**Como transformar vídeos de Youtube em texto com langchain - Projeto real LLM, chatgpt, langchain**

**Transcrito por** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell)**.** [**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **para remover esta mensagem.**

Quer aprender como transformar vídeos do YouTube em texto utilizando o LangChain e o ChatGPT? Fica aqui comigo nesse vídeo. Oi, pessoal, aqui é o Rafa. Esse é o segundo vídeo da série de LangChain onde eu estou construindo um projeto real, do começo ao fim.

Vou relatar aqui para vocês como se fosse quase um vlog de todas as dificuldades que eu vou sentir no projeto, todas as ideias que eu vou ter no projeto. E essa é a primeira parte realmente mão na massa. O último vídeo que saiu foi um vídeo um pouco mais de planejamento, teórico.

Eu expliquei para vocês os projetos, expliquei o que eu queria fazer, quais são as interfaces gráficas até que eu queria chegar. Como a maioria de vocês falou nos comentários que não se interessa tanto por essa parte de front-end, eu vou deixar isso aí bem de lado. Talvez no final eu faça alguma coisa no Streamlit ou em alguma outra ferramenta bem simplesinha, só para a gente ter uma aplicação de começo ao fim.

E eu vou focar na parte de back, principalmente na parte de A, LangChain, que acredito que é o que vocês estão buscando aqui nesse vídeo. E nesse vídeo de hoje eu vou resolver o primeiro feature do nosso aplicativo, que é simplesmente eu colocar um link de vídeo qualquer do YouTube e a aplicação me dê qual é o nicho desse vídeo. Por quê? Porque a partir disso eu vou interagir com a API do YouTube e vou conseguir pesquisar vídeos relevantes para fazer análise da nossa aplicação.

Lembrando, se você está caindo aqui nesse vídeo, está aqui no card, aqui em cima o vídeo da primeira série. A minha ideia é construir um app para ajudar criadores de conteúdo a ver quais são as tendências gerar ideias de conteúdo. Então a ideia é que o criador de conteúdo pegue o canal que ele se interessa, pode ser o canal dele mesmo ou ele pode ver algum concorrente, coloca essa URL na nossa aplicação, dá Enter e aparece toda uma interface com estatísticas de outros canais e uma lista de sugestões de roteiro para ele gravar, com base nos principais vídeos que ele viu na semana, tendências, coisas que estão rolando no nicho dele.

Então a nossa primeira tarefa, a nossa primeira missão é justamente, a partir de um só link do YouTube, a gente chegar nos nichos. Claro que isso ainda é por debaixo dos panos, caixa preta, o cliente final o criador de conteúdo não vai ver isso. Mas para a gente estar desenvolvendo é importante ter essas etapas.

Toda tarefa complexa a gente tem que quebrar ela em partes simples que a gente consegue ter começo, meio e fim e resolver. E essa é uma parte bem simples. Primeiro, antes de a gente buscar coisas no YouTube relevantes a gente tem que definir o nicho.

E a partir do nicho a gente vai selecionar palavras e termos interessantes para fazer essa busca. Então o vídeo de hoje é coloque o link, a gente recebe o nicho no final. E agora vamos botar a mão na massa no código e começar a trabalhar nessa aplicação.

Então pessoal, a gente está aqui no Jupyter Notebook para começar a botar a mão na massa. Mas antes de começar a trabalhar de fato na parte de lang chain e A, a gente vai ter que entender um pouco como fazer a interação com a API do YouTube. Não tem segredo para ninguém.

Digita no Google YouTube Cloud você vai ver uma tela parecida como essa aqui vai em console ela vai aparecer se vai ter que criar um projeto, como eu tenho um projeto aqui criado, só clico aqui em APIs ele abaixa aqui e aí você carrega a página carregou a página eu tenho aqui algumas estatísticas do serviço que eu tenho usado venho aqui em credenciais e clico aqui para criar uma chave caso você não tenha. Como eu já tenho uma chave aqui, não vai me deixar criar uma outra chave para esse projeto. Mas em 5 minutos da documentação, vocês vão entender como fazer o cadastro, gerar a sua chave e aí vocês podem definir ela como variável de ambiente da mesma maneira que a gente já vem fazendo com a chave da OpenAI.

A gente vai utilizar ela aqui. Essa chave aqui eu já defini como a variável de ambiente, já carreguei ela aqui no notebook antes de começar o vídeo. Então a única coisa que você tem que ter certeza pra isso aqui funcionar, se você quiser replicar, é pegar esse YouTube API Key e setar ele como variável de ambiente, assim como a gente vai ter que fazer com a OpenAI que a gente já tem feito.

Acredito que não seja um processo muito difícil mas se tiver alguma dúvida, algum problema é só deixar nos comentários que a gente dá o risco de ajudar. Uma coisa que eu mudei um pouco no estilo dos vídeos, eu acho que foi bem fadonho os vídeos que eu digito o código aqui na hora. Eu acredito que eu perco muito tempo digitando os vídeos então a partir de hoje eu vou fazer um modelo um pouco diferente, eu vou digitar os códigos antes, fora de gravação e vou fazer passo a passo com vocês explicando linha por linha de código, qual que é a minha ideia, quais são as dificuldades que eu tive, porque eu tomei tal decisão ou outro caminho.

E a primeira coisa que a gente tem que entender, porque a gente está aqui no Jupyter Notebook e não no VS Code. Sempre que você vai começar um projeto novo principalmente um projeto que você não tem certeza de como fazer a gente tem que realizar uma POC que está aqui. POC é o que? Proof of Concept.

É você provar o conceito. E o que a gente quer provar aqui com esse código? Que é possível a partir da URL do vídeo qualquer a gente ter o nicho desse vídeo. Então no final desse vídeo, se eu conseguir ter esse resultado, pegar um link uma URL, colocar ela como input e sair os nichos do canal como output, minha prova de conceito está validada, eu posso começar a pensar em como colocar isso aqui dentro de um código profissional.

Como eu quero classes, outros níveis de abstração e por aí vai. Mas primeiro mantenha o foco no que você quer provar, no que você quer comprovar com esse vídeo, certo? E para começar, comecei a estudar a API do YouTube e é uma API muito bem documentada. Aqui eu já pesquisei, por exemplo, Search List, que é uma coisa que eu usei.

Primeiro link milhares de coisas sobre como usar a Search List. Mais resultados. Aqui, Search List no YouTube.

Aqui, chegou a documentação. Uma documentação muito bem cuidadosa, super clara, tradução em português. Fala aqui, por exemplo, os parâmetros que você tem que usar, esse parte é um parâmetro obrigatório.

Escreve todos os parâmetros opcionais que você tem de filtros, o que você pode colocar, os tipos de vídeo que ele retorna, se é um short, se é um vídeo longo, localização, data. É uma API muito bem documentada, muito bem documentada mesmo. Até então, eu nunca tinha trabalhado com essas APIs do YouTube.

Achei fantástica a API, super tranquila de você ler a documentação e entender. Então, o meu primeiro desafio foi eu pegar um link qualquer do YouTube. Vou pegar aqui o primeiro link do meu canal, por exemplo.

A minha primeira intenção foi talvez se eu pegar direto o link do canal, fosse mais fácil. Como assim o link do canal? Aqui a gente tem o vídeo, propriamente dito, que é esse link aqui em cima, mas a gente também tem o vídeo, deixa eu baixar o som aqui para não atrapalhar, o vídeo aqui do canal mesmo, o Carreira em Inteligência Artificial. Eu pensei em pegar essa URL aqui e extrair já esse padrão, esse arroba aqui, mas eu descobri que isso não é padrão.

A minha primeira tentativa foi tentar extrair esse cara aqui e eu falei isso aqui não é padrão. E depois eu pensei, mais uma coisa, as pessoas geralmente não buscam o projeto logo de cara, o nome do canal. É muito mais natural pensando na user experience um usuário pegar um vídeo qualquer.

Então a minha ideia é pegar qualquer vídeo, joga aqui e a partir do vídeo eu entendo qual é o canal e eu extraio qual é o seu vídeo desse canal. Que é isso que a gente vai fazer aqui. E pensando um pouco nisso, fui estudar API e eu descobri que eu preciso achar o ID do canal.

Existe um ID único do canal e eu preciso extrair ele a partir do ID do vídeo. E para extrair a partir do vídeo eu preciso do ID do vídeo. Então temos três etapas.

Primeira etapa, eu preciso extrair o ID do vídeo. Então a gente tem aqui essa função que faz exatamente isso. Essas são funções nativas URL lib, pra gente parcializar URLs.

Isso aqui tem um padrão. Depois do V, ele tem o ID do vídeo. Como é que eu descobri isso? Mágica? Não, lendo a documentação.

Sempre leia a documentação. Essencial para você ter sucesso na carreira é ler a documentação. Vai virar rato de documentação.

Leia a documentação com carinho. Explore ela, exporte as possibilidades, estressa o ambiente máximo que você puder. E aqui a partir dessa leitura eu entendi que esse padrão que eu tenho que extrair então essa função não faz nada mais do que isso.

Lembrando que esse código não é um código colhido. A ideia da prova de conceito tem que ser rápida. Eu não estou fazendo nada pra escalar, eu não estou fazendo nada pra produção.

Eu quero provar o meu conceito. URL chega em nichos. Então, à medida que você vai crescendo na carreira, você vai pensando que o esforço, ele tem que ser otimizado.

Certo? Se tudo que você for fazer você for polir o código, deixar maravilhoso escalável, pronto pra produção você não vai produzir nada. Vai demorar muito. E na indústria as pessoas te cobram produtividade, entrega.

Então quando você tem que ser cuidadoso você tem que ser cuidadoso e demora mais. Quando você não precisar simplesmente faça o código mais rápido que você puder. Uma coisa que eu sempre tenho pra mim é tentar encapsular tudo em funções.

Isso ajuda depois quando você for transformar essa maçaroca do logbook em um código de verdade pra produção. Certo? Seguindo então, essa aqui é a parte, essa função faz exatamente isso, me deu o ID do código. Com o ID do vídeo aqui eu consigo, a partir do getChannelID byVideoID.

Então, de novo, fui lá na documentação entendi que a partir dessa videos.list eu posso passar um ID do vídeo e receber um ID do canal. Certo? Então já tenho o ID agora do canal, essa segunda função, crio como se fosse equivalente a um client. Imagina se a gente for entrar por baixo no código, é o que ele está chamando, deve ter algum request, é por aí que ele está chamando.

Assim que eu criei, quando eu criei esse build de onde ele veio, veio de novo, deu uma biblioteca aqui do Google, que recomenda-se que você instale essa biblioteca, depois, se vocês quiserem, o requirements vai estar disponível, é só vocês entrarem lá no grupo do Telegram que a gente vai conseguir disponibilizar isso pra vocês. Mas, basicamente, eu acredito que isso é um client, não fui muito a fundo pra você ver o que é isso aqui não. Mas eu sei que esses parâmetros aqui são exatamente os parâmetros que eu achei aqui na documentação.

O channelID está aqui, o part, que é o obrigatório, está aqui. Então, lendo a documentação, eu descobri como utilizar essa parte da API. E eu descobri que a gente consegue extrair, a partir disso, os links dos últimos vídeos vistos, é o que essa função faz.

Então, eu passo aqui o ID do meu canal, aí eu seto um parâmetro do número máximo de vídeos recentes, aí eu estou ordenando ele por data, ou seja, eu vou ordenar ele pelo vídeo mais recente. Então, se eu passar o channelID aqui de 7, sei lá, esse é o meu canal, e 40 aqui, ele vai pegar os 40 últimos vídeos do canal 7, que ele descobriu o ID a partir de um único vídeo. Aqui, eu acho que não tem muito o que explorar com vocês.

Items aqui, ele vai trazer a maioria dos vídeos, eu seto quantos vídeos eu quero aqui, eu transformo tudo numa lista e ele me devolve esse dicionário, que é o ID do vídeo, o título do vídeo e o URL do vídeo. Vamos rodar isso aqui uma vez. Vamos pegar aqui primeiro o do meu canal, e aqui eu vou fazer o seguinte, eu vou printar todos esses resultados intermediários, isso aqui ajuda a gente a compreender o que tem em cada uma dessas coisas, tanto no ID do vídeo, quanto no ID do canal, quanto em todos os vídeos.

Esse é o ID do vídeo que eu peguei do meu canal, esse vídeo que estava aberto ali. Esse aqui é o ID do meu canal, e essa lista aqui tem 10 vídeos. Vamos dar uma olhada na cara dessa lista.

Uma lista de dicionários, novo ID, ali é o nosso último vídeo que saiu, é o vídeo do Gemini, depois esse aqui do Python Deploy, temos o vídeo aqui LangChain Corregg, como estudar inteligência artificial, as séries que apareceram aqui. Então, são os últimos 10 vídeos que apareceram no meu canal. A gente pode trocar isso aqui, não precisa necessariamente ser o meu canal, esse aqui é o canal de culinária que eu peguei, por exemplo.

Bom, daqui de agora em diante, eu só faço macarrão e carne moída assim, misturei café e ovo, surpreendi todo mundo com essa combinação, fiz um quilo de queijo, um quilo e litro de leite, todos me pediram a receita. Canais variados, que a gente vai testar na nossa aplicação. Esse aqui é um canal de ciência.

Ser inteligente é péssimo, o sabor da banana não tem gosto de banana, elementos químicos que não existem, canal de ciência. E eu peguei um canal de hobby aqui, aquele jogo de cartas, o Magic The Gathering. Então tem sobre jogos, sobre o que as pessoas jogam, níveis, por aí vai.

Então a gente tem uma gama variada de canais e rapidamente eu consigo acessar os últimos vídeos que as pessoas postaram. A partir dessa interação aqui, repara, não é um código profissional, mas também não é uma saroca que ninguém entende nada. Então sempre tem esse cuidado, esse trade-off de uma coisa pra outra.

Bom, seguindo aqui, a gente tem agora a parte de fato, a gente vai gerar um curve pro nosso trabalho. E como é que eu vou fazer isso? Eu descobri também ali na API, que é possível a gente extrair as transcrições do YouTube. Então sabe aquela legenda que automaticamente ele coloca nos seus vídeos? Eu consigo extrair ela.

Então olha que maravilha, eu não preciso trabalhar direto com o áudio. Eu não preciso converter áudio em texto. Eu posso diretamente pegar a transcrição, a tradução.

Não é uma tradução perfeita, a gente sabe, principalmente pra linguagem não inglesa. Em inglês é bem legal a tradução. Português não é perfeito, mas como a gente vai usar LLMs pra processar tudo isso, não importa.

É tranquilo a gente utilizar os LLMs pra isso. Porque vai ter um corpo grande de informação pra trabalhar. E é um jeito barato pra gente fazer isso.

E como é que eu descobri isso? Agora não foi lendo a documentação do Google, foi lendo a LangChain, vendo o que o pessoal tá fazendo na comunidade. E eu descobri, na comunidade de LangChain, esse YouTuber Loader aqui. Vamos dar uma olhada nele.

Dentro da biblioteca LangChain você tem uma ferramenta que a comunidade já criou pra carregar o YouTube. Pra fazer esse serviço pra gente. Muito interessante o serviço open source.

Esse aqui você não precisa estar carregado em nada do YouTube. De graça esse serviço. Muito interessante.

Então eu tô utilizando ele justamente pra isso, nessa parte do código aqui. Eu passo uma lista de URLs pra ele. Aqui eu passo a língua que eu quero a transcrição.

Passo a URL aqui. Ele tenta carregar. Carregando, ele vai pegar o conteúdo de texto aqui pra mim.

E com esse conteúdo de texto, ele vai colocar nesse corpus. Vamos fazer o seguinte pra ficar mais claro. Eu vou pegar esse Loader aqui e vou rodar ele fora.

Vamos fazer um pequeno teste. E a minha URL... Vou pegar esse primeiro vídeo aqui. Standard Arena.

Já tá carregado. Vamos colocar o primeiro. O item zero.

E eu quero a URL desse dicionário. Rodei. E olha só que interessante.

Bem legal, né gente? Tá aqui toda a transcrição do vídeo. Então, bem acredito que estamos ao vivo em um dia de muita vontade. Boa noite pra quem já tá assistindo.

Se vocês abrirem o link do Criador de Conteúdo, aqui vai aparecer toda a transcrição. E tem muita coisa pra gente trabalhar. Muito legal mesmo essa parte da API.

Então, eu vou fazendo o quê? Vou fazendo esse loop. Então, rodo uma vez, pego o texto, coloco nessa string. Dou um espaço, rodo isso até ter um corpo gigante.

Então, eu tô criando uma super string com tudo que foi transcrito. Vamos colocar ela aqui. E agora eu vou passar minha lista de URLs.

Esta lista aqui, cada um desses dicionários pra esta função. Extraio a URL e passo a URL pra cá. Pra gerar esse corpus.

A título de curiosidade, esse corpus é bem grandinho. Ele tem 10 vídeos. Vamos ver o tamanho do bicho se terminar de rodar.

Tem 10 vídeos mesmo. Ele tem basicamente 587 mil palavras. Muito mais tokens do que isso.

A gente não vai conseguir processar tudo isso de uma vez. Por isso, pensando nisso, eu criei essa função super simples. Tem maneiras mais profissionais de fazer isso.

Por exemplo, criar um generator pra você quebrar isso, mas não estou interessado em performance, não estou interessado em um código pronto pra produção. Estou interessado em resolver um problema simples. Então, ele vai quebrar basicamente esse generator em pedaços de 3 mil e eu tenho 196 pedaços aqui basicamente.

E agora entramos na parte interessante, na parte de lang chain. A parte que é o que motivou e trouxe vocês até aqui. Primeiro, eu vou declarar uma lista de summaries que eu vou iterar.

Eu vou querer resumos sobre esses transcripts. A minha ideia é que eu vou passar cada um desses 196 transcripts primeiro pra esse prompt, baseado no conteúdo dessa transcrição. Aliás, uma como ficou.

Ela tem 3 mil caracteres. Por exemplo, essa aqui. Então, baseado nesse pedação de texto aqui.

Aí tem um, dois, três. Defina em apenas uma frase o nicho desse canal. Então, eu vou ter 196 frases tentando definir o nicho desse canal.

Depois disso, o que eu vou fazer? Além de definir o nicho do canal, eu vou falar pra cada uma dessas frases, agora resuma em no máximo 3 palavras o nicho deste output aqui. Então, primeiro eu vou passar isso aqui e aí ele vai me resumir o nicho do canal em um parágrafo. Não delimitei o quanto ele vai resumir.

A partir do output dele, que tá aqui. O input é o transcription, certo? Transcrição. E o output dele vai ser isso aqui.

YouTube summary. O output do primeiro, LLM, que é esse daqui que tá destacado, é muito importante que ele seja igual ao input do segundo. Porque ele vai automaticamente pegar o que ele gerar aqui e colocar aqui.

Aí ele vai fazer o resumo e no máximo 3 palavras o nicho do seguinte canal. Que é esse canal aqui em cima. Aqui a gente tá criando a nossa chain, que é o LLM, junto com o prompt que a gente criou.

Lembra dos outros vídeos que estão aqui, se tem alguma dúvida de como usar o prompt template. Um detalhe, vocês viram que eu instanciei o LLM aqui de maneira genérica. E isso é uma boa estratégia por quê? Se eu quiser amanhã testar, por exemplo, o Gemini, ou se eu quiser testar algum dos modelos da meta, ou qualquer outro LLM open source, eu só vim aqui trocar.

Eu consigo ver se eu tenho a mesma performance, sem ter que necessariamente correr em custos com a API. Depois que ele faz toda essa interação, eu vou ter então aqui uma lista, que ele vai aprendendo aqui. Isso aqui, como ele roda, essa simple sequential chain, ele vai rodar as chains em sequência.

Então eu tenho o L da corrente aqui, certo? Tenho esse outro L da corrente aqui. Eu simplesmente pego esse summary e coloco ele numa lista e passo para esse overall chain. Ele vai pegar esse primeiro input e ir aqui, que é um dos meus chunks, certo? E vai rodar tudo isso daqui.

No caso que eu defini ele como 10, porque senão vai ficar um vídeo muito longo. Ele demora um pouquinho para processar, então por isso eu só peguei 10. E 10 vocês vão ver que é suficiente para a gente ter uma qualidade muito boa.

Inclusive isso é importantíssimo vocês fazerem esses testes depois, justamente porque a partir desses testes, vocês vão conseguir entender se de fato a aplicação de vocês precisa de mais ou menos tokens para fazer isso. Quanto mais você colocar iterações aqui, mais vocês vão gastar na API. Por isso que é legal entender qual é o número ótimo, o número mínimo necessário para que vocês tenham um resultado aceitável, certo? E depois que eu fiz essas 196, o que eu vou fazer aqui? Eu tenho um último prompt aqui separado, que eu vou concatenar essas 196 summaries aqui em uma string só.

Vou passar para ele, com base nesses tópicos, me liste de 3 a 5 nichos desse canal. Que é de fato, a gente quer chegar a partir do link, nos nichos. Certo? Então, vamos rodar ele aqui agora.

Está demorando um pouquinho, vai demorar uns segundos para rodar. E aqui temos uma resposta. Entretenimento de cartas, cultura pop, jogos de cartas colecionáveis, ele não bem vindo de cartas colecionáveis, faz sentido no canal de Magic.

Vamos fazer o seguinte agora, vamos estressar esse cara. Além do canal de Magic, vou colocar o meu canal agora. Esse aqui não preciso.

Roda aqui, roda aqui. Vamos rodar tudo de novo. Esse aqui não preciso.

E vamos ver quais são os nichos que ele entende do meu canal. Vou colocar menos aqui para a gente conseguir rodar rápido e ter os resultados praticamente rápidos aqui no nosso vídeo. Beleza, tecnologia e inovação, programação e a carreira e conteúdo, engenharia e data science, matemática e Python.

Bem os nichos que eu trabalho aqui no canal. Vamos vai para um canal de receitas aqui, um canal de culinária. Vamos rodar isso aqui.

Enquanto roda esse último experimento do canal de receitas, vou aproveitar o vídeo, se você ficou até aqui, por onde você gosta do meu conteúdo, você gosta do meu jeito de falar, você gosta do meu jeito de explicar e eu quero te aproveitar para fazer um convite no dia 23 de maio, às 20 horas, eu vou realizar um evento online, onde eu vou mostrar para vocês como você é de uma matriz, sim aquela matriz, aquela coisa matemática que a gente meio abstrata, que a gente não sabe para que serve lá no ensino médio, e como você chegar nessa matriz e chegar num modelo de recomendação que potencialmente pode ser usado para Netflix, pode ser usado para recomendação de compra de remédios, recomendação de vídeos, então é um evento totalmente dedicado a como a gente transfere da matemática para o mundo real. Vou mostrar para vocês um método que eu estou desenvolvendo de estudos, que é o método IA profissional, esse é um método que vai ser baseado em três pilares, que eu já trago aqui no YouTube, que eu decidi formalizar essa coisa, estudar e mostrar para vocês. O primeiro pilar, modelo, matemática, a gente tem que entender a base, da maneira que a gente faz aqui no YouTube, passo a passo, sem grandes saltos, vocês entendendo todas as etapas.

O segundo pilar é o algoritmo, não adianta nada entender a matemática se não souber programar de maneira eficiente essa matemática, então a gente vai aprender a programar, como a gente pega uma matriz, cria uma solução de algoritmo em cima dessa matriz, e o terceiro pilar é o mundo, ou seja, como a gente faz um deploy dessa coisa no mundo real, na AWS. Não adianta nada você saber a base da matemática, não adianta nada você saber transformar isso num algoritmo, se você não souber as nuances de colocar isso em produção no mundo real. Então a ideia desse evento é apresentar para vocês toda essa metodologia, e no final tem uma grande surpresa para vocês, o link para a inscrição no evento vai estar aqui embaixo no canal, espero que vocês possam me prestigiar nessa noite do dia 23 com a presença de vocês, vou ficar muito feliz e vamos continuar agora aqui no nosso vídeo.

Vemos aqui que temos o resultado já, receita, fitness, receita vegana, dicas de comida para iniciantes, então, eu acredito que já está muito bem provado como é que funciona, que realmente a nossa epoca funciona, vou pegar só mais um aqui que é do nicho de ciências, e olha que interessante, 10 partes do nosso corpus já foi o suficiente para criar nichos de qualidade, ele não viajou na maioria de nenhum dos nichos, a gente testou aqui para o nicho de meu, tecnologia e IA, a gente testou para o nível de games, cards magic, testou para o nível de culinária, receita, funcionou, vamos ver agora o resultado para o nicho de ciência, isso que chama ciência, curiosidades, química, biologia, se não me engano é esse canal. Educação científica para crianças e adolescentes, media aprendizado lógico, divulgação científica abordando apenas temas como neurociência e psicologia, de forma acessível e interessante para o público em geral, exploração da evolução em biologia, aprendendo a descobrir as ciências curiosas do assunto, programação de ciência e tecnologia, mostrando os... olha só que interessante, eu acho que isso aqui deve ser um dos últimos vídeos que ele passou, se eu souber lembrado o meu, ele falava alguma coisa de Geminar, ele falava sobre educação, porque recentemente eu fiz um vídeo sobre como se estudava Python. Ele dizia que talvez valeria a pena, por exemplo, a gente aumentar um pouco esse cara aqui.

Só por curiosidade, vocês aguentem esperar um tempinho aí, vamos colocar 30 aqui, nem sei se tem 47, vamos lá, vamos processar os 47 chunks, que é tudo que ele tem do último vídeo. Vai demorar um tempinho, mas vamos ver se ele deixa mais acurado. Essa parte de validação de LLM é muito difícil, diferente de técnicas tradicionais de AI Machine Learning, que a gente tem a fonte da verdade, que é o famoso, o dataset de teste, que a gente separa do treino, e que a gente consegue, a partir disso, validar os nossos modelos, a gente não tem isso com LLM, a gente tem que ter uma validação mais qualitativa e sutil.

Por exemplo, como eu faço e estou fazendo isso muitos nos últimos meses, eu já peguei essas utilidades, olha, opa, ele está com uma memória de muito curto prazo, principalmente porque eu uso meus vídeos, eu sei o meu conteúdo, eu sei que nas últimas semanas eu trabalhei esses conteúdos sobre como estudar Python e foi para esse lado mais educacional. Como ele foi para um lado assim, quando ele quase descreveu o último vídeo desse canal de ciência, eu acho que ele está com uma memória de relativamente curto prazo. Então, vamos tentar aumentar e deixar mais informação para ele, para ver se ele generaliza um pouco mais e cria nichos um pouco mais genéricos para a gente, principalmente porque o que a gente vai usar como resultado, a gente vai colocar essa peça em outra parte do nosso programa, que a gente vai pesquisar os vídeos relevantes para gerar o conteúdo.

Olha só que interessante, eu aumentei um pouco o tamanho dele e ele já generalizou mais. Divulgação científica acessível e didática, esse canal é super falso sobre isso, neurociência e saúde, fala bastante sobre isso, evolução em ciências naturais, astronomia e universo, história da ciência e curiosidades. Então olha só que coisa sutil, se a gente dá mais informação para ele, ele consegue generalizar mais.

Por outro lado, se eu quero que ele pegue mais tendências, talvez nos últimos dias, eu tenho que diminuir esse range que eu alimento ele. Cumprimos aqui a nossa missão, terminamos aqui de criar o nosso vídeo, onde a gente parte de uma URL aleatória de qualquer vídeo de YouTube e esse código consegue falar para a gente quais são os nichos desse canal. Muito interessante, não é, gente? Vou pensar um pouquinho agora quais os próximos passos, se a gente já começa a transformar isso no código final ou se a gente faz outras validações.

Minha tendência é, vamos fazer outras validações, seria que, qual que é o próximo passo? Talvez pensar como a gente vai conseguir achar conteúdo relevante dentro desses nichos. Se você gostou do conteúdo, não esquece de seguir o canal, deixa esse like aqui. Fica alguma dúvida, deixa os comentários e a gente se vê no próximo vídeo.

Tchau, tchau, gente!

**Transcrito por** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell)**.** [**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **para remover esta mensagem.**