



CURADORIA DE DADOS · MODELOS EM PORTUGUÊS TREINADOS DO ZERO

Curadoria **por token**, não por volume

Como escolher bem cada token de português rende mais do que ingerir mais texto: uma curadoria em duas camadas sobre a web em português e uma escada de três modelos treinados do zero, de 340 milhões a 2,9 bilhões de parâmetros. O maior, com 60 bilhões de tokens, fica na fronteira eficiente em 9 de 12 testes e chega perto, em conhecimento, de um modelo brasileiro que viu 515 bilhões. Com os resultados negativos e os limites do que foi demonstrado.

RESULTADOS MEDIDOS ATÉ OUTUBRO DE 2026

9 de 12

testes em que nenhum modelo com o mesmo orçamento de tokens, ou menor, supera a melhor versão do Yris 2,9 B (41 modelos públicos)

< 1/8

dos tokens do Tucano-2b4 para ficar 2 pontos atrás dele no ENEM em texto (60 bilhões contra 515 bilhões)

46 %

dos documentos de uma amostra da web pública em português recebem nota de qualidade baixa (≤ 2 de 5)

Curadoria por token, não por volume

Guilherme Biagi · Yris Lab — relatório técnico, versão pública 1.0 · outubro de 2026

Resumo

A Yris é uma inteligência artificial brasileira. O modelo que atende hoje no chat é um modelo aberto de terceiros; em paralelo, construímos uma trilha própria, com modelos de linguagem treinados do zero em português. Este relatório descreve o que essa trilha mediu até outubro de 2026.

A tese é de eficiência: para um orçamento de treino fixo, escolher bem os tokens rende mais do que ingerir mais tokens. Numa amostra de documentos da web pública em português rotulada por modelo de linguagem, cerca de 46 % recebem nota de qualidade baixa (≤ 2 numa escala de 0 a 5). Uma curadoria em duas camadas reduziu mais de cem bilhões de tokens a **20,8 bilhões**, concordando 95,2 % com a decisão que um classificador caro tomaria vendo todos os documentos, por cerca de um quinto do esforço de computação.

Sobre esse corpus e seus sucessores treinamos uma escada de três modelos do zero. O de **1 bilhão de parâmetros**, treinado com 20,7 bilhões de tokens, faz **62,0** no CALAME-PT e **41,1** no LAMBADA-PT, contra 59,1 e 30,3 do Tucano-1b1 (250 bilhões de tokens) no mesmo programa de avaliação. O de **2,9 bilhões**, treinado com 60 bilhões de tokens (um terço em português), fecha em **62,2 / 43,8**, acima dos pisos definidos antes do treino. Numa comparação com **41 modelos públicos em 12 testes**, a melhor versão do nosso modelo de 2,9 bilhões em cada teste está na **fronteira eficiente** em 9 deles: nenhum modelo treinado com a mesma quantidade de texto, ou menos, foi melhor — com a ressalva de que, nessa faixa de orçamento, há poucos modelos públicos para comparar.

Também registramos o que não funcionou. A distância que resta para modelos do mesmo tamanho treinados com trilhões de tokens é de conhecimento, não de formato, e três tentativas de comprá-la com doses pequenas de dado novo não moveram o resultado. Todos os critérios de sucesso foram escritos antes de cada treino, e os resultados são reportados como saíram, inclusive os inconclusivos.

Nota de leitura

Este é um relatório técnico de um produto fechado. Ele descreve o que medimos e como medimos. Não descreve a receita de treino: formato dos modelos, hiperparâmetros, composição exata dos dados e infraestrutura ficam de fora.

Todos os números são **medidos**. Os de outros modelos foram obtidos rodando esses modelos **no nosso próprio programa de avaliação**, salvo quando o texto diz "publicado"; cada modelo roda com a configuração de contexto adequada a ele. Em CALAME-PT e LAMBADA-PT usamos acerto exato sem diferenciar maiúsculas de minúsculas. A margem de erro (IC95) da diferença entre dois modelos é de cerca de **$\pm 3,0$ pontos** no CALAME-PT e **$\pm 1,9$** no LAMBADA-PT.

1. Por que treinar do zero

Os grandes modelos de linguagem são treinados principalmente em inglês, e o português entra como uma parte pequena do treino. Um modelo treinado do zero permite que o português seja a base, que cada fonte de dado seja escolhida e conferida, e que a tecnologia seja de quem a constrói.

O gargalo, na escala de um laboratório pequeno, não é arquitetura nem capital: é substrato. Coletar dez vezes mais português de qualidade não é uma opção. Separar o melhor do que já existe, é. Daí a frase que dá nome a este relatório:

Curadoria por token compra o que volume compra.

A aposta metodológica é a mesma de trabalhos brasileiros como o Tucano e o TeenyTinyLlama: modelo escrito do zero, avaliação própria e resultados honestos, inclusive os negativos. A diferença é que a Yris é um produto fechado; por isso este relatório publica resultados e método de avaliação, e não pesos nem receita.

2. Curadoria dos dados

2.1 Quanto da web em português é bom

Uma amostra de 1.600 documentos da web pública em português, estratificada por tamanho, foi rotulada de 0 a 5 por um modelo de linguagem. A distribuição dos rótulos foi: 0 — 6,3 % · 1 — 16,4 % · 2 — 23,3 % · 3 — 35,9 % · 4 — 17,9 % · 5 — 0,1 %. **Cerca de 46 % dos documentos da amostra tiram nota ≤ 2 .** É a medida, para o português, do problema que motiva toda a curadoria.

% dos 1.600 documentos da amostra

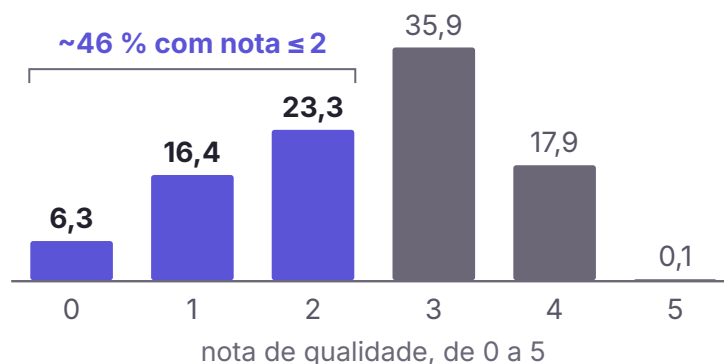


Figura 1 · Notas de qualidade, de 0 a 5, de 1.600 documentos da web em português. Em roxo, as notas 2 ou menos: cerca de 46 % da amostra.

2.2 Duas camadas

Depois de um filtro heurístico e de uma deduplicação de páginas repetidas, os documentos restantes, somando mais de cem bilhões de tokens, receberam uma nota de qualidade.

A decisão de manter ou descartar cada documento é tomada em cascata. Um classificador barato, que roda em CPU, decide os casos fáceis: documentos claramente acima ou abaixo do corte. Um classificador mais caro, que combina dois sinais complementares, reavalia apenas a faixa de documentos perto do corte, onde a decisão é incerta.

O achado que torna a cascata possível: o classificador barato **empata** com um classificador neural de embeddings em correlação com os rótulos (Spearman +0,519 contra +0,510), a centenas de vezes a vazão. O neural sozinho não se justifica. O que se justifica é a combinação dos dois (+0,595), aplicada só onde ela muda a decisão.

2.3 Quanto a cascata erra

A referência é a decisão que o classificador caro tomaria vendo 100 % dos documentos. Medida numa amostra de 20 mil documentos estratificada para concentrar todo o erro possível:

POLÍTICA	CONCORDÂNCIA COM O SCORING INTEGRAL
Só o classificador barato	88,3 % (IC95 88,1–88,4)
Cascata	95,2 % (IC95 95,0–95,4)

A cascata alcança essa concordância com cerca de **um quinto** do esforço de computação do scoring integral. Os erros residuais não se acumulam na borda do corte, e o classificador caro alterou de fato 38,7 % das decisões que reviu. O corpus final tem **20,8 bilhões de tokens**.

concordância com o scoring integral, %

Só o classificador barato



IC95: 88,1 a 88,4 %

Cascata



IC95: 95,0 a 95,4 %

A cascata gasta cerca de 1/5 do esforço de computação.

Figura 2 · Concordância com o scoring integral, numa amostra de 20 mil documentos. O traço na ponta de cada barra é o IC95. A cascata chega a 95,2 % com cerca de um quinto do esforço de computação.

2.4 Um defeito que a curadoria não viu

Uma lição registrada porque nos custou: na coleta de código, uma etapa de limpeza removeu a indentação de todo o Python. A curadoria pontua documentos, não sintaxe, e o defeito passou por todos os filtros de qualidade. Ele foi encontrado no meio do treino do modelo de 2,9

bilhões e corrigido sem reiniciar o treino (§3.4). Depois da correção, os testes de programação saíram do zero em poucos milhares de passos.

3. A escada de modelos

Subimos um degrau de cada vez. Cada degrau tem critérios de sucesso e de falsificação escritos e assinados antes do primeiro token de treino, e o critério não muda depois de visto o resultado.

3.1 340 milhões de parâmetros

O primeiro modelo, treinado com um corpus anterior, menor e menos curado. Ele mostrou duas coisas. Especialização funciona: um ajuste fino dirigido levou a tarefa de gerar SQL executável a partir de uma pergunta em português de 0 % para 64 % de acerto, com 52 % em esquemas de banco de dados nunca vistos no treino. Mas trocar a mistura do ajuste fino deixou de render: três ciclos seguidos ficaram dentro de 2 pontos. O gargalo era o texto de pré-treino. O passo seguinte foi corpus, não parâmetros.

3.2 1 bilhão de parâmetros

Treinado do zero sobre o corpus curado, por exatamente uma época (20,7 bilhões de tokens vistos), sem repetir nenhum token. O desenho de uma época permitiu observar se o corpus acabava dentro do orçamento, e acabou: o último quarto do treino não acrescentou nada dentro da margem de erro em CALAME-PT e LAMBADA-PT.

Depois de um ajuste fino multitarefa, o modelo chegou a 86,4 % na tarefa de SQL executável e respondeu em prosa a perguntas gerais, curando um defeito do modelo anterior, que respondia tudo em SQL.

3.3 Que mistura de dados usar no degrau seguinte

O corpus do degrau seguinte soma 60 bilhões de tokens: português curado, código, ciência e matemática, e inglês. Para escolher as proporções, treinamos seis versões de um modelo pequeno com o mesmo orçamento de tokens e misturas diferentes, incluindo duas repetições da mesma mistura com sementes diferentes.

O resultado foi **inconclusivo**, e o motivo é o próprio achado: **a distância entre duas corridas da mesma mistura foi maior do que a distância entre misturas diferentes**. O ruído entre corridas é real e domina nessa escala. Dois achados sobreviveram:

1. **Texto em português é o que melhora o português.** A versão treinada só com português fez 33,8 / 12,3 em CALAME-PT / LAMBADA-PT, contra cerca de 22 / 9 das misturas em que dois terços do texto não eram português. A diferença não fechou com mais tokens.
2. **Os testes de programação e SQL ficaram em zero em todas as versões.** Nessa escala, a régua não mede o que a mistura técnica poderia comprar. A pergunta só pôde ser respondida no degrau seguinte.

Também registramos um erro de método: na triagem inicial, um dos critérios de veto media quanto cada versão tinha treinado em certo tipo de texto, e não a sua qualidade. O critério foi detectado, declarado e retirado do protocolo seguinte antes de qualquer avaliação existir.

3.4 2,9 bilhões de parâmetros

Treinado do zero sobre 60 bilhões de tokens, por uma época, entre 30 de agosto e 9 de setembro de 2026. Um terço do texto é português curado — o mesmo português que existia —; o resto é código, ciência e matemática, e inglês. A hipótese do degrau tinha duas partes: (a) com um terço de português, o modelo não perde em português para o de 1 bilhão; (b) ganha uma competência técnica que os degraus anteriores não tinham.

A primeira tentativa foi abandonada. O treino divergia depois de milhares de passos estáveis. O problema foi diagnosticado, corrigido, e o treino foi refeito do zero; a segunda tentativa chegou ao fim da época sem repetir o colapso.

O protocolo de parada antecipada mediu o modelo em 5, 25, 50, 75 e 100 % do treino, comparando-o com o de 1 bilhão no mesmo volume de português visto. Num dos marcos o modelo ficou abaixo do limite por margem mínima; o treino seguiu com uma condição mais dura no marco seguinte, que passou.

Fechamento (100 % do treino):

TESTE	2,9 B	CRITÉRIO ESCRITO ANTES	1 B
CALAME-PT	62,2	≥ 62,0	62,0
LAMBADA-PT	43,8	≥ 41,1	41,1
HumanEval (164 problemas)	8 (4,9 %)	≥ 5 problemas	0
MBPP (500 problemas)	30 (6,0 %)	—	—
ARC-PT / HellaSwag-PT	30,0 / 39,1	acima do acaso (25)	28,5 / 37,5

Os pisos passaram; as metas mais altas, também escritas antes (65 e 44), não foram atingidas, e o CALAME-PT passa o piso por 0,2 ponto. As duas partes da hipótese se sustentam. (a) Com um terço de português, o CALAME-PT empata com o modelo de 1 bilhão e o LAMBADA-PT sobe 2,7 pontos, fora da margem de erro. (b) Os testes de programação saem do zero, mas ficam em 5–6 %: é substrato, não habilidade.

3.5 Memória maior

Depois do pré-treino, o contexto do modelo foi ampliado, com o critério de não perder mais de 1 ponto em CALAME-PT e LAMBADA-PT. O modelo estendido ficou em 61,7 / 42,9, dentro do critério.

4. Como avaliamos

4.1 O mesmo programa para todos

Comparar o nosso número com o número publicado por outro grupo mistura protocolos diferentes (exemplos no prompt, normalização, formato da resposta). Por isso rodamos os modelos públicos **no nosso próprio programa de avaliação**, com o mesmo código de pontuação e geração, as mesmas questões e as mesmas regras. Quando um número publicado aparece neste relatório, ele vem marcado como tal e serve para situar, não para decidir.

4.2 Que testes contam

Seguindo a análise do Tucano, a régua principal nesta faixa de tamanho é a de previsão de texto: CALAME-PT e LAMBADA-PT, em que o modelo precisa acertar a palavra que completa uma frase. São os testes que de fato acompanham a escala. Provas como ENEM, BLUEX e OAB entram como diagnóstico de conhecimento.

4.3 Texto e letra: uma régua que precisou de calibração

Uma prova de múltipla escolha pode ser aplicada de dois jeitos. Em **texto**, pontuamos cada alternativa pela probabilidade que o modelo dá ao seu conteúdo. Em **letra**, mostramos a questão com as alternativas rotuladas e pontuamos só a letra escolhida. A diferença é grande:

ENEM (1.431 questões), sem exemplos no prompt

MODELO	TEXTO	LETRA
Gemma-3-4B-it (4 trilhões de tokens)	35,6	62,8
Sabiá-7B	36,6	53,5
Tucano-2b4 (515 bilhões de tokens, do zero)	32,4	19,9
Manacá-1B (40 bilhões de tokens, do zero)	32,2	19,0
Yris 2,9 B (60 bilhões de tokens, do zero)	30,3	19,1

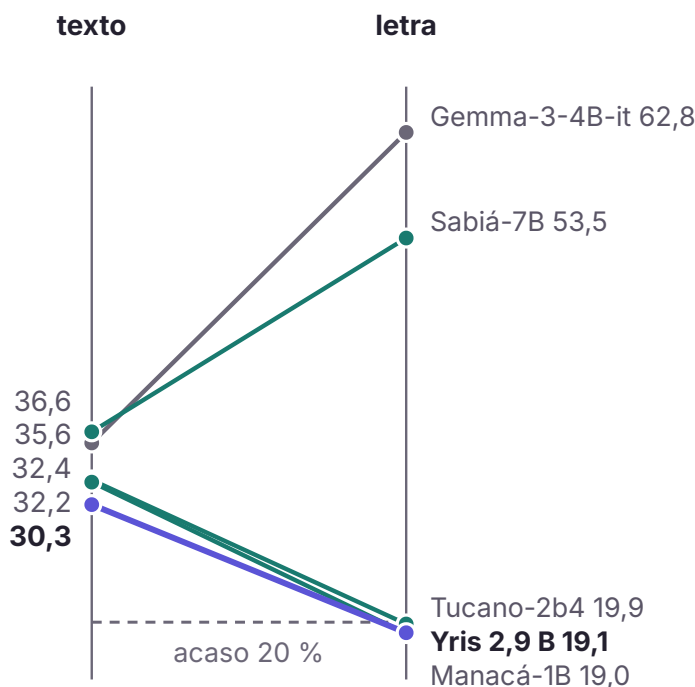


Figura 3 · ENEM em texto e em letra, sem exemplos no prompt. Cada linha liga o acerto de um modelo nos dois formatos. Gemma e Sabiá sobem em letra; os três modelos pequenos treinados do zero caem para o acaso (linha tracejada, 20 %).

O formato **texto comprime**: modelos muito diferentes caem na faixa de 30 a 37. O formato **letra mede conhecimento e também a habilidade de escolher uma alternativa rotulada**, e os três modelos pequenos treinados só com web em português ficam no acaso, com forte

preferência por uma mesma letra. O limite não é o tamanho: o Gemma-3-4B, de porte parecido, faz 62,8. Consequência: para comparar conhecimento entre modelos que não foram ajustados para provas, a régua certa é o texto.

5. Resultados

5.1 Previsão de texto, contra modelos públicos

Todos medidos no nosso programa de avaliação.

MODELO	PARÂMETROS	TOKENS DE TREINO	CALAME-PT	LAMBADA-PT
Tucano-630m	0,6 B	211 B	56,8	30,0
Tucano-1b1	1,1 B	250 B	59,1	30,3
Yris 1 B	1,0 B	20,7 B	62,0	41,1
Tucano-2b4	2,4 B	515 B	60,6	36,1
Manacá-1B	1,7 B	40 B	61,0	46,9
Yris 2,9 B	2,9 B	60 B	62,2	43,8
Curió-1.1b	1,1 B	1.150 B	65,0	52,5

Como ler, aplicando a margem de erro:

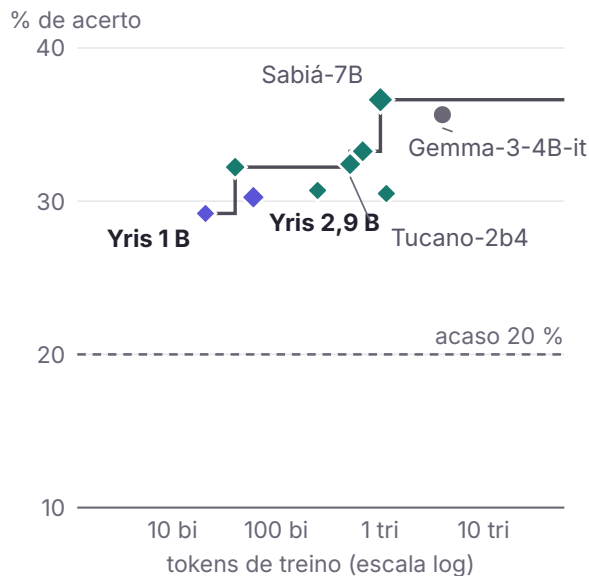
- **LAMBADA-PT é o resultado sólido.** O Yris 1 B supera o Tucano-1b1 por 10,8 pontos tendo visto cerca de 12 vezes menos texto; o Yris 2,9 B supera o Tucano-2b4 por 7,7 pontos com mais de 8 vezes menos.
- **No CALAME-PT, a vantagem é pequena.** Contra os Tucanos de tamanho parecido (1b1 e 2b4), as diferenças ficam dentro ou no limite da margem de erro.
- **O Curió continua à frente** nos dois testes, com mais de 50 vezes o texto do nosso 1 B. E o Manacá-1B, com 40 bilhões de tokens, nos supera em LAMBADA-PT. Curadoria estica o orçamento; não substitui volume.

5.2 A fronteira eficiente

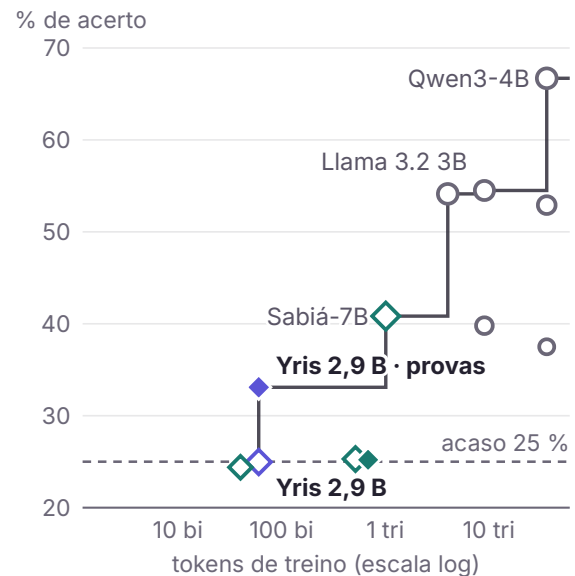
A pergunta deste trabalho é de eficiência: quanto um token curado compra. A forma mais direta de mostrá-la é pôr no eixo horizontal o total de tokens de pré-treino de cada modelo, em

escala logarítmica, e no eixo vertical a nota. A **fronteira** liga, para cada orçamento, a maior nota entre os modelos que viram aquela quantidade de texto ou menos.

ENEM (texto)



MMLU-PT (letra)



CALAME-PT



LAMBADA-PT

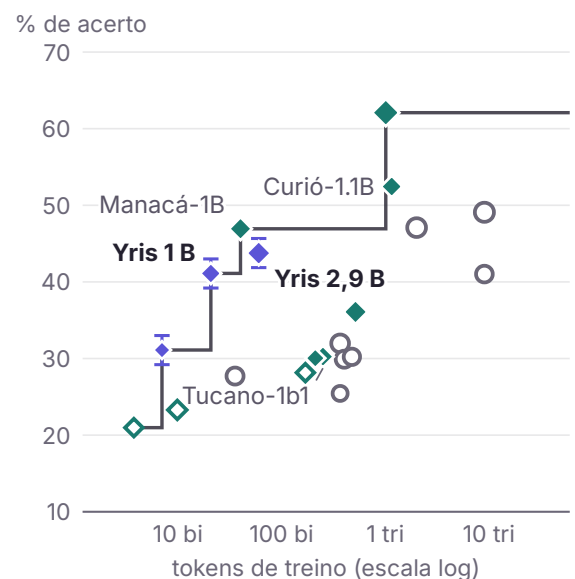


Figura 4 · Quatro dos doze testes. Losango: modelo brasileiro (roxo, a Yris; verde, os demais); círculo cinza: modelo estrangeiro; marcador vazado: número publicado por terceiros ou medido em condição diferente. Tamanho proporcional aos parâmetros. Linha escura: fronteira eficiente. No CALAME-PT e no LAMBADA-PT, as barras roxas são a margem de erro. Mostramos os modelos brasileiros de qualquer tamanho e os estrangeiros de até 4,5 bilhões de parâmetros.

Comparamos o modelo de 2,9 bilhões com **41 modelos públicos em 12 testes**: MMLU-PT, ENEM, BLUEX, OAB, Belebele, CALAME-PT, LAMBADA-PT, ARC-PT, HellaSwag-PT, SQL executável, HumanEval e MBPP. Usando, em cada teste, a melhor versão do nosso modelo, ele está na fronteira em **9 dos 12**. A leitura tem limite: até 60 bilhões de tokens há poucos modelos

públicos (cinco), em vários testes o rival nessa faixa é um número publicado, e no SQL executável não há rival externo. As três exceções:

- **LAMBADA-PT:** o Manacá-1B, treinado do zero com 40 bilhões de tokens, fica à frente (46,9 contra 43,8).
- **HumanEval e MBPP:** o phi-1, treinado com texto didático sintético de programação, mostra o que dado de código de qualidade compra com um orçamento parecido com o nosso.

9 de 12 testes na fronteira

MMLU-PT	✓ na fronteira
ENEM	✓ na fronteira
BLUEX	✓ na fronteira
OAB	✓ na fronteira
Belebele	✓ na fronteira
CALAME-PT	✓ na fronteira
LAMBADA-PT	✗ à frente: Manacá-1B
ARC-PT	✓ na fronteira
HellaSwag-PT	✓ na fronteira
SQL executável	✓ na fronteira
HumanEval	✗ à frente: phi-1
MBPP	✗ à frente: phi-1

Até 60 bi tokens há só 5 modelos públicos;
no SQL executável não há rival externo.

Figura 5 · Em quais dos doze testes o melhor ponto da Yris está na fronteira eficiente. Nas três exceções, o rótulo diz quem está à frente com o mesmo texto ou menos.

Ressalvas: o eixo horizontal soma o orçamento de pré-treino em qualquer língua, incluindo o do modelo de partida no caso de modelos adaptados; e números publicados usam protocolos próprios, por isso aparecem vazados.

5.3 Conhecimento e provas

Em texto, a régua que todos sabem responder, o Yris 2,9 B fica no mesmo patamar do Tucano-2b4, que viu mais de 8 vezes mais texto: no ENEM, 30,3 contra 32,4, 2 pontos abaixo.

Em letra, o modelo base fica no acaso, como todo modelo pequeno treinado só com web em português. Um ajuste fino com dezenas de milhares de questões de múltipla escolha, verificadas contra vazamento das provas de avaliação, ensinou a operação de escolher a alternativa:

Provas em formato letra, sem exemplos no prompt, nosso programa

PROVA (QUESTÕES)	YRIS 2,9 B AJUSTADO PARA PROVAS	YRIS 2,9 B AJUSTADO PARA CONVERSA	TUCANO-2B4- INSTRUCT	ACASO
ENEM (1.431)	38,6	19,6	20,2	20
BLUEX (722)	35,3	18,8	19,9	20
OAB (2.210)	36,4	23,0	23,9	25
MMLU-PT (14.042)	33,1	23,1	25,2	25
Belebele (900)	45,8	22,6	—	25

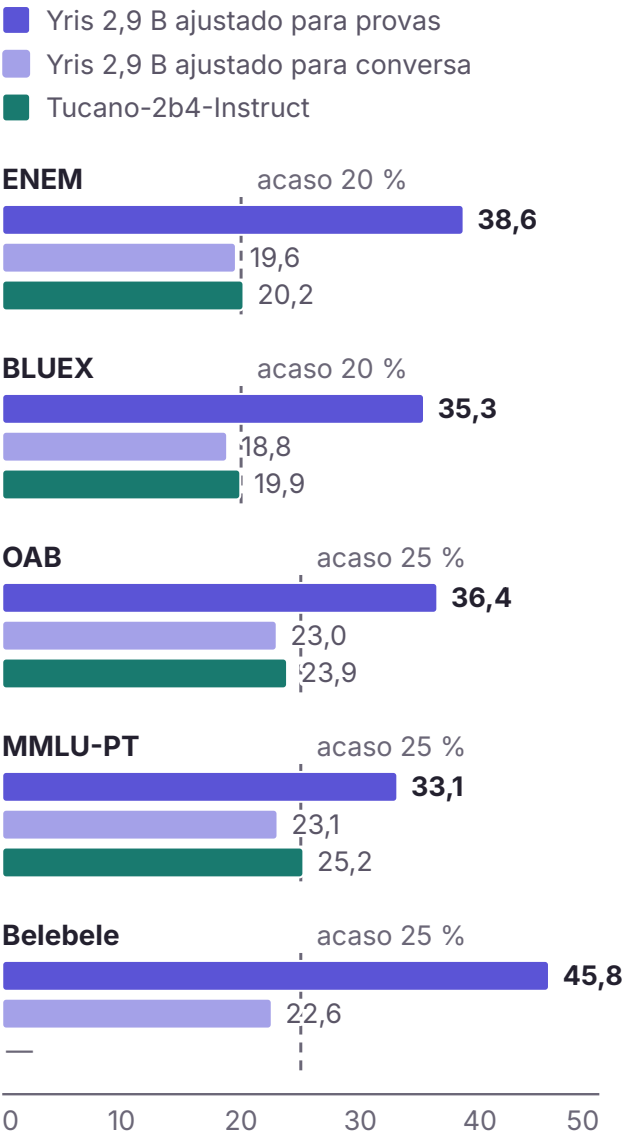


Figura 6 · Provas em formato letra, sem exemplos no prompt. A linha tracejada de cada prova é o acaso; o traço (—) marca a prova em que o modelo não foi medido. Só o modelo ajustado para provas se afasta do acaso.

Na OAB, com o protocolo público de avaliação em português (três exemplos no prompt), o modelo ajustado faz 35,6, contra 25,4 do nosso modelo base e 26,7 do Tucano-2b4 base, ambos

no acaso. Modelos de tamanho parecido treinados com trilhões de tokens ficam acima: 39,5 o Llama 3.2 3B e 50,3 o Qwen2.5-3B (números publicados).

A leitura honesta: o ajuste moveu a **habilidade de responder prova**, não o conhecimento. Nos testes que não dependem do formato, o modelo antes e depois do ajuste empata. Em texto, o Tucano-2b4-Instruct fica 1 a 2 pontos acima (ENEM 33,3 contra 31,0; OAB 32,8 contra 31,0).

Conclusão: **conhecimento no mesmo patamar, 1 a 2 pontos abaixo, com menos de um oitavo dos tokens**. E o ajuste tem custo: -4,1 pontos em LAMBADA-PT em relação à versão anterior e perda da habilidade de SQL, que não estava na mistura do ajuste.

5.4 Onde estamos atrás

- **Conhecimento geral.** No MMLU-PT, o modelo ajustado acerta 33 %. Modelos de 3 a 4 bilhões de parâmetros treinados com 4 a 36 trilhões de tokens acertam de 54 % a 67 % (números publicados, com protocolos diferentes do nosso). Essa distância é de conhecimento, não de formato.
- **Programação.** HumanEval e MBPP ficam em 5–6 % no modelo base.
- **Conversa.** Todo ajuste para conversa custou de 3 a 15 pontos de LAMBADA-PT, conforme a receita. Misturar texto do pré-treino no ajuste segurou parte da queda, mas nenhuma receita testada ensinou a conversar sem esquecer português.



Figura 7 · MMLU-PT: o modelo ajustado para provas contra a faixa dos modelos de porte parecido treinados com trilhões de tokens (números publicados, com protocolos diferentes do nosso).

6. O que não funcionou

Publicamos os resultados negativos porque eles orientam o próximo passo tanto quanto os positivos.

Estender o contexto do modelo de 1 bilhão, sem retreino longo. A extensão custou 3,2 pontos de LAMBADA-PT enquanto a perda de validação global melhorava: uma das três vezes, neste projeto, em que a perda de validação andou na direção contrária da capacidade que interessava. O resultado levou o degrau seguinte a ter contexto maior desde o início.

Doses pequenas de dado novo num modelo pronto. Testamos três pré-treinos continuados sobre o modelo de 2,9 bilhões, cada um com regra escrita antes:

DADO NOVO	O QUE MEDIU	RESULTADO
Texto acadêmico em português	ENEM e MMLU-PT	absorvido, mas sem mover o conhecimento (MMLU-PT 27,4 → 27,5)
Texto didático gerado por outro modelo	MMLU-PT e ENEM contra um controle	sem ganho sobre o controle; custou 2–3 pontos de português
Código de programação	HumanEval e MBPP	sem ganho; interrompido pela regra de futilidade

As doses foram de cerca de 1 % do pré-treino. A leitura: **doses pequenas não mudam um modelo já treinado**. Isso não é um teto do tamanho do modelo: o Gemma-3-4B, de porte parecido, está muito acima com trilhões de tokens. A distância é de volume de dado, e português de qualidade nesse volume não existe.

Instrumentos que davam números plausíveis e errados. Uma régua de prova que media preferência por uma letra e não conhecimento (§4.3); uma curva de acompanhamento com rótulos deslocados por um problema de sincronização; juízes automáticos que não passaram na própria calibração e, por isso, não decidem nada neste relatório. A defesa foi sempre a mesma: escrever o critério antes, medir o instrumento antes de medir o modelo e publicar o resultado como saiu.

7. Limitações

1. **Falta a comparação direta da tese.** O experimento que testaria a curadoria sozinha (mesmo modelo, mesmo orçamento, corpus curado contra não curado) ainda não foi feito.
2. **Rótulos de qualidade de um único modelo de linguagem**, sem medida de concordância entre rotuladores.
3. **A fronteira é uma foto, não uma lei de escala.** Ela compara modelos com arquiteturas, vocabulários e protocolos diferentes; os pontos publicados servem para situar.
4. **Conhecimento do gerador.** A verificação contra vazamento protege contra a repetição de questões, não contra conhecimento transmitido por dados gerados. Por isso o número limpo para a tese de eficiência é o de texto.
5. **O ajuste fino apaga o que não está na mistura** e custa língua em todas as receitas testadas.
6. **Juízes automáticos não decidem nenhuma afirmação deste relatório**, porque não sobreviveram à própria calibração.

8. Próximos passos

Com o português de qualidade que existe, a aposta do próximo degrau é **um modelo maior, que tire mais de cada token**.

Enquanto isso, a Yris segue no ar com um modelo aberto, e esta página será atualizada a cada degrau.

Referências

1. Corrêa, N. K. et al. *Tucano: Advancing Neural Text Generation for Portuguese*. arXiv:2411.07854, 2024.
2. Corrêa, N. K. et al. *TeenyTinyLlama: open-source tiny language models trained in Brazilian Portuguese*. arXiv:2401.16640, 2024.
3. Corrêa, N. K. et al. *Tucano 2 Cool: Better Open Source LLMs for Portuguese*. arXiv:2603.03543, 2026.
4. ClassiCC-Corpus. *Curió-1.1b* (modelo). HuggingFace.
5. LNCC. *Manacá-1B* (modelo). HuggingFace.
6. Pires, R. et al. *Sabiá: Portuguese Large Language Models*. arXiv:2304.07880, 2023.
7. Rae, J. W. et al. *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*. arXiv:2112.11446, 2021.
8. Hoffmann, J. et al. *Training Compute-Optimal Large Language Models*. arXiv:2203.15556, 2022.
9. Gunasekar, S. et al. *Textbooks Are All You Need*. arXiv:2306.11644, 2023.
10. Qwen Team. *Qwen2.5 Technical Report*. arXiv:2412.15115, 2024; *Qwen3 Technical Report*. arXiv:2505.09388, 2025.
11. Gemma Team. *Gemma 3 Technical Report*. arXiv:2503.19786, 2025.
12. Meta. *Llama 3.2* (model cards).
13. Bandarkar, L. et al. *The Belebele Benchmark*. arXiv:2308.16884, 2023.
14. Lai, V. D. et al. *Okapi* — protocolo multilíngue de ARC e HellaSwag.

Yris Lab · yrisia.com.br/pesquisa · versão pública 1.0, outubro de 2026.