; código: $+3{,}30\%$) em respostas totais apresentaram essa tendência. No grupo de **Moderado Recurso**, nenhuma linguagem apresentou mudança imediata significativa. Em geral, todas as linguagens exibiram leve tendência de aumento pós-intervenção, com exceção de Bash e Dart.

Entretanto, o ajuste do modelo para respostas foi consistentemente inferior ao observado em perguntas, como ilustrado na Figura 3 para R. Esse comportamento pode ser reflexo da natureza heterogênea das respostas, que podem variar de um único link a blocos extensos de conteúdo, produzindo resíduos com alta variabilidade semanal que o modelo linear não ajusta bem. Esse efeito é mais evidente em linguagens com menor volume de respostas, como Dart e Bash, que apresentaram os maiores desvios residuais ($\sigma_\varepsilon \approx 0{,}10 - 0{,}20$) e os menores R^2, dado que com poucas observações semanais cada resposta atípica tem peso desproporcional na mediana agregada. Assim, as conclusões sobre extensão de respostas devem ser tratadas com baixa significância.

C. Menção a IAs em respostas

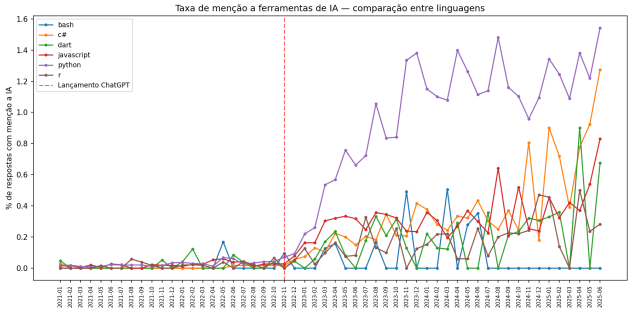


Figura 4: Série temporal comparativa de menções à IA.

A Figura 4 ilustra a evolução mensal das taxas de menção a IAs no conteúdo das respostas. Como esperado, antes de 2022, quase todas as linguagens convergem a zero citações, sendo os poucos pontos presentes referentes ao GitHub Copilot (2021), indicando que os dados observados estão associados ao período pós-ChatGPT. Após o lançamento, Python se destaca dos demais e mantém taxas entre 0,8% e 1,5% ao longo de 2023 e 2025, com variação de +0,771 p.p.. Já as outras linguagens crescem de forma mais modesta entre 0,1% e 0,4%. Bash por exemplo toca zero repetidas vezes no período pós-ChatGPT, registrando o menor crescimento (+0,059 p.p.), possivelmente pela natureza da linguagem e por ter menor volume de respostas, o que torna a ausência de menções esperada.

O ponto mais interessante dessa análise exploratória com checagem manual é ver que a maior parte dessas citações fo-

ram referentes a dicas, integrações possíveis e documentação, considerando o uso da IA como ferramenta. Contudo, as taxas absolutas de menção são baixíssimas em todos os grupos de linguagens (mesmo Python não atinge 1%), indicando pouca adoção da comunidade.

D. Evolução temática em perguntas

1) Evolução temática do grupo de Alto Recurso: Diferente do esperado, o grupo de Alto Recurso manteve grande parte de seus tópicos entre os períodos pré e pós-ChatGPT. A principal mudança foi a **fragmentação de tópicos amplos em subtemas mais específicos**, o que pode sugerir que as LLMs absorveram as perguntas mais genéricas, deixando espaço para que questões mais especificas aparecessem, dadas as limitações de redução de tópicos.

Python foi a linguagem mais estável do grupo: tópicos como *Data Filtering and Transformation* (pré) migraram para *Pandas DataFrame Operations* (pós), com alta similaridade. A mudança de nome e palavras-chave é um indício de que parte do conteúdo lógico foi absorvido e a prevalência de filtros está mais atrelada ao Pandas. O único tema com comportamento emergente foi *Type Checking*, com baixa similaridade aos demais, refletindo crescente atenção à qualidade de código.

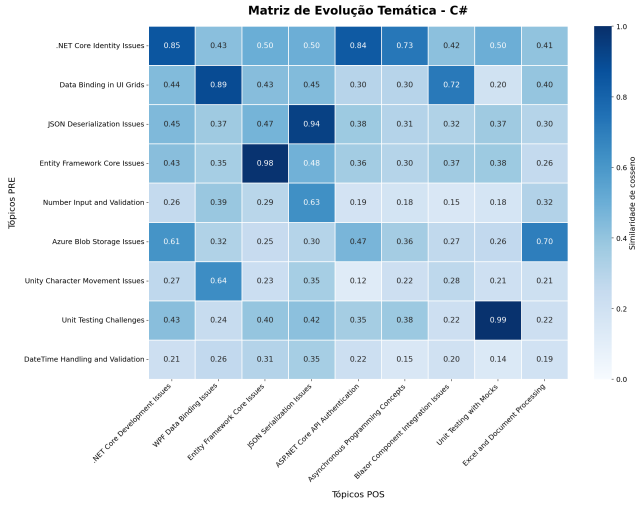


Figura 5: Matriz de similaridade para os tópicos de C# — alta fragmentação no período pós-ChatGPT.

C# exibe essa fragmentação com mais clareza (Figura 5): o tópico .NET Core Identity (pré) se divide em dois temas dedicados, autenticação e programação assíncrona, assuntos mais complexos que ganham espaço próprio no pós, enquanto temas triviais como Number Input/Validation desaparecem. JavaScript segue o caminho inverso consolidando vários temas do pré em um único bloco de React/TypeScript, padrão que pode refletir tanto a maturidade da comunidade quanto um sintoma da junção forçada pela redução de tópicos.

2) Evolução temática do grupo de Moderado Recurso: O grupo de Moderado Recurso mostrou continuidade quase integral, com especializações pontuais. R foi a linguagem mais estável de todo o estudo (Figura 6), com correspondência quase um para um entre os períodos. A única alteração relevante é *Shiny/RMarkdown*, que no pós aparece mais ligada à correção de erros. Já Dart preserva sua identidade em torno do Flutter enquanto Bash consolida temas distintos do pré em um único tópico de processamento de texto.

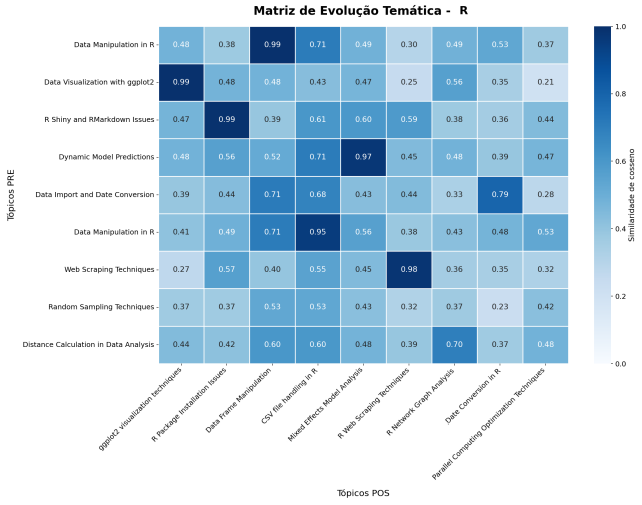


Figura 6: Matriz de similaridade para R.

3) Padrões evolutivos entre grupos: A Tabela III mostra as três macro-categorias compartilhadas pelos dois grupos: *Programação geral / Erros & ambiente*, *Dados & Visualização* e *Web scraping & Arquivos* (tópicos completos na Tabela A.4). Em todas elas se predominou a continuidade, com a diferença sendo de grau: o Alto Recurso apresentou maior rotatividade de vocabulário e subtemas, enquanto o Moderado se manteve quase estável.

Tabela III: Comportamento pós-ChatGPT por grupo de recurso nas macro-categorias comuns.

Categoria comum	**Alto recurso (Py, JS, C#)**	**Moderado recurso (R, Dart, Bash)**
Programação geral / Erros & ambiente	Maior especialização: surgem subtemas próprios (autenticação, *async*, verificação de tipos)	Especialização voltada a empacotamento/ambiente (instalação de pacotes em R, Docker/CI em Bash)
Dados & Visualização	Continuidade; refinamento de vocabulário (*pandas* mais explícito em Python)	Continuidade quase total (R: `dplyr`/`ggplot2` $\approx 0{,}99$)
Web scraping & Arquivos	Persistência com leve crescimento (scraping em Python; novos nichos em JS)	Persistência (manipulação de CSV e scraping em R)

Por fim, a qualidade dos modelos (Tabela IV) é coerente com os padrões observados. Os valores de c_v ($\approx 0{,}40$ – $0{,}60$) são razoáveis para textos curtos de fórum, enquanto os *Silhouette* modestos indicam que os temas formam mais uma continuidade do que blocos isolados. O efeito da redução

a dez tópicos é visível justamente nas linguagens com mais consolidação: JavaScript no pós tem o pior índice (0,19) e o maior *outlier rate* (40,4%), e Dart apresenta valores igualmente baixos. Python e R, as linguagens mais estáveis, exibem os melhores índices ($\approx 0{,}44 - 0{,}49$).

Tabela IV: Qualidade dos tópicos obtidos

Contexto / Linguagem	Período	Outlier rate (%)	Silhouette	c_v	Nº tópicos
Grupo Alto Recurso					
Python	Pré	29.80	0.4890	0.4102	10
Python	Pós	23.56	0.4445	0.4406	10
C#	Pré	31.65	0.2407	0.5604	9
C#	Pós	25.21	0.3384	0.5981	9
JavaScript	Pré	34.49	0.3331	0.4544	9
JavaScript	Pós	40.36	0.1863	0.5128	9
Grupo Moderado Recurso					
Bash	Pré	26.68	0.3697	0.4316	9
Bash	Pós	34.39	0.3458	0.4126	9
Dart	Pré	26.34	0.1408	0.4042	9
Dart	Pós	30.61	0.2077	0.4134	9
R	Pré	25.83	0.4154	0.5292	9
R	Pós	30.67	0.3562	0.4856	9

V. Conclusão

Este trabalho investigou o impacto das LLMs no Stack Overflow a partir de duas perspectivas complementares: a evolução quantitativa do volume e da complexidade das postagens, e a evolução qualitativa dos tópicos discutidos. Os resultados mostram um um cenário de reestruturação e não de declínio total. O impacto no volume foi significativo e assimétrico: linguagens de Alto Recurso sofreram quedas imediatas e aceleradas, enquanto o Moderado Recurso apresentou substituição mais lenta, coerente com a menor cobertura dessas linguagens nas ferramentas de IA. Ao mesmo tempo, o conteúdo das perguntas remanescentes se tornou levemente mais extenso em ambos os grupos, sinalizando possível aumento na complexidade média das dúvidas postadas.

Essa tendência é reforçada pela análise temática: tópicos triviais deram espaço para subtemas mais específicos no grupo de Alto Recurso, enquanto o grupo de Moderado Recurso manteve continuidade quase integral. Esse padrão conjunto aponta para uma mudança na intenção do usuário: quem permanece recorrendo ao Stack Overflow tende a usá-lo para questões mais contextuais, complexas ou simplesmente para obter uma opinião humana. Além disso, as menções explícitas a IA em respostas foram baixíssimas em todas as linguagens analisadas, indicando que a adoção desse tópico foi bastante limitada.

Por fim, os resultados deste trabalho devem ser compreendidos dentro de algumas limitações. O comportamento atípico de Dart dificulta separar o efeito das LLMs do crescimento orgânico da comunidade, enquanto o ajuste inferior dos modelos ITS para respostas reduz a força das inferências nesse subconjunto. Além disso, a redução forçada de tópicos e a remoção de código no pré-processamento podem ter mesclado ou descartado informações semanticamente relevantes, indicando melhorias importantes para trabalhos futuros.

Apêndice

Tabela A.1: Mudança nas taxas de resposta e resolução das perguntas

	Taxa de resposta			**Taxa de resolução**		
Contexto / Linguagem	Pré (%)	Pós (%)	Var. (%)	Pré (%)	Pós (%)	Var. (%)
Grupo Alto Recurso						
Python	82.32	70.80	-14.00	45.44	34.24	-24.65
C#	77.58	69.86	-9.95	42.35	36.07	-14.83
JavaScript	81.14	69.53	-14.31	44.12	33.02	-25.16
Grupo Moderado Recurso						
Bash	84.85	81.03	-4.50	51.43	46.75	-9.10
Dart	85.72	71.65	-16.42	45.03	30.11	-33.13
R	83.29	77.41	-7.06	55.01	49.43	-10.14

Tabela A.2: Mudança no padrão do número de linhas (normalizado em log) — Respostas aceitas

	Mudança imediata		Tendência pós-ChatGPT		R^2 ajustado	
Contexto / Linguagem	β_2 texto	β_2 código	β_3 texto	β_3 código	texto	código
Grupo Alto Recurso						
Python	0.0246^*	0.0134	0.0018^*	0.0012^*	0.8357	0.6799
C#	0.0495^*	0.0485^*	0.0022^*	0.0022^*	0.5678	0.2488
JavaScript	0.0113	0.0157	0.0018^*	0.0018^*	0.6419	0.4720
Grupo Moderado Recurso						
Bash	-0.0256	-0.0269	0.0025^*	0.0020^*	0.3428	0.2308
Dart	0.0155	-0.0409	0.0023^*	0.0012^*	0.1764	0.0578
R	-0.0066	-0.0010	0.0015^*	0.0009^*	0.6585	0.6571

* $p < 0.05$

Tabela A.3: Mudança no padrão do número de linhas (normalizado em log) — Todas as respostas

	Mudança imediata		Tendência pós-ChatGPT		R^2 ajustado	
Contexto / Linguagem	β_2 texto	β_2 código	β_3 texto	β_3 código	texto	código
Grupo Alto Recurso						
Python	0.0167	0.0039	0.0011^*	0.0004^*	0.8420	0.5083
C#	0.0371^*	0.0369	0.0019^*	0.0020^*	0.6171	0.2211
JavaScript	0.0236^*	0.0325^*	0.0010^*	0.0007^*	0.5497	0.2324
Grupo Moderado Recurso						
Bash	0.0213	0.0089	0.0014^*	0.0002^*	0.4718	0.3752
Dart	0.0128	0.0033	0.0011^*	-0.0006^*	0.1459	-0.0027
R	-0.0016	0.0026	0.0011^*	0.0004^*	0.7612	0.7225

* $p < 0.05$

Tabela A.4: Macro-tópicos gerais identificados.

#	Tópico	Escopo	Linguagens
0	Mobile	UI/estado Flutter, navegação, animação, Firestore, JSON, data	dart (16), c# (1), js (1)
1	Programação geral	Erros, ambiente, Docker, async, validação, bots	python (13), bash (3), c# (3), dart (2), js (1)
2	Dados & Viz.	DataFrames, transformação, plotagem (ggplot2/matplotlib), CSV, modelagem	r (16), python (4), js (1)
3	Web scraping & Arquivos	Scraping web, upload de arquivos, planilhas, Azure Blob	js (3), c# (2), python (2), r (2)
4	.NET / Tipagem	.NET Core, EF, WPF/Blazor, ASP.NET, JSON, tipos	c# (10), python (1)
5	Shell scripting	Bash, sed/grep, jq, SSH, automação (Jenkins/Git), arquivos	bash (15), js (1)
6	Testes	Testes unitários, mocks, frameworks	c# (2), js (2)
7	Frontend JS	React, estado, async, formulários, SVG, Firebase	js (9)

Tabela A.5: Tópicos: Python (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
Data Filtering & Transformation [Dados & Visualização]	4762(22.7%)	**Pandas DataFrame Operations** [Dados & Visualização]	3093(13.5%)
String Manipulation & Parsing [Programação geral]	4506(21.5%)	**Data Structure Transformation** [Programação geral]	4462(19.5%)
Web Scraping Techniques [Web scraping & Arquivos]	1939 (9.2%)	**Web Scraping Issues** [Web scraping & Arquivos]	4851(21.2%)
Django & SQLAlchemy Issues [Web scraping & Arquivos]	1860 (8.9%)	**Discord Bot Development** [Programação geral]	779 (3.4%)
Python Environment Issues [Programação geral]	1689 (8.0%)	**Python Installation Errors** [Programação geral]	2894(12.6%)
Plotting & Visualization [Dados & Visualização]	1675 (8.0%)	**Data Visualization Techniques** [Dados & Visualização]	1219 (5.3%)
TensorFlow & PyTorch Issues [Dados & Visualização]	1639 (7.8%)	**Model Training Errors** [Dados & Visualização]	1807 (7.9%)
GUI Development with Python [Programação geral]	1341 (6.4%)	**GUI Automation & Control** [Programação geral]	1745 (7.6%)
Parallel Processing [Programação geral]	917 (4.4%)	**Socket Communication** [Programação geral]	990 (4.3%)
Numpy Array Manipulation [Dados & Visualização]	662 (3.2%)	**Python Type Checking** [.NET / Tipagem & Serialização]	1053 (4.6%)

Tabela A.6: Tópicos: C# (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
.NET Core / Identity [Ecossistema .NET]	6533(31.9%)	**.NET Core Development** [Ecossistema .NET]	5016(22.4%)
Data Binding in UI Grids [Ecossistema .NET]	4197(20.5%)	**WPF Data Binding** [Ecossistema .NET]	4579(20.4%)
JSON Deserialization [Tipagem & Serialização]	2750(13.4%)	**JSON Serialization** [Tipagem & Serialização]	3083(13.7%)
Entity Framework Core [Ecossistema .NET]	2344(11.5%)	**Entity Framework Core** [Ecossistema .NET]	3143(14.0%)
Number Input & Validation [Programação geral]	1390 (6.8%)	**ASP.NET Core API Auth** † [Ecossistema .NET]	2768(12.3%)
Azure Blob Storage [Web scraping & Arquivos]	1272 (6.2%)	**Async Programming** † [Programação geral]	1786 (8.0%)
Unity Character Movement ∗ [Programação geral]	1083 (5.3%)	**Blazor Components** [Ecossistema .NET]	888 (4.0%)
Unit Testing Challenges [Testes de unidade]	523 (2.6%)	**Unit Testing with Mocks** [Testes de unidade]	617 (2.8%)
DateTime Handling ∗ [Programação geral]	378 (1.8%)	**Excel & Document Processing** [Web scraping & Arquivos]	546 (2.4%)

† tema emergente/especializado no pós. ∗ tema sem par forte no pós.

Tabela A.9: Tópicos: R (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
Data Manipulation (dplyr) [Dados & Visualização]	6056(27.3%)	**Data Frame Manipulation** [Dados & Visualização]	4492(21.6%)
Visualization (ggplot2) [Dados & Visualização]	5147(23.2%)	**ggplot2 Visualization** [Dados & Visualização]	5491(26.4%)
Shiny & RMarkdown [Programação geral]	3783(17.0%)	**Package Installation Issues** [Programação geral]	4825(23.2%)
Model Predictions (glm) [Dados & Visualização]	2177 (9.8%)	**Mixed Effects Models** [Dados & Visualização]	1641 (7.9%)
Data Import & Date [Dados & Visualização]	1917 (8.6%)	**Date Conversion** [Programação geral]	419 (2.0%)
Data Manipulation (apply/loop) [Dados & Visualização]	1620 (7.3%)	**CSV File Handling** [Web scraping & Arquivos]	2395(11.5%)
Web Scraping (rvest) [Web scraping & Arquivos]	837 (3.8%)	**Web Scraping (RSelenium)** [Web scraping & Arquivos]	698 (3.4%)
Random Sampling ∗ [Dados & Visualização]	347 (1.6%)	**Network Graph (igraph)** ∗ [Dados & Visualização]	459 (2.2%)
Distance / Spatial Data [Dados & Visualização]	338 (1.5%)	**Parallel Computing** ∗ [Programação geral]	372 (1.8%)

∗ tema marginal sem par forte. Demais: correspondência forte (≥ 0.95).

Tabela A.7: Tópicos: Dart (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
Flutter UI & State [Mobile]	5311(31.4%)	**Flutter App Dev / Build** [Mobile]	5919(37.1%)
Flutter API & JSON [Mobile]	4152(24.5%)	**Flutter JSON Serialization** [Mobile]	2962(18.6%)
Flutter Build & Errors [Mobile]	4042(23.9%)	**Scroll & Animation** † [Mobile]	2629(16.5%)
Firestore Data Retrieval [Mobile]	1667 (9.9%)	**State Management (BLoC)** † [Mobile]	1561 (9.8%)
Date & Time Parsing [Programação geral]	465 (2.7%)	**Navigation** † [Mobile]	993 (6.2%)
TextField Issues [Mobile]	424 (2.5%)	**Firestore Data Retrieval** [Mobile]	784 (4.9%)
Error Handling [Programação geral]	312 (1.8%)	**Date & Time Formatting** [Programação geral]	431 (2.7%)
Video Playback ∗ [Mobile]	283 (1.7%)	**Custom Color Themes** ∗ [Mobile]	368 (2.3%)
Maps & Geolocation ∗ [Mobile]	264 (1.6%)	**Error Handling** [Programação geral]	319 (2.0%)

† especializado a partir de "Flutter UI & State". ∗ tema marginal sem par forte.

Tabela A.8: Tópicos: Bash (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
Shell Scripting & Automation [Shell scripting]	1510(21.0%)	**Text Processing (sed/grep/awk)** ‡ [Shell scripting]	3053(47.5%)
Bash + Docker / Env [Shell scripting]	1474(20.5%)	**Bash Scripting in Jenkins** [Shell scripting]	763 (11.9%)
File & Directory Management [Shell scripting]	1060(14.8%)	**Process Output Handling** [Shell scripting]	718 (11.2%)
Quoting / Variables [Shell scripting]	811 (11.3%)	**Environment Configuration** † [Programação geral]	436 (6.8%)
Text Processing Techniques [Shell scripting]	673 (9.4%)	**Docker Configuration** † [Programação geral]	354 (5.5%)
Shell Scripting Issues [Shell scripting]	618 (8.6%)	**JSON Processing (jq)** [Shell scripting]	291 (4.5%)
sed String Replacement [Shell scripting]	569 (7.9%)	**SSH & Shell Scripting** [Shell scripting]	280 (4.4%)
jq JSON Manipulation [Shell scripting]	255 (3.6%)	**Git Commit Automation** ∗ [Shell scripting]	276 (4.3%)
Curl Command Usage [Shell scripting]	208 (2.9%)	**Curl / API & Downloads** [Shell scripting]	257 (4.0%)

‡ absorveu vários temas (consolidação). † especializado a partir de "Bash + Docker / Env". ∗ tema emergente sem par forte.

Tabela A.10: Tópicos: JavaScript (Pré vs Pós)

PRE-LLM		POST-LLM	
Tópico	**Vol. (%)**	**Tópico**	**Vol. (%)**
Form & Table / DOM [Frontend / JS web]	4607(23.5%)	**React State Management** ‡ [Frontend / JS web]	6455(36.1%)
Component State (React) [Frontend / JS web]	4202(21.4%)	**Web Development / Tooling** † [Programação geral]	4387(24.5%)
Date & Time Manipulation [Programação geral]	2743(14.0%)	**Scroll & Animation** [Frontend / JS web]	2216(12.4%)
Array & Object Manipulation [Programação geral]	2481(12.6%)	**Async Programming** † [Programação geral]	1622 (9.1%)
SVG Animation & Positioning [Frontend / JS web]	1997(10.2%)	**Browser File Handling** [Web scraping & Arquivos]	1223 (6.8%)
Firebase Auth / CORS / Async [Frontend / JS web]	1940 (9.9%)	**Chart Rendering** [Dados & Visualização]	661 (3.7%)
File Upload & Processing [Web scraping & Arquivos]	769 (3.9%)	**Date & Time Handling** [Programação geral]	543 (3.0%)
Discord Bots (Node) ∗ [Programação geral]	493 (2.5%)	**JavaScript Testing** [Testes de unidade]	465 (2.6%)
Testing JS Applications [Testes de unidade]	386 (2.0%)	**Google Sheets Script** ∗ [Web scraping & Arquivos]	304 (1.7%)

‡ absorveu vários temas (consolidação). † tema especializado no pós. ∗ tema sem par forte.

REFERÊNCIAS

[1] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web? nier track," in *Proceedings of the 33rd International Conference on Software Engineering*. IEEE, 2011, pp. 804–807.

[2] OpenAI, "Introducing chatgpt," https://openai.com/blog/chatgpt, 2022, acessado em: 2025-01-15.

[3] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, "Large language models for software engineering: A survey," *arXiv preprint arXiv:2308.10620*, 2023.

[4] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023, acessado em: 2025-09-08. [Online]. Available: https://arxiv.org/abs/2303.18223

[5] S. Kabir, D. N. Udo-Imeh, B. Kou, and T. Y. Zhang, "Who answers it better? an in-depth analysis of chatgpt and stack overflow answers to software engineering questions," *arXiv preprint arXiv:2308.02312*, 2023.

[6] F. Cassano, J. Gouwar, F. Lucchetti, C. Schlesinger, C. J. Anderson, M. Q. Feldman, M. Greenberg, A. Jangda, and A. Guha, "Knowledge transfer from high-resource to low-resource programming languages for code llms," *Proceedings of the ACM on Programming Languages (PACMPL)*, vol. 8, no. OOPSLA2, 2024.

[7] S. Kabir, D. N. Udo-Imeh, B. Kou, and T. Zhang, "Is stack overflow obsolete? an empirical study of the characteristics of chatgpt answers to stack overflow questions," in *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*, 2024, pp. 1–17, accessed: 2025-09-08. [Online]. Available: https://dl.acm.org/doi/full/10.1145/3613904.3642596

[8] G. Burtch, D. Lee, and Z. Chen, "The consequences of generative ai for online knowledge communities," *Scientific Reports*, vol. 14, no. 1, p. 10413, 2024, accessed: 2025-09-08. [Online]. Available: https://www.nature.com/articles/s41598-024-61221-0

[9] T. S. Denis Helic, "Stack overflow is not dead yet: Crowd answers still matter," https://arxiv.org/html/2509.05879v1, 2025.

[10] Stack Exchange, "Stack Exchange Data Dump," https://archive.org/details/stackexchange_20250630_rev2, Jun. 2025, acesso em: 30 nov. 2025.

[11] Stack Overflow, "Tags – Stack Overflow," https://stackoverflow.com/tags, 2025.

[12] Continue.dev, "LLMs are helpful with Python, but what about all of the other programming languages?" https://continue.ghost.io/programming-languages/, Aug. 2023.

[13] F. Calefato, F. Lanubile, and N. Novielli, "How to ask for technical help? Evidence-based guidelines for writing questions on Stack Overflow," *Information and Software Technology*, vol. 94, pp. 186–207, 2018.

[14] DS4PS, "Interrupted time series analysis," https://ds4ps.org/pe4ps-textbook/docs/p-020-time-series.html, 2019, acessado em: 2025-12-02.

[15] W. K. Newey and K. D. West, "A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix," *Econometrica*, vol. 55, no. 3, pp. 703–708, 1987.

[16] M. Grootendorst, "Bertopic," https://maartengr.github.io/BERTopic/api/bertopic.html, 2025, acessado em: 2025-12-02.

[17] W. G. Cochran, *Sampling Techniques*, 3rd ed. New York: John Wiley & Sons, 1977.

[18] GeeksforGeeks, "What is silhouette score?" https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/, 2024, acessado em: 2 dez. 2025.

[19] M. Grootendorst, "BERTopic api documentation: reduce_topics," https://maartengr.github.io/BERTopic/api/bertopic.html#bertopic._bertopic.BERTopic.reduce_topics, 2025, acessado em: 2025-12-02.

[20] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008, capítulo 6: Scoring, term weighting and the vector space model.

[21] M. Grootendorst, "BERTopic: Dynamic topic modeling (topics over time)," https://maartengr.github.io/BERTopic/getting_started/topicsovertime/topicsovertime.html, 2025, acessado em: 2025-12-02.

[22] Gensim, "models.coherencemodel — gensim documentation," https://radimrehurek.com/gensim/models/coherencemodel.html, 2024, acessado em: 2 dez. 2025.